ARMY RESEARCH LABORATORY

Composite Classifiers for Automatic Target Recognition

Lin-Cheng Wang, Sandor Der, and Nasser M. Nasrabadi

ARL-TR-1661 November 1998

Approved for public release; distribution unlimited.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-1661 **November 1998**

Composite Classifiers for Automatic Target Recognition

Lin-Cheng Wang, Sandor Der, and Nasser M. Nasrabadi

Sensors and Electron Devices Directorate

Abstract

Composite classifiers that are constructed by combining a number of component classifiers have been designed and evaluated on the problem of automatic target recognition (ATR) using forward-looking infrared (FLIR) imagery. Two existing classifiers, one based on learning vector quantization and the other on modular neural networks, are used as the building blocks for our composite classifiers. We analyze a number of classifier fusion algorithms, which combine the outputs of all the component classifiers, and classifier selection algorithms, which use a cascade architecture that relies on a subset of the component classifiers. Each composite classifier is implemented and tested on a large data set of real FLIR images. The performances of the proposed composite classifiers are compared based on their classification ability and computational complexity. We demonstrate that the composite classifier based on a cascade architecture greatly reduces computational complexity, with a statistically insignificant decrease in performance in comparison to standard classifier fusion algorithms.

Contents

1 Introduction 1

2 Two ATR Classifiers 4

3 Composite Classifier Architectures 7

3.1 Averaged Bayes Classifier 8

3.2 Stacked Generalization Method 8

3.3 Quality-Based Cascade Classifier 10

4 Simulation and Experimental Results 14

4.1 Individual Classifier Performance 14

4.2 Results of Combination Algorithms 17

4.2.1 Result of Averaged Bayes Classifier 17

4.2.2 Result of Stacked Generalization Method 17

4.2.3 Result of Quality-Based Cascade Classifier 18

5 Conclusions 23

References 24

Distribution 27

Report Documentation Page 29

Figures

1 Examples of 10 target chips at viewing angle 90° 3

2 Performance curve of classifier ATR_LVQ as function of codebook usage 6

3 Architecture of stacked generalization classifier 9

4 Architecture of cascade classifier 10

5 Distributions of confidence levels for classifiers 12

6 Performance comparison of ATR_LVQ and ATR_MNN classifiers on the training and testing sets 16

7 Performance and CPU time of cascade classifier at various threshold values 19

Tables

1 Comparison of classification probability and computation complexity 4

2 Heuristic comparison of ATR_LVQ and ATR_MNN classifiers 6

3 Confusion matrices of ATR_LVQ on training and testing sets. 15

4 Confusion matrices of ATR_MNN on training and testing sets 16

5 Performance of averaged Bayes classifier on training and testing sets 18

6 Performance of stacked generalization method on training and testing sets 18

7 Performance of cascade classifier on training and testing sets 20

8 Performance of stacked generalization method with different numbers of hidden nodes 20

9 Performance of stacked generalization method on the training and testing sets (99.65 and 95.29%) 21

10 Performance of quality-based multiple-stage classifier method on the training and testing sets (99.65 and 95.29%) 22

1. Introduction

We propose methods of constructing a classifier architecture that combines several individual classifiers to form a composite classifier for the problem of automatic target recognition (ATR) using forward-looking infrared (FLIR) imagery. Many methods of combining several classifiers have been proposed to improve classification performance. These may be grouped into two basic approaches: *classifier fusion* and *classifier selection* [1,2]. In a classifier fusion algorithm, all classifiers are executed, and a mixing algorithm combines all the outputs from all the classifiers. Since all the classifiers contribute their outputs to a mixing algorithm, the computational complexity is no less than the sum of the computational complexities of all the classifiers. Cho and Kim [3] use fuzzy integrals to combine individual neural network classifiers trained with varying numbers of features. Turner and Ghosh [4] introduce an order statistic fusion algorithm and analyze its bias and variance as a function of the bias and variance of component classifiers. Jacobs et al [5] use an algorithm that adjusts mixing weights as a function of each input datum. A large number of voting schemes have been proposed, such as simple majority vote, the Borda count, and approval voting [6]. Other examples of classifier fusion algorithms can be found elsewhere [1,7–11]. In a classifier selection algorithm, a separate selection component chooses the appropriate classifier to classify each particular target image. The computational complexity of this approach is much lower than that of the classifier fusion algorithms, since not all the classifiers participate in classifying a target image. The classifier fusion algorithms tend to have better performance. For our ATR application, it is necessary to design a composite classifier that improves both classification performance and computational efficiency.

The idea behind our integration of several classifiers is to have individual classifiers learn particular *subspaces of data* and to choose these subspaces so that they are as disjoint as possible, which allows performance improvement to be achieved by combining the classifiers. Approaches that one could take include: mixture of experts, different features, and different learning algorithms. In a mixture of experts algorithm [12], the experts are specialized when they learn different subspaces of data, and the final decision is obtained by the combination of the outputs of the experts in a linear manner. An integrating unit, called a gating network, is used to

provide the weightings for the combination of the experts and/or to select an appropriate expert for classifying an input datum. In the second approach, various data transformations create different representations of data (different features). These representations can then be used to train separate classifiers. While the transformations do not add any information, they do affect the information captured by the learning algorithm, making some information more readily learnable and concealing some information from the learning algorithm. This means that algorithms trained using different transformations will not be perfectly correlated, and can potentially be combined to boost performance. For example, data with different time/frequency resolutions, created by a wavelet transform, are used to design stage neural networks in a parallel consensual neural network [7]. We have used the above two approaches to design two ATR classifiers, Automatic Target Recognizer Learning Vector Quantizer (ATR_LVQ) [13] and Automatic Target Recognizer Modular Neural Networks (ATR_MNN) [14]. The third approach is to design several classifiers with different learning algorithms and combine them with a mixing strategy. In this approach, each component classifier naturally creates different learning subspaces. We adopt this third approach to design and combine the two classifiers. The two ATR classifiers, ATR_LVQ and ATR_MNN, which have different learning algorithms, are used to implement the proposed composite classifiers.

In this report, we propose a multi-stage selection strategy. The composite classifier consists of multiple stages of classifiers. The selection is based on the estimated quality of classification produced by the component classifiers. All target images are classified by the first stage classifier. Next, we define the *certainty* of classification as the output margin between the highest and the second highest confidence outputs of a classifier. By experiment we found that both ATR single classifier candidates tend to produce high certainties when a correct classification is made, while they produce low certainties when an incorrect classification is made. Because the probability of correct classification is correlated with certainty, we want to reject the classification results with low certainty and accept the classification results with high certainty. The classifier in the second stage classifies the low certainty target images and generates a new certainty of classification. The second stage classifier also rejects some target images with low certainty, which will be classified by the next stage classifier. This cascading method can be extended to any number of classifiers until satisfactory results are achieved.

ATR is a technique that discloses potential targets in an image and then identifies the types of the targets. Most ATR designs consist of several

stages [15]: At the first stage, a target detector, operating on an entire image, detects some potential targets, and marks the regions of interest for further processing. In order to reduce the false-alarm rate, the second stage uses a more sophisticated classification technique to reject false target-like objects (clutter) from the real targets. The third stage classifies each target image into one of a number of classes. This report focuses on the last stage: classification.

