Neural Networks for Automatic Target Recognition

STEVEN K. ROGERS, JOHN M. COLOMBI, CURTIS E. MARTIN, JAMES C. GAINNEY, KEN H. FIELDING, TOM J. BURNS, DENNIS W. RUCK, MATTHEW KABRISKY AND MARK OXLEY

Air Force Institute of Technology

(Received and accepted 13 December 1994)

Abstract—Many applications reported in artificial neural networks are associated with military problems. This paper reviews concepts associated with the processing of military data to find and recognize targets—automatic target recognition (ATR). A general-purpose automatic target recognition system does not exist. The work presented here is demonstrated on military data, but it can only be considered proof of principle until systems are fielded and proven "under fire". ATR data can be in the form of non-imaging one-dimensional sensor returns, such as ultra-high range-resolution radar returns for air-to-air automatic target recognition and vibration signatures from a laser radar for recognition of ground targets. The ATR data can be two-dimensional images. The most common ATR images are infrared, but current systems must also deal with synthetic aperture radar images. Finally, the data can be three-dimensional, such as sequences of multiple exposures taken over time from a nonstationary world. Targets move, as do sensors, and that movement can be exploited by the ATR. Hyperspectral data, which are views of the same piece of the world looking at different spectral bands, is another example of multiple image data; the third dimension is now wavelength and not time. ATR system design usually consists of four stages. The first stage is to select the sensor or sensors to produce the target measurements. The next stage is the preprocessing of the data and the location of regions of interest within the data (segmentation). The human retina is a ruthless preprocessor. Physiology motivated preprocessing and segmentation is demonstrated along with supervised and unsupervised artificial neural segmentation techniques. The third design step is feature extraction and selection: the extraction of a set of numbers which characterize regions of the data. The last step is the processing of the features for decision making (classification). The area of classification is where most ATR related neural network research has been accomplished. The relation of neural classifiers to Bayesian techniques is emphasized along with the more recent use of feature sequences to enhance classification.

The principal theme of this paper is that artificial neural networks have proven to be an interesting and useful alternate processing strategy. Artificial neural techniques, however, are not magical solutions with mystical abilities that work without good engineering. Good understanding of the capabilities and limitations of neural techniques is required to apply them productively to ATR problems.

Keywords—Artificial neural networks, Automatic target recognition, Segmentation, Feature selection, Classification, Pattern recognition.

1. INTRODUCTION

Artificial neural network models have played a significant role in the development of military automatic target recognition (ATR) systems. The most common model used in any neural network application, including ATR, is based on the perceptron architecture (Rosenblatt, 1959; Widrow & Hoff, 1960). A weighted linear combination of inputs is passed through a nonlinearity to model the number of action potential pulses coming out of a biological neuron in response to stimulation. Backpropagation, the general learning algorithm developed in the 1970s and popularized in the early 1980s (Werbos, 1974; Rummelhart & McClelland, 1986), enabled the development of multilayer perceptrons. These multilayer perceptron networks can be used for finding regions of interest, extracting features, selecting features, or classification. They may not, however, be the best neural technique for a given problem.

This paper details the design of ATR systems, illustrating the wide array of options, both neural and
conventional, available to the system designer. The outline of the article is depicted in Figure 1. The next section introduces the data sets that are used for testing the various ATR systems described here. Then preprocessing and segmentation is addressed. The initial preprocessing of the data can be inspired by physiological systems. After segmentation, the problem of feature extraction and selection is considered. While feature extraction can never add information (Fukunaga, 1990), it may provide an increase in signal-to-noise. Several neural feature selection algorithms are presented. In the final section, the step of classifier design is considered. A neural and conventional baseline Bayes analysis of classification is shown on some high range-resolution radar data, and then several ATR systems are described to show the large number of options available at this step. These systems include classifiers which use feature sequences, including hidden Markov models (HMM), recurrent perceptron networks, adaptive time delay neural networks (ATDNN), and hidden control neural networks (HCNN).