Target recognition using FLIR imagery of natural scenes is difficult because of the variability of target thermal signatures. Collected under different meteorological conditions, times of the day, locations, ranges, etc, target signatures exhibit dramatic differences in appearance, as shown in figure 1. Moreover, partial target obscuration and the presence of target-like objects in the background make target recognition unreliable. The high variability of target signatures implies that large FLIR data sets are required if statistical learning algorithms are to generalize well [16].

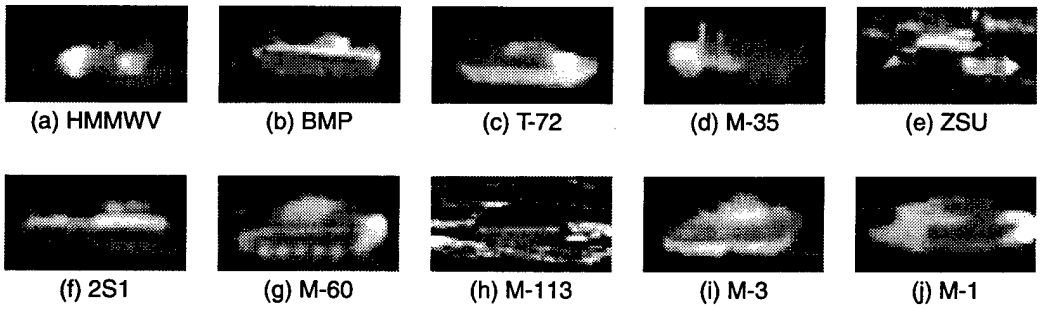


Figure 1. Examples of 10 target chips at viewing angle 90°.

2. Two ATR Classifiers

Two recently developed ATR classifiers, one based on learning vector quantization that we call ATR_LVQ [13] and another based on modular neural networks called ATR_MNN [14], were used to implement the proposed composite classifier. ATR_LVQ is designed by separating the training images into target-aspect groups. Each target-aspect group contains only one target type within a restricted range of viewing angles. The training images are decomposed into wavelet subbands, and then each wavelet subband of each target-aspect group is clustered using the *K*-means algorithm in order to create a set of target templates (codevector). A modified learning vector quantization (LVQ) algorithm is then applied to the template's enhance discriminatory ability. Testing is performed by calculating the mean square error between an input image and every code vector. This classifier performs well on both training and testing sets; it achieves almost 100 percent accuracy on the training set and 93.50 percent on the testing set, as shown in table 1. Note that the computation complexity is measured by the CPU time for running 3,463 target images on the testing set on a SUN Ultra 1. This outcome is not surprising because VQ is a *universal classifier*, in that it can classify targets arbitrarily well given a sufficiently large number of codevectors and an adequate training set, at the cost of high computational complexity.

The classifier ATR_MNN employs a hierarchical neural network architecture, specifically a mixture of experts modular neural network [12], with each expert consisting of a committee of neural networks [9] architecture. In the mixture of experts modular network, each expert is trained for a particular subset of data vectors. A gating network is trained to select or combine the outputs of expert networks in order to form the final output. The partitioning of the data set into several subsets is based on the similarity of target silhouettes. The silhouettes are binary images formed from computer-aided design (CAD) models of the targets, and the clustering is performed using the well-known *K*-means algorithm [16] on

Table 1. Comparison of classification probability and computation complexity.

Classifier	Training	Testing		Complexity (s)
Averaged Bayes classifier	99.11	94.48	2133.0	(= 1936.0 + 197.0)
Stacked generalization method	99.66	95.32	2133.0	(= 1936.0 + 197.0)
Cascade classifier	99.39	94.63	603.6	(= 197.0 + (1936.0 × 0.21))

the binary silhouettes. No optimal algorithm exists for partitioning the data set, so the choice of partitioning was guided by intuition and confirmed by experiment. Undoubtedly, further experimentation could lead to a superior partition. In the committee of networks, each member network receives distinct inputs, which are features extracted from one local region of the target image. That is, the input of a member network in the committee of networks is a *sub-vector* (the input data vector is partitioned). This reduces the dimensionality of the data, allowing each neural net to have fewer weights. The technique requires that the targets be reasonably close to the center of the input image because the networks are not shift invariant. It was determined experimentally that the centering error in our detector algorithm was small enough that it did not cause significant error in the recognition results. The classification decisions of the individual members of a committee of networks are combined using *stacked generalization* [10], which uses an additional neural network whose input is the output of the low-level neural nets. The classifier achieves good performance with much lower computational complexity than ATR_LVQ. The probabilities of correct classification for the training and testing sets are 97.44 and 92.06 percent, respectively. A heuristic comparison of the approaches of both classifiers is given in table 2.

Both classifiers can accurately classify targets in the training set, but leave a performance gap for the testing set. It is doubtful that large improvements in performance could be obtained by minor changes to the architectures of these classifiers. For example, figure 2 shows the performance curve of ATR_LVQ in terms of its codevector usage. The first half of frequently used codevectors contributes around 95 and 88 percent accuracy for the training and the testing sets, respectively. The increase in performance due to the other half of the codevectors is only about 5 percent for both data sets. Both performance curves nearly reach their maximum. Combining several classifiers to form a composite classifier seems a more promising way to improve classification performance.

One of the major obstacles to achieving a high classification performance is that the training set is too small to adequately characterize the variability of FLIR signatures. This forces the ATR designer into a tradeoff between classifier complexity and generalization capability. If the algorithm is too complex for the training set, the performance on the training set will be quite high, while the performance on the testing set (the generalization) will be low. Typically, the number of free parameters in a learning algorithm should be approximately one tenth the number of training samples. This forces the classifier to be too simple. In other words, the performance of learning-based ATRs is data limited. Significantly

improved performance is possible with a sufficiently large data set, but the cost of such a set is prohibitive.

Table 2. Heuristic comparison of ATR_LVQ and ATR_MNN classifiers.

Approach	ATR_LVQ	ATR_MNN
Training	Clustering	Discriminant function
Classification	Template matching	Posteriori probability
Implementation	VQ, LVQ	Neural network (MLP)
Learning algorithm	Largely unsupervised	Supervised
Adaptation scope	Localized	Global
Feature type	Wavelet	Directional variance
Feature decomposition	Wavelet subbands	Local receptive field
Parameterized	Aspect window by target	Silhouette category
Testing speed	Slow	Fast
CPU time to test testing set	1936.0 s	197.0 s
Performance	High	High
Training set	99.47%	97.34%
Testing set	93.50%	92.20%

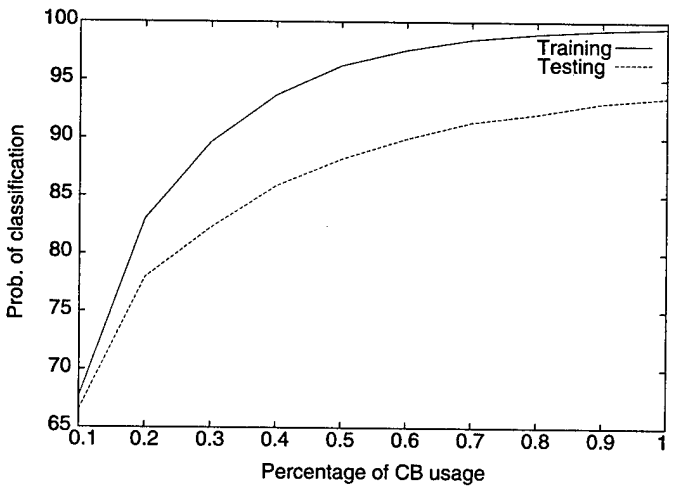


Figure 2. Performance curve of classifier ATR_LVQ as function of codebook usage.

3. Composite Classifier Architectures

Let us define the notation used in this report. Suppose we have a set of K classifiers, $\mathcal{C}_k$, each of which classifies targets into one of Q distinct classes, where $k = 1, 2, \ldots, K$. The output vector of classifier $\mathcal{C}_k$, given a target $\mathbf{X}$, is represented by a column vector

$$\mathbf{Y}_k = \{y_{k,q}; q = 1, 2, \ldots, Q\}, \tag{1}$$

where the qth component of the output vector, $y_{k,q}$, represents the estimated posteriori probability that target $\mathbf{X}$ belongs to the class q, estimated by classifier $\mathcal{C}_k$. We can then express the estimated posteriori probability as the desired posteriori probability $p(q|\mathbf{X})$ plus an error $\epsilon_{k,q}(\mathbf{X})$:

$$y_{k,q} = p(q|\mathbf{X}) + \epsilon_{k,q}(\mathbf{X}). \tag{2}$$

For notational convenience, we do not explicitly express the dependence of outputs $y_{k,q}$ on variable $\mathbf{X}$, where $\mathbf{X}$ is the input of each individual classifier $\mathcal{C}_k$. The ground truth class of a target $\mathbf{X}$ is θ_T. The output vector of a composite classifier, $\mathcal{C}$, given a target $\mathbf{X}$, is represented by a column vector

$$\mathbf{Y}(\mathbf{X}) = \{y_q(\mathbf{X}); q = 1, 2, \ldots, Q\}, \tag{3}$$

where $y_q(\mathbf{X})$, the qth component of the output vector, is the estimated posteriori probability that target $\mathbf{X}$ belongs to the class q.

The classification decision of classifier $\mathcal{C}_k$ is

$$\theta_k = \arg \max_{1 \le q \le Q} y_{k,q}. \tag{4}$$

The final decision of a composite classifier $\mathcal{C}$ is

$$\theta = \arg \max_{1 \le q \le Q} y_q. \tag{5}$$

The efficiency of a composite classifier is also considered. The computational complexities of classifier $\mathcal{C}_k$ and a composite classifier $\mathcal{C}$ are denoted as τ_k ($k = 1, 2, \ldots, K$) and τ, respectively.

We propose a multi-stage classifier using quality-based classifier selection. We also implement two other composite algorithms that use classifier fusion for comparison to our proposed algorithms, which are presented first.

3.1 Averaged Bayes Classifier

This simple mixing algorithm takes the average outputs of a set of classifiers as a new estimated posteriori probability of a composite classifier.

$$\mathbf{Y} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{Y}_k, \tag{6}$$

or equivalently,

$$y_q = \frac{1}{K} \sum_{k=1}^{K} y_{k,q}. \tag{7}$$

The final decision C made by this composite classifier is given by

$$\theta = \arg \max_{1 \le q \le Q} y_q. \tag{8}$$

Thus, each individual classifier is weighted equally.

This mixing algorithm is called an averaged Bayes classifier by Xu et al [11], where they assumed that all individual classifiers are Bayes classifiers. Perrone [9] has shown theoretically that the averaging over the outputs of a set of neural networks can improve the performance of a neural network, in terms of any convex cost function. This algorithm provides not only better performance, but also better generalization than a single classifier.

Because the computational overhead to combine classifiers is minimal, the computational complexity approximately equals

$$\tau = \sum_{k=1}^{K} \tau_k. \tag{9}$$

3.2 Stacked Generalization Method

The averaged Bayes classifier algorithm treats each individual classifier equally. However, it is possible that some classifiers can make better decisions than others for some targets. Thus, we can reduce the probability of misclassification if we assign larger weightings to some classifiers than to other classifiers for some targets. Two approaches that provide linear weighting to individual classifiers are explored by Benediktsson [7] and Perrone [9]. The first approach, called *generalized committee* [9], obtains the weighting of each component classifier by solving the error correlation matrix. The second approach, similar to the generalized committee, is supported by consensus theory [7]. The optimal weighting is obtained by solving the Weiner-Hopf equation.

Better performance can be expected if nonlinear weighting is applied to individual classifiers, since a linear function is a special case of a nonlinear function. The stacked generalization method [10] is a general approach that combines a set of classifiers to obtain a final decision. A multilayer perceptron (MLP) neural network that receives the output of all classifiers can be trained to implement the combination:

$$\mathbf{Y} = \Phi(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_k), \tag{10}$$

where $\Phi(\cdot)$ is an MLP that implements the stacked generalization. This MLP provides a nonlinear weighting to the outputs of individual classifiers. The architecture is shown in figure 3.

The computational complexity of the stacked generalization method is expressed as follows:

$$\tau = \sum_{k=1}^{K} \tau_k + \tau_{sg}, \tag{11}$$

where τ_{sg} is the computational complexity computed by the stacked generalization neural network. In general, τ_{sg} is much smaller than any τ_k.

The above two algorithms are data fusion algorithms, which execute all classifiers in parallel. Thus, the computational complexities are equal to the sum of the computational complexities of all classifiers and the combination algorithm.

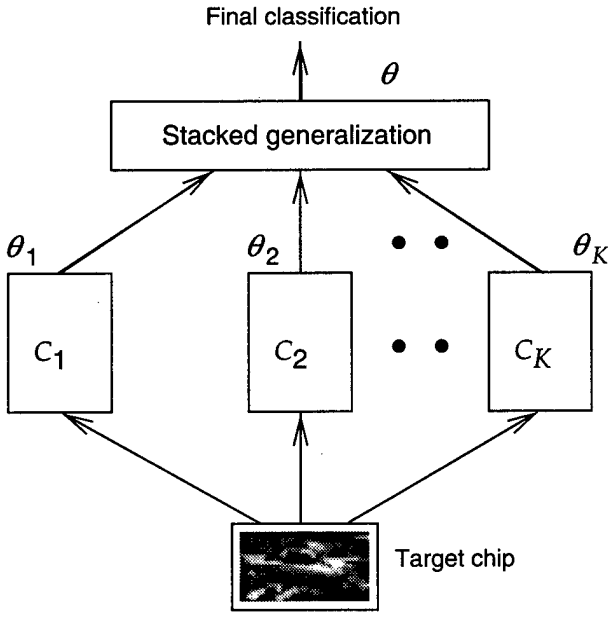


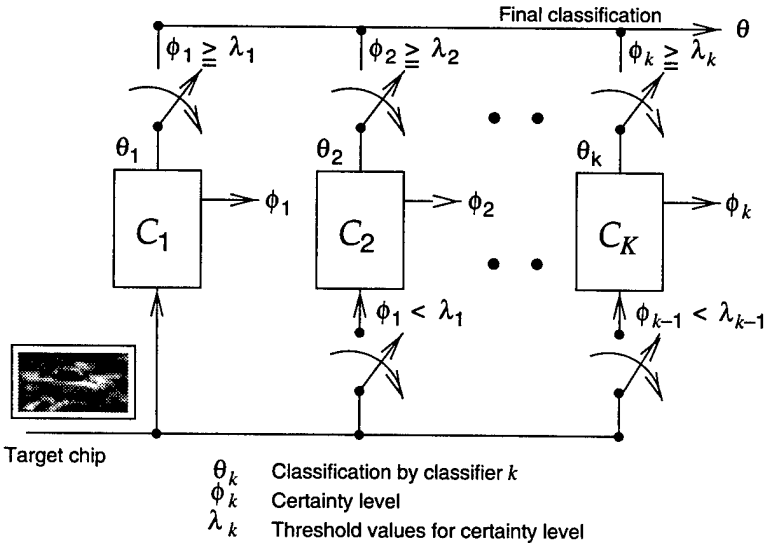
Figure 3. Architecture of stacked generalization classifier.