2. ATR DATA SETS

Perhaps the first choice in the design of an ATR system is what sensor system to use to produce the measurements from which the targets may be recognized. Although there are often many factors which govern this choice, one of the important considerations is the ability to separate the classes in the measurement space; the design steps of segmentation, feature extraction, and classification may be greatly simplified by the use of an appropriate sensor system. The results presented in this paper were obtained using data produced by a wide range of different sensor systems, including both imaging and non-imaging sensors. This section describes the various types of data considered. These include ultra-high range-resolution radar (UHRR) range profiles, vibration signatures measured by lasar radar, forward-looking infrared (FLIR) and synthetic aperture radar (SAR) imagery, and sequences of FLIR imagery, as well as hyperspectral data.

2.1. UHRR

UHRR uses a wide-band radar to produce a profile of the electromagnetic energy returned as a function of distance along the line of sight of the radar. This profile is obtained by sending a radar pulse toward the target. As the pulse interacts with scattering locations on the target (such as engine inlets and leading or trailing edges of an airborne target), energy is returned from the target to the radar detector. This return signal is plotted as a function of time delay or distance to produce the range profile. Figure 2 shows this acquisition process and shows a simple range profile.

The locations of the peaks in a range profile can be directly related to the physical distances (along the range direction) to scattering centers on the target. Since the scattering centers have a fixed geometrical relationship to each other, the down-range distances between them change as the direction of observation changes. Thus, the range profiles produced by UHRR radar are highly dependent on aspect angle.

2.2. Vibration Signatures

Another non-imaging signal for ATR applications measures the vibration of the target using a continuous wave (CW) fixed-frequency lasar radar (LADAR) or millimeter wave (MMW) radar. The surface vibration is measured by sensing the changes in the frequency of the CW signal as the transmitted
signal is modified by the vibrating surface. The Doppler shift caused by the radial velocity of the vibrating surface provides the changes in frequency. Figure 3 provides a model of the interaction between a frequency-stable CW LADAR and a planar vibrating surface. Figure 4 shows a Welsh periodogram (60–1024 point DFTs) of the vibration signature of six vehicles measured at the same aspect and engine speed. These signals are also aspect angle dependent. Classification results obtained using vibration signatures are given in Guerts et al. (1992).

2.3. Image and Image Sequences

Perhaps the most frequently considered ATR systems are based on data from imaging systems. Forward-looking infrared (FLIR) imagery is the most common ATR data source. Synthetic aperture radar is another popular single-image ATR sensor. Because the phenomenology of these sensors is rather well-understood, it is not discussed here. Sample images from each sensor are shown in Figures 5 and 6.

An alternative to using single image frames in an ATR system is to use multiple images, taken as the
target moves relative to the observer, as depicted in Figure 7. The transitions between images in a sequence provide additional information to separate different target classes. Although this concept is demonstrated here using FLIR images, it has also been exploited with both UHRR and SAR data.

Hyperspectral data represents another example of data that is not simply two-dimensional (2-D) which could be used in developing an ATR system. In hyperspectral data, each pixel in an "image" may be represented by a vector of values, rather than a simple gray-scale value. One hyperspectral sensor, the airborne visual and infrared imaging spectrometer (AVIRIS), produces 224 coregistered images of a geographic location, where each image is obtained in a different spectral band as shown in Figure 8.
3. PREPROCESSING AND SEGMENTATION

Preprocessing and segmentation may be the toughest task in computer vision and ATR. Preprocessing is frequently required because ATR data have characteristically low contrast and high clutter. Much of the necessary preprocessing can be accomplished using models of the retina. In the late 1980s, Carver Mead’s group at Cal Tech designed a silicon retina that emulates early vision processes (Mead & Mahowald, 1988; Mahowald & Mead, 1991). Their highly successful experimental microchip models the initial three neural layers of the vertebrate retina. In these layers, photonic energy is transformed into neural potentials. The representation produced by these three layers has been demonstrated to discount the illuminance, suppress noise, and compress the dynamic range. Spatial discontinuities or edges (high spatial frequencies) are also enhanced. This hardware concept has been used as a smart focal plane processor for infrared missile seekers. Prototypes which emphasize physiological “smart” focal plane processing have been demonstrated to achieve retina-like performance (King & Massie, 1994).

Segmentation may be done on a single image frame, or target motion may be incorporated to enhance the segmentation effectiveness. This section contains five subsections. First, a scanning window neural architecture for segmentation of stationary targets is presented. Second, the scanning window concept is incorporated in a hybrid wavelet/radial basis function (RBF) network segmenter. Third, the segmentation of hyperspectral data is presented. The fourth and fifth subsections present retinal and retinal/cortical models which utilize relative target motion to segment targets from multiframe image sequences.