3.3 Quality-Based Cascade Classifier

A cascade classifier consists of several different classifiers cascaded together. All target images are classified by the first stage classifier, and targets that are not classified with high certainty are passed to the next stage classifier. We define the *certainty* of classification as the output for the winning class minus the highest output among the losing classes. By experiment we found that both ATR single classifier candidates tend to produce high certainties when a correct classification is made, while they produce low certainties when an incorrect classification is made. Because the probability of correct classification is correlated with certainty, we want to reject the classification results with low certainty and accept the classification results with high certainty. We define the threshold at stage k to be λ_k. The second stage classifier classifies each of the target images with certainty less than λ_1, and generates a new certainty of classification for each input image. The second stage classifier also rejects those target images with certainty below λ_2, which will be classified by the next stage classifier. This constructive method can be extended to any number of classifiers until satisfactory results are achieved. The λ_k were chosen by experiment. Each λ_k was set between 0 and 1 in increments of 0.1, and the resulting classification performance and computational complexity tradeoff was calculated on the training set. The proposed method is illustrated in figure 4 and outlined below.

STEP 0: Given the target data set $T = [\mathbf{X}_l; l = 1, 2, \ldots, M]$ and a set of K well-designed classifiers C_k, where $k = 1, 2, \ldots, K$, select the threshold values λ_k for each stage. Set $k \leftarrow 1$.

Figure 4. Architecture of cascade classifier.



STEP 1: Set $l \leftarrow 1$.

STEP 2: If $l < M$ then set $k \leftarrow 1$; otherwise, STOP.

STEP 3: Classify $\mathbf{X}_l$ by classifier C_k. Obtain the certainty of classification $\phi_k(\mathbf{X}_l)$.

$$\text{top1} = \max_{1 \le q \le Q} y_{k,q}(\mathbf{X}_l) \tag{12}$$

$$\text{top2} = \max_{1 \le q \le Q, q \ne \theta_k} y_{k,q}(\mathbf{X}_l) \tag{13}$$

$$\phi_k(\mathbf{X}_l) = \text{top1} - \text{top2} \tag{14}$$

STEP 4: If current classifier is the last stage classifier or a high certainty of classification is obtained at this stage, then the decision of this stage classifier is the final decision. Otherwise, this target image is sent to the next stage classifier.

If $\{ (k = K) \text{ or } (\phi_k(\mathbf{X}_l) \ge \lambda_k) \}$, then
$\theta = \theta_k$,
$l \leftarrow l+1$,
go to step 2.
Otherwise,
Set $k \leftarrow k+1$,
go to step 3.

Figure 5 shows the distributions of certainty of classification for both correct and incorrect classifications for classifiers ATR_MNN and ATR_LVQ. It can be seen in figure 5 that most of the correctly classified target images have high certainty, whereas most of the incorrectly classified target images have low certainty.

The average computational complexity is the sum

$$\tau = \tau_1 + \sum_{k=2}^{K} p_k \, \tau_k, \tag{15}$$

where the p_k is the percentage of the data that are classified by classifier C_k $(k = 1, 2, \ldots, K)$. The classifiers at the other stages classify only a portion of the data that are classified by the previous stage classifiers, so that

$$0 \le p_k \le 1, \qquad 2 \le k \le K, \text{ and} \tag{16}$$

$$1 = p_1 \ge p_2 \ge \ldots \ge p_k \ge 0. \tag{17}$$

Compared to the other composite algorithms discussed, the computational complexity of this algorithm is lower:

$$\left\{ \tau_1 + \sum_{k=2}^{K} p_k \, \tau_k \right\} \le \left\{ \sum_{k=1}^{K} \tau_k \right\}. \tag{18}$$

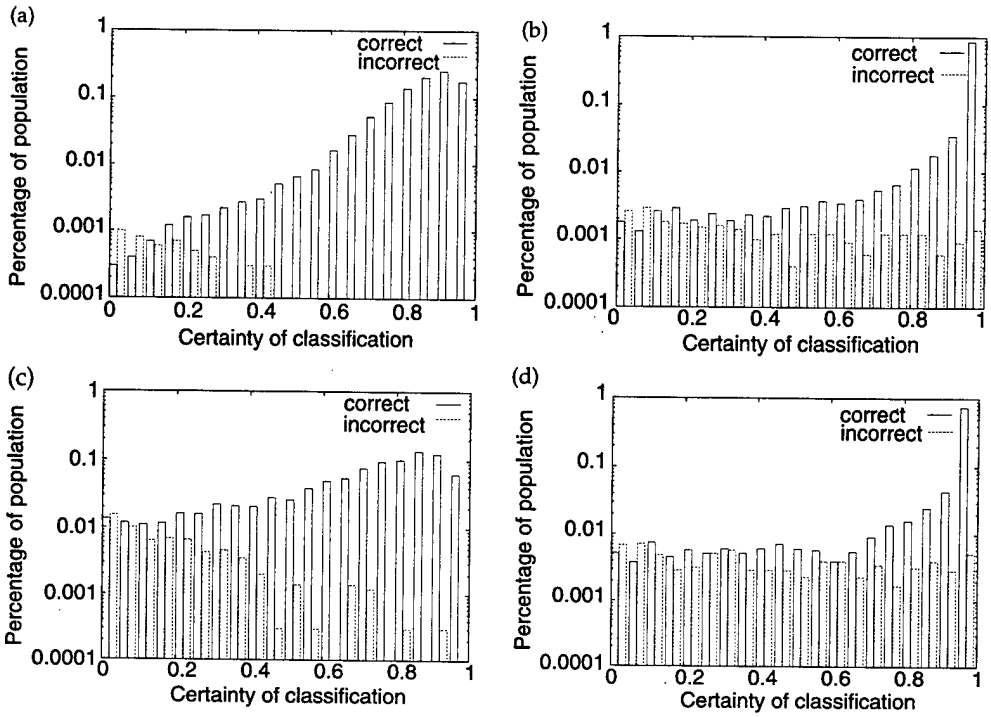


Figure 5. Distributions of confidence levels for classifiers ATR_LVQ and ATR_MNN (in semi-log scale): (a) training set of ATR_LVQ, (b) training set of ATR_MNN, (c) testing set of ATR_LVQ, and (d) testing set of ATR_MNN.

The complexities are equivalent only when the threshold values λ_k at each stage are set to the highest value 1.0 so that the classifiers at each stage reject all the target images. Since this combination algorithm selects the classification decision of one of the classifiers to be the final decision, the largest possible correct classification set is the union of all of the correct classifications of all the classifiers.

Let Ω be the data set and Ω_k be the subset of this data set that classifier C_k can correctly classify. Let $N()$ be the number of target images in a set. The set of data correctly classified by the cascade algorithm must always be a subset of

$$\Omega_u = \bigcup_{k=1}^{K} \Omega_k. \tag{19}$$

The best possible probability of correct classification for this cascade algorithm is

$$\frac{N(\Omega_u)}{N(\Omega)}. \tag{20}$$

It should be noted that the classifier fusion algorithms are not restricted by this upper bound. It is possible for the fused classifier to correctly classify a target that none of the individual classifiers correctly classify. We shall

see that the this upper bound does not seriously affect performance, since the classifier fusion algorithms classify almost none of the target images that are not correctly classified by any of the components.

4. Simulation and Experimental Results

The classifier has been trained and tested on the U.S. Army Comanche imagery set. This data set contains 10 military ground vehicles viewed from a ground-based, second-generation FLIR. The targets are viewed from arbitrary aspect angles, which are recorded in the ground truth (rounded to the nearest 5°). The images contain cluttered backgrounds and some partially obscured targets. The target signatures vary greatly, because the imagery was collected at different times of day and night, at different locations (Michigan, Arizona, and California), during different seasons, under varying weather conditions, and in different target exercise states. Figure 1 shows examples of target chips for the 10 military ground vehicles, selected from 2 FLIR imagery signature sets, called SIG and ROI, at viewing angle 90°. The data set contains 17,316 images, which are partitioned into a training set of 13,853 images, and a test set of 3,463 images. Classifiers are trained or designed using only the training set. The testing test is used to evaluate the genaralization performance of the classifiers.

4.1 Individual Classifier Performance

The two classifiers, ATR_LVQ and ATR_MNN, were designed using the same training and testing sets. The probabilities of correct classification for the training and testing sets, performed by the classifier ATR_LVQ, are 99.47 and 93.50 percent, respectively. The probabilities of correct classification using ATR_MNN are 97.34 and 92.20 percent, respectively. The confusion matrices of both classifiers for the testing data sets are shown in tables 3 and 4. The performance of ATR_LVQ is slightly better than that of ATR_MNN, in terms of the probability of correct classification. However, the computational complexity of ATR_MNN is much lower; it needs only one tenth of the CPU time of ATR_LVQ. The neural networks execute quite quickly once training is completed, while the ATR_LVQ algorithm must compare the input images to each of the 1159 code vectors in the codebook.

Figure 6 shows the performance of classifiers ATR_LVQ and ATR_MNN on the training and testing sets. We have also broken down the probabilities of correct classification into categories to show the level of independence of the two classifiers. The figure shows that 4.76 percent of

Table 3. Confusion matrices of ATR_LVQ on training (99.47%) and testing (93.50%) sets.

Type	HMMWV	BMP	T-72	M-35	ZSU-23	2S1	M-60	M-113	M-3	M-1	Percentage	Total
Training set												
HMMWV	1691	1	0	0	0	0	1	0	1	2	99.71	1696
BMP	4	1657	0	0	0	0	1	0	0	0	99.70	1662
T-72	2	4	1805	1	1	2	3	0	1	1	99.18	1820
M-35	2	0	2	1576	1	2	0	0	1	1	99.43	1585
ZSU-23	2	0	4	0	1165	0	0	1	1	0	99.32	1173
2S1	1	0	2	3	1	1165	0	2	0	0	99.23	1174
M-60	2	1	1	2	0	1	1699	0	0	1	99.53	1707
M-113	0	2	0	0	1	0	0	695	0	2	99.29	700
M-3	4	2	0	0	0	1	0	1	1161	0	99.32	1169
M-1	0	1	1	0	0	0	0	0	0	1165	99.83	1167
Testing set												
HMMWV	415	1	2	0	1	0	2	0	2	1	97.88	424
BMP	11	390	4	1	1	2	1	0	5	1	93.75	416
T-72	2	12	425	2	1	0	3	1	6	2	93.61	454
M-35	7	2	2	377	3	1	4	0	0	1	94.96	397
ZSU-23	1	0	2	5	276	0	3	0	3	3	94.20	293
2S1	5	3	3	6	0	271	0	3	1	2	92.18	294
M-60	3	2	7	3	5	1	392	1	5	8	91.80	427
M-113	4	3	2	4	0	1	1	154	3	2	88.51	174
M-3	6	5	0	6	3	0	1	4	267	1	91.13	293
M-1	3	1	1	1	2	0	7	2	3	271	93.13	291

the target images in the testing set are correctly classified by ATR_LVQ but are not correctly classified by ATR_MNN, and 3.47 percent of the target images in the testing set are correctly classified by ATR_MNN but are not correctly classified by ATR_LVQ. This implies that a composite classifier that has the ability to choose between using the ATR_LVQ and ATR_MNN outputs (i.e., a hard limited gating network) has a theoretical upper bound of 96.97 percent. The disjoint outcome of these two classifiers suggests that we can improve the performance of an ATR classifier by combining both classifiers to form a final decision.

Table 4. Confusion matrices of ATR_MNN on training (97.34%) and testing (92.20%) sets.

Type	HMMWV	BMP	T-72	M-35	ZSU-23	2S1	M-60	M-113	M-3	M-1	Percentage	Total
Training set												
HMMWV	1654	12	9	5	4	1	3	4	2	2	97.52	1696
BMP	5	1626	4	1	2	11	3	2	3	5	97.83	1662
T-72	3	9	1768	2	1	8	14	1	6	8	97.14	1820
M-35	8	0	1	1561	0	2	4	4	1	4	98.49	1585
ZSU-23	1	0	2	0	1149	5	4	2	5	5	97.95	1173
2S1	0	17	1	1	3	1140	2	5	2	3	97.10	1174
M-60	5	9	16	9	6	4	1648	2	5	3	96.54	1707
M-113	3	3	2	2	2	4	7	666	0	11	95.14	700
M-3	2	8	3	3	0	5	5	7	1133	3	96.92	1169
M-1	0	12	2	2	0	0	6	4	1	1140	97.69	1167
Testing set												
HMMWV	408	2	2	4	1	2	0	3	0	2	96.23	424
BMP	3	398	1	1	4	7	0	0	0	2	95.67	416
T-72	1	10	415	1	2	4	9	3	4	5	91.41	454
M-35	5	6	1	369	3	2	3	4	1	3	92.95	397
ZSU-23	4	4	1	0	265	4	7	1	6	1	90.44	293
2S1	1	10	4	3	1	268	1	4	1	1	91.16	294
M-60	1	2	11	5	5	1	393	2	3	4	92.04	427
M-113	2	1	2	3	0	4	2	156	1	3	89.66	174
M-3	2	5	2	2	3	3	4	8	253	11	86.35	293
M-1	3	3	3	2	0	4	4	2	2	268	92.10	291