3.1. Scanning Supervised Learning Segmentation

The tracking of stationary targets through multiple image frames taken from a moving platform can be accomplished using a scanning receptive field neural network with a back-end neural network classifier (Tarr, 1991). Figure 9 shows a diagram of this network structure, and Figure 10 compares the FLIR image segmentation achieved by this structure with that obtained by a standard thresholding technique. The advantage of the neural technique is that variation of an arbitrary threshold is not required.

Although the quality of any segmentation technique is somewhat subjective (Harrup, 1994), the final objective of classification or tracking can be used to determine the segmentation quality. The goal of the segmenter in this application is to provide data of sufficient quality to maintain aim points on the target. Figure 11 shows the performance of the scanning window architecture through a terminal guidance sequence. The network was trained to recognize regions of interest using representative patterns selected by an operator from a single frame taken early in the sequence. Once trained, the system successfully segmented and tracked the objects of interest for the rest of the terminal guidance sequence, as shown in the figure.
3.2. Scanning Window SAR Segmentation using Wavelets

The scanning neural window architecture can be used to combine multiresolution analysis (wavelets) with RBF neural networks for SAR segmentation. In this section, the ability to automatically distinguish between trees, fields, and SAR shadows is demonstrated. This is accomplished by training an RBF network to classify scanning windows of the wavelet decomposition of the image.

3.2.1. Multiresolution Analysis. Wavelet analysis provides a method of identifying the spatial frequency content of an image and the local regions within the image where those spatial frequencies exist. Since increasing window lengths correspond to successively coarser resolutions (in time or space), wavelet analysis is sometimes referred to as multiresolution analysis. Wavelets have been shown to closely represent the response profile of neurons in striate cortex (Jones & Palmer, 1987), which provides additional physiological motivation for their use.

The multiresolution representation can be viewed as passing an image through a sequence of low-pass filters with successively narrower bandwidths. This approximation process results in a set of successively lower resolution representations of the original image. The information lost at each stage of the smoothing process is retained in a detail signal (Mallat, 1989). Table 1 shows the frequency content of the various resolution levels corresponding to a 2048 × 2048 original image. The highest possible frequency in a 2048 × 2048 image is 1024 cycles/
### TABLE 1
SAR Frequency Content for Varying Multiresolution Levels

<table>
<thead>
<tr>
<th>Level</th>
<th>Detail Signal</th>
<th>Approximation Signal</th>
<th>Smallest Detectable Change (ft)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cycles/Object</td>
<td>Cycles/Object</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>N/A</td>
<td>0–1024</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1024–512</td>
<td>0–512</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>512–256</td>
<td>0–256</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>256–128</td>
<td>0–128</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>128–64</td>
<td>0–64</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>64–32</td>
<td>0–32</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>32–16</td>
<td>0–16</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>16–8</td>
<td>0–8</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>8–4</td>
<td>0–4</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>4–2</td>
<td>0–2</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>2–1</td>
<td>0–1</td>
<td>1024</td>
</tr>
</tbody>
</table>

image, which corresponds to an object which is 2 feet (2 pixels) wide. One of the great advantages of SAR is that the resolution is independent of the range, since a range-angle image is formed as opposed to an angle-angle image. The resulting “God’s eye” view allows the correct scale of the wavelet multiresolution analysis to be chosen for a particular ATR task.

3.2.2. Wavelet/RBF Image Segmentation. With the wavelet decomposition of an image, an RBF network can be trained to segment regions of interest. The RBF architecture is used for its complex receptive field covering of the input space. In addition, any multivariate function can be reasonably approximated using a linear combination of radial basis functions with their centers on or near data points (Poggio & Girosi, 1990). The segmentation is accomplished by choosing a wavelet decomposition level based on the size and frequency content of the objects or regions of interest, and then partitioning the image by centering small overlapping receptive fields about each coefficient. The overlapping receptive field dictates the feature size for the RBF classifier.