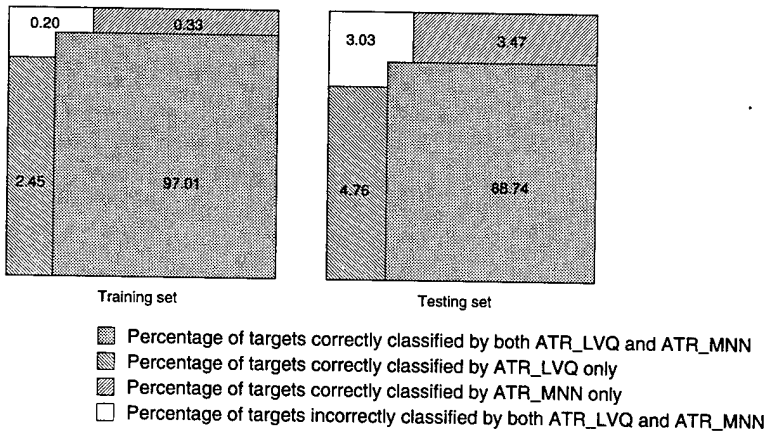


Figure 6. Performance comparison of ATR_LVQ and ATR_MNN classifiers on the training and testing sets.

4.2 Results of Combination Algorithms

4.2.1 Result of Averaged Bayes Classifier

The averaged Bayes classifier takes the average outputs of classifiers ATR_LVQ and ATR_MNN as a new estimated posteriori probability, as shown in eq (6). The result of the averaged Bayes classifier is shown in table 5. The probabilities of correct classification for the training and testing sets are 99.11 and 94.48 percent, respectively. It improves 0.98 percent for the testing set, but drops around 0.36 percent for the training set.

Table 5 shows the probabilities of correct classification of the four categories, performed by the averaged Bayes classifier, for the training and the testing sets. The averaged Bayes classifier can correctly classify, with 100-percent accuracy, the target images that are correctly classified by both classifiers ATR_LVQ and ATR_MNN. About 0.28 and 3.12 percent out of the 0.33 and 3.47 percent in the subset of the data correctly classified by ATR_MNN only are recognized correctly. This corresponds to 85 and 90 percent for the training and testing sets. The averaged Bayes classifier recognized only 1.82 and 2.63 percent, which are 74 of 2.45 percent and 55 of 4.76 percent, respectively, for the subset of the training and testing sets that were correctly classified by ATR_LVQ only. By averaging the outputs of these two classifiers, we see that both classifiers contribute the improvement in different levels.

4.2.2 Result of Stacked Generalization Method

An MLP neural network was designed to implement the stacked generalization. The inputs of the MLP are the outputs of the two classifiers, so that the MLP has 20 input nodes. The outputs of the MLP are the estimated posteriori probabilities for 10 target classes, so that the MLP has 10 output nodes. We designed several MLPs with different numbers of hidden nodes. The simulation results of these MLPs are shown in table 6. All the MLPs do improve the performance of both data sets. The MLPs with different numbers of hidden nodes generate almost equal performance, suggesting that the optimal output combination algorithm is relatively simple. Among them, the MLP with 20 hidden nodes is the best for both data sets. As we can see, stacked generalization is better than the averaged Bayes classifier, because the nonlinear weighting function is somewhat superior to the averaged sum function. Actually, we will see later that the stacked generalization method is the best combination method among the methods implemented in this report, in terms of the probability of correct classification.

Table 5. Performance of averaged Bayes classifier on training and testing sets, partitioned by single classifier performance.

Category	Training	Testing
Correctly classified by both	97.01/97.01	88.74/88.74
Correctly classified by ATR_LVQ only	1.82/2.45	2.63/4.76
Correctly classified by ATR_MNN only	0.28/0.33	3.12/3.47
Incorrectly classified by both	0.00/0.20	0.00/3.03
Averaged Bayes classifier (sum of above)	99.11/100	94.48/100

Table 6. Performance of stacked generalization method on training and testing sets, partitioned by single classifier performance.

Category	Training	Testing
Correctly classified by both	97.01/97.01	88.74/88.74
Correctly classified by ATR_LVQ only	2.43/2.45	3.96/4.76
Correctly classified by ATR_MNN only	0.22/0.33	2.57/3.47
Incorrectly classified by both	0.00/0.20	0.03/3.03
Stacked generalization method (sum of above)	99.65/100	95.29/100

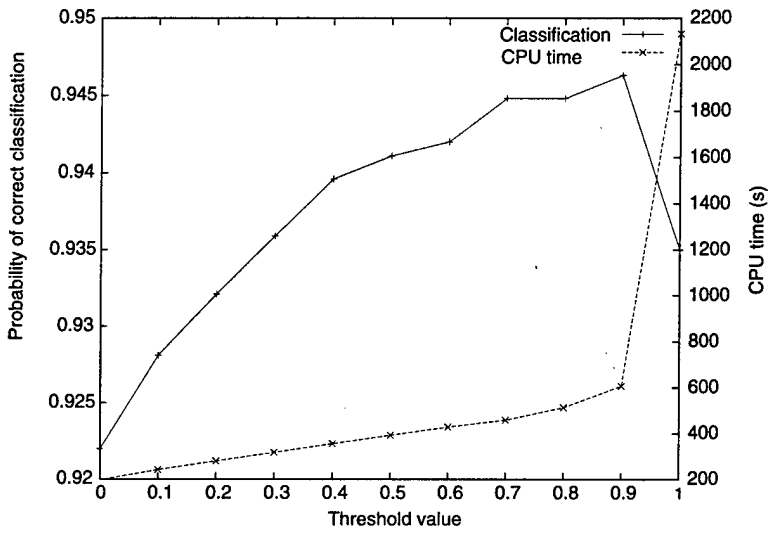
Table 6 shows the probabilities of correct classification of the four categories, performed by the stacked generalization method, for the training and the testing sets. It can be seen that the stacked generalization method can correctly classify, with 100-percent accuracy, the target images that are correctly classified by both classifiers ATR_LVQ and ATR_MNN. The MLP favors the outputs of the ATR_LVQ.

Conceivably, the MLP may learn how to correctly classify a significant portion of those target images that are not correctly classified by either classifier. However, we do not obtain this result in our simulation. We correctly classify only one image target (0.03%), which is incorrectly classified by both classifiers for the testing set.

4.2.3 Result of Quality-Based Cascade Classifier

We implemented a two-stage composite classifier to demonstrate the effectiveness of the quality-based cascade classifier. Since the computation of the ATR_MNN is more efficient than that of the ATR_LVQ, we use the ATR_MNN as the first stage and the ATR_LVQ as the second stage. Figure 7 shows the probability of correct classification and the CPU time of this two-stage classifier on the testing set, at 11 different thresholds for the certainty of classification, using increments of 0.1 from 0.0 to 1.0. If the threshold value is set to 1.0, the ATR_MNN (the first stage) rejects every target image and the ATR_LVQ (the second stage) has to perform classification for the whole data set, so that performance is equivalent to the ATR_LVQ alone with CPU time equal to the sum of the classifiers. When the threshold value is set to 0.0, the ATR_MNN (the first stage) is

Figure 7. Performance and CPU time of cascade classifier at various threshold values.



used to classify all target images and leaves no target images to the ATR_LVQ (the second stage). This combination is equivalent to the ATR_MNN alone in both performance and CPU time.

By using different threshold values, we obtain different two-stage classifiers. Among them, the combination with a threshold value 0.9 has the highest probability of correct classification, 94.63 percent for the testing set. A total of 2730 (78.83%) out of 3463 test images have a certainty of classification of at least 0.9. Thus, the ATR_LVQ classifies only 733 (21.17%) test images. The CPU time is much less than the CPU time needed by the ATR_LVQ, while the probability of correct classification for the cascade classifier is improved by 1.13 and 2.43 percent, when compared to that of the ATR_LVQ and the ATR_MNN, respectively. The probability of correct classification performed by the ATR_LVQ on these 2730 target images is 98.97 percent.

Table 7 shows the probabilities of correct classification of the four categories, performed by this two-stage classifier, for the training and the testing sets. As expected, this two-stage classifier attains 100 percent performance on the target images that are correctly classified by both ATR_LVQ and ATR_MNN.

We also implemented a two-stage classifier that uses the ATR_LVQ as the first stage and the ATR_MNN as the second stage (see table 8). The highest probabilities of correct classification for the training and testing sets are 99.60 and 94.54 percent, respectively, when the threshold value is set to 0.3. This two-stage composite classifier does improve the performance, but the CPU time is slightly greater than that of the ATR_LVQ alone.

Table 7. Performance of cascade classifier on training and testing sets, partitioned by single classifier performance.

Category	Training	Testing
Correctly classified by both	97.01/97.01	88.74/88.74
Correctly classified by ATR_LVQ only	2.26/2.45	4.33/4.76
Correctly classified by ATR_MNN only	0.12/0.33	1.56/3.47
Incorrectly classified by both	0.00/0.20	0.00/3.03
Proposed cascade classifier (sum of above)	99.39/100	94.63/100

Table 8. Performance of stacked generalization method with different numbers of hidden nodes.

Data set	Number of hidden nodes			
	5	10	20	40
Training	99.55	99.59	99.65	99.60
Testing	94.98	95.29	95.29	95.24

We observe the confusion matrices of the ATR_LVQ and the ATR_MNN for the testing set, as shown in tables 3 and 4. Both classifiers have excellent classification accuracy for target class high-mobility, multiwheeled vehicle (HMMWV), while both have poor performance for target classes M-113 and M-3. Target class ZSU-23 can be classified well by the ATR_LVQ, but poorly classified by the ATR_MNN, while target class M-60 is poorly classified by the ATR_LVQ, but classified well by the ATR_MNN. We can evaluate the performance of the stacked generalization method and the cascade classifier for these five target classes, as seen in tables 9 and 10.

Both the stacked generalization method and the cascade classifier do improve the accuracy for target class HMMWV. The stacked generalization method has a large improvement for target classes M-113 and M-3, although the probabilities of correct classification for these two classes are still the lowest among the 10 target classes. The cascade classifier also improves the probabilities of correct classification for these two classes, but not as significantly as stacked generalization does. The stacked generalization does not classify target class ZSU-23 as well as the ATR_LVQ. The cascade composite classifier outperforms both single classifiers.

Table 9. Performance of stacked generalization method on the training (99.65%) and testing (95.29%) sets.

Type	HMMWV	BMP	T-72	M-35	ZSU-23	2S1	M-60	M-113	M-3	M-1	Percentage	Total
Training set												
HMMWV	1693	2	0	0	0	0	0	0	0	1	99.82	1696
BMP	1	1658	2	0	0	1	0	0	0	0	99.76	1662
T-72	2	1	1811	1	0	1	2	0	1	1	99.51	1820
M-35	1	0	1	1580	0	2	0	0	0	1	99.68	1585
ZSU-23	0	0	2	0	1168	0	1	2	0	0	99.57	1173
2S1	0	0	1	0	2	1170	0	1	0	0	99.66	1174
M-60	2	1	1	1	0	1	1700	0	0	1	99.59	1707
M-113	1	1	0	0	1	0	0	696	0	1	99.43	700
M-3	4	0	0	0	0	1	0	0	1164	0	99.57	1169
M-1	0	0	2	0	0	0	0	0	0	1165	99.83	1167
Testing set												
HMMWV	418	2	2	0	0	0	0	0	0	2	98.58	424
BMP	6	402	2	0	0	3	0	0	2	1	96.63	416
T-72	0	8	431	2	0	2	7	0	2	2	94.93	454
M-35	7	1	2	384	0	1	2	0	0	0	96.73	397
ZSU-23	3	2	2	1	275	2	3	0	5	0	93.86	293
2S1	2	4	2	3	0	279	1	1	2	0	94.90	294
M-60	3	2	7	2	3	0	403	2	2	3	94.38	427
M-113	1	2	2	2	0	1	2	162	0	2	93.10	174
M-3	0	5	2	2	1	2	2	4	272	3	92.83	293
M-1	1	0	0	0	1	4	7	3	1	274	94.16	291

Table 10. Performance of quality-based multiple-stage classifier method on the training (99.65%) and testing (95.29%) sets.

Type	HMMWV	BMP	T-72	M-35	ZSU-23	2S1	M-60	M-113	M-3	M-1	Percentage	Total
Training set												
HMMWV	1691	2	0	0	0	0	0	0	1	2	99.71	1696
BMP	3	1655	0	0	0	2	1	0	0	1	99.58	1662
T-72	2	4	1807	0	0	1	5	0	0	1	99.29	1820
M-35	2	0	1	1577	1	2	0	0	1	1	99.50	1585
ZSU-23	1	0	2	0	1166	1	0	2	1	0	99.40	1173
2S1	1	4	2	3	1	1161	0	2	0	0	99.66	1174
M-60	2	3	3	1	1	1	1695	0	0	1	99.30	1707
M-113	1	2	0	1	1	0	1	691	0	3	98.71	700
M-3	4	0	0	0	0	1	0	2	1162	0	99.40	1169
M-1	0	2	1	0	0	0	0	0	0	1164	99.74	1167
Testing set												
HMMWV	418	0	2	1	0	0	0	0	2	1	98.58	424
BMP	9	399	3	0	0	2	0	0	2	1	95.91	416
T-72	2	11	429	2	1	0	3	1	4	1	94.49	454
M-35	7	2	1	380	1	1	4	0	0	1	95.72	397
ZSU-23	2	1	2	4	278	0	2	0	2	2	94.88	293
2S1	4	7	1	4	0	274	0	1	1	2	93.20	294
M-60	3	2	6	3	5	1	397	1	5	4	92.97	427
M-113	3	2	2	2	0	1	0	160	2	2	91.95	174
M-3	5	4	0	5	2	0	2	4	266	5	90.78	293
M-1	2	0	1	1	1	2	5	0	3	276	94.85	291

5. Conclusions

We have proposed, implemented, and demonstrated the utility of classifier selection combination algorithms that demonstrate high performance with relatively low average computational complexity. Performance and computational complexity of these algorithms are compared to standard techniques. The individual classifiers used in the composite classifiers have been described elsewhere. All experiments have been performed on a large database of challenging FLIR images.

References

1. T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Analysis and Machine Intelligence* **PAMI-16**(1), 66–75 (1994).

2. K. Woods, W. P. Kegelmeyer, Jr., and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Analysis and Machine Intelligence* **PAMI-19**(4), 405–410 (1997).

3. S. B. Cho and J. H. Kim, "Combining multiple neural networks by fuzzy integral for robust classification," *IEEE Trans. Systems, Man, and Cybernetics* **SMC-25**(2), 380–384 (1995).

4. K. Turner and J. Ghosh, "Theoretical foundations of linear and order statistics combiners for neural pattern classifiers," Computer Vision Research Center, University of Texas at Austin, TR-95-02-98 (1995).

5. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation* **3**, 79–87 (1991).

6. D. G. Saari, *The Geometry of Voting*, Springer-Verlag, New York (1994).

7. J. A. Benediktsson, J. R. Sveinsson, O. K. Ersoy, and P. H. Swain, "Parallel consensual neural networks," *IEEE Trans. Neural Networks* **NN-8**(1), 54–64 (1997).

8. C. Ji and S. Ma, "Combinations of weak classifiers," *IEEE Trans. Neural Networks* **NN-8**(1), 32–42 (1997).

9. M. P. Perrone, "General averaging results for convex optimization," *Proc. Connectionist Models Summer School*, 364–371 (1993).

10. D. H. Wolpert, "Stacked generalization," *Neural Networks* **5**, 241–259 (1992).

11. L. Xu, A. Krzyżak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Systems, Man, and Cybernetics* **SMC-22**(3), 418–435 (1992).

12. R. A. Jacobs and M. I. Jordan, "Learning piecewise control strategies in a modular neural network architecture," *IEEE Trans. Systems, Man, and Cybernetics* **23**, 337–345 (1993).

13. L. A. Chan, N. M. Nasrabadi, and V. Mirelli, "Multi-stage target recognition using modular vector quantizers and multilayer perceptrons," *Proc. Computer Vision Pattern Recognition*, 114–119 (1996).

14. L. C. Wang, S. Z. Der, and N. M. Nasrabadi, "A committee of networks classifier with multi-resolution feature extraction for automatic target recognition," *Proc. IEEE International Conference on Neural Networks*, III-1596 (1997).

15. B. Bhanu, "Automatic target recognition: state of the art survey," *IEEE Trans. Aerospace and Electronic Systems* **AES-22**(4), 364–379 (1986).

16. S. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Co., New York (1994).

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Ofc of the Secy of Defns
Attn ODDRE (R&AT) G Singley
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

OSD
Attn OUSD(A&T)/ODDDR&E(R) R Trew
Washington DC 20301-7100

AMCOM MRDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

Army Rsrch Ofc
Attn AMXRO-GS Bach
PO Box 12211
Research Triangle Park NC 27709

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

Dept of the Army (OASA) RDA
Attn SARD-PT R Saunders
103 Army
Washington DC 20301-0103

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Prsnnl
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20301-0300

Dpty Assist Secy for Rsrch & Techl
Attn SARD-TT F Milton Rm 3E479
The Pentagon
Washington DC 20301-0103

Hdqtrs Dept of the Army
Attn DAMO-FDT D Schmidt
400 Army Pentagon Rm 3C514
Washington DC 20301-0460

US Army CECOM RDEC Night Vision & Elec Sensors Dir
Attn AMSEL-RD-NV-VISPD C Hoover
Ste 430
Attn AMSRL-RD-NV-UAB C Walters
Ste 4010221 Burbeck Rd,
FT Belvoir VA 22060-5806

US Army Edgewood Rsrch, Dev, & Engrg Ctr
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmnd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Natick Rsrch, Dev, & Engrg Ctr
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

US Army Rsrch Ofc
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmnd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automtv & Armaments Cmnd
Attn AMSTA-AR-TD M Fisette
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Tank-Automtv Cmnd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

Distribution (cont'd)

US Army Test & Eval Cmnd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Army Train & Doctrine Cmnd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

US Military Academy
Dept of Mathematical Sci
Attn MAJ M D Phillips
West Point NY 10996

Nav Surface Warfare Ctr
Attn Code B07 J Pennella
17320 Dahlgren Rd Bldg 1470 Rm 1101
Dahlgren VA 22448-5100

DARPA
Attn B Kaspar
3701 N Fairfax Dr
Arlington VA 22203-1714

ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
University of Texas
Austin TX 78712

College of Staten Island/CUNY
Dept of Engrg Sci & Physics
Attn S Rizvi
2800 Victory Blvd Rm 1N-223
Staten Island NY 10314

Univ of Maryland Dept of Elec Engrg
Attn Q Zheng
Attn R Chellappa
A V Williams Bldg Rm 2365
College Park MD 20742-3285

ERIM
Attn C Dwan
Attn D Zhao
Attn J Ackenhusen
Attn Q Holmes
1975 Green Rd
Ann Arbor MI 48105

Natl Inst Standards/Tech
Attn J Phillips
Bldg 225 Rm A216
Gaithersburg MD 20899

Palisades Inst for Rsrch Svc Inc
Attn E Carr
1745 Jefferson Davis Hwy Ste 500
Arlington VA 22202-3402

Sanders Lockheed Martin Co
Attn PTP2-A001 K Damour
PO Box 868
Nashua NH 03061-0868

US Army Rsrch Lab
Attn AMSRL-D J Lyons
Attn AMSRL-DD J Rocchio
Attn AMSRL-CI-LL Techl Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-SE J M Miller
Attn AMSRL-SE J Pellegrino
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE H Kwon
Attn AMSRL-SE-SE H Moon
Attn AMSRL-SE-SE L A Chan
Attn AMSRL-SE-SE L Bennett
Attn AMSRL-SE-SE L-C Wang
Attn AMSRL-SE-SE M Lander
Attn AMSRL-SE-SE M Venkatraman
Attn AMSRL-SE-SE M Vrabel
Attn AMSRL-SE-SE N Nasrabadi
Attn AMSRL-SE-SE P Rauss
Attn AMSRL-SE-SE S Der (5 copies)
Attn AMSRL-SE-SE T Kipp
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1998	Interim, 10/96 to present

4. TITLE AND SUBTITLE Composite Classifiers for Automatic Target Recognition

5. FUNDING NUMBERS

PE: 61102A

6. AUTHOR(S) Lin-Cheng Wang, Sandor Der, and Nasser M. Nasrabadi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE email: sder@arl.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1661

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 7NE0M1
AMS code: 611102.305

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Composite classifiers that are constructed by combining a number of component classifiers have been designed and evaluated on the problem of automatic target recognition (ATR) using forward-looking infrared (FLIR) imagery. Two existing classifiers, one based on learning vector quantization and the other on modular neural networks, are used as the building blocks for our composite classifiers. We analyze a number of classifier fusion algorithms, which combine the outputs of all the component classifiers, and classifier selection algorithms, which use a cascade architecture that relies on a subset of the component classifiers. Each composite classifier is implemented and tested on a large data set of real FLIR images. The performances of the proposed composite classifiers are compared based on their classification ability and computational complexity. We demonstrate that the composite classifier based on a cascade architecture greatly reduces computational complexity, with a statistically insignificant decrease in performance in comparison to standard classifier fusion algorithms.

14. SUBJECT TERMS
Composite classifiers, ATR neural nets, FLIR, learning decomposition, object recognition, classifier fusion

15. NUMBER OF PAGES
36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102