Figure 12 shows the original image and segmentation of homogeneous regions as provided by the RBF outputs. Daubechies’ 2 and 6 compactly supported wavelets (Daubechies, 1988) and Mallat’s spline wavelet (Mallat, 1989) were used in the decomposition, and the segmentation was done at level 4 with a 3 × 3 receptive field. Notice that small regions of trees in the original image are correctly segmented and identified.

3.3. Unsupervised Hyperspectral Segmentation

Hyperspectral data is another example of a multi-frame imaging modality. However, unlike the scanning window architecture presented earlier, the segmentation presented here uses all the frames.
simultaneously. As stated earlier, the AVIRIS hyperspectral sensor produces 224 coregistered images of a geographic location; each image is taken in a different (roughly 10 nm wide) spectral band, as depicted in Figure 8. An alternative view is that each pixel outputs a 224-dimensional vector, representing the spectrum of the radiance from the piece of the world observed by that pixel. Segmentation of the AVIRIS "image" may be accomplished by grouping pixels together with a vector quantization scheme.

One vector quantization algorithm which has been used extensively in communications applications is the algorithm of Linde, Buzo, and Gray (LBG) (Linde et al. 1980). Figure 13 shows the results obtained by using the LBG algorithm to define 32 cluster centers, then assigning each pixel to a cluster using a minimum distance classifier. The advantage of this method is that very few assumptions are required to produce the cluster centers. A significant disadvantage is that it is often impossible to know a priori how many actual clusters exist in the data, and the clustering
algorithm will distribute the user-specified number of cluster centers throughout the data, whether or not that many clusters truly represent the data. This limitation is addressed by the topology representing network (TRN), proposed by Martinetz and Schulten (1994), which provides a means of using the training data to group clusters into "super-clusters".

Figure 14 shows the results obtained by using the TRN algorithm to group 64 LBG cluster centers of the AVIRIS data into 32 "super-clusters". Comparing this result with Figure 13, the
The histogram has only one peak, so machine estimation of the threshold would be difficult. Even with optimum threshold, by visual inspection, the results are poor.

The effectiveness of this method is most evident in the bay area, shown in the upper-left corner of both images. In the TRN-connected image, all the water pixels are grouped into one large cluster, while the LBG algorithm leaves the water pixels dispersed among four or five distinct clusters. The hybrid TRN/LBG approach provides a means for automatically locating pixels with similar spectral
FIGURE 11. Hydro-electric power plant tracking sequence produced by the scanning window neural network segmenter.

Accuracy: 100.00 Percent on target.
content which may then be used to isolate objects of interest in the hyperspectral “image”.

3.4. Retina Model

To this point, segmentation has been accomplished on “still” imagery. Even in the case of the fixed-target tracker, the segmentation was done on a frame by frame basis. In many military ATR applications, the objects to be segmented move through a sequence of frames, due to target motion, sensor motion, or both. In fact, the motion of the target through a sequence may make it easier to segment it out. This type of segmentation and preprocessing has been shown to occur in the human visual system. This subsection and the next present two models based on the human visual system which have been applied to motion segmentation. This subsection presents a retinal model that extends previous models, while the next subsection presents a new physiological model which uses 3-D wavelets and cooperative and competitive cells between scales (Jones & Palmer, 1987; Burns et al., 1994).

Motion segmentation is the process of removing all stationary objects from spatio-temporal inputs (imagery) and passing only the transient aspects of the scene. In the human eye, the early vision information is passed to the final two retinal layers (amacrine and ganglion cells), where segmentation of moving elements in the field of view begins. These layers are sensitive to transient aspects of the scene and formulate the retina’s output channels. The retina passes its coded data through the optic chiasm to the lateral geniculate nucleus where it is further
processed. One visual pathway leads to the striate or primary visual cortex. Thereafter the information is processed according to form, color, and motion (Zeki, 1994).

Öğmen & Gagné (1990) have modeled the fly's visual perception of motion. Their paper presented the fly model and gave a treatment of the synchronous and asynchronous forms of the non-recurrent cell activity. Their formal definition of the neurotransmitter and cell activity formulates a two-layer neural network, shown in Figure 15. The nodes in the input array (the synapse) are characterized by the neurotransmitter equation, and connected to a smaller output array of nodes (the soma) which are characterized by the cell activity equation.

Two extensions of Öğmen and Gagné's (O&G) model are discussed here. First, the model is expanded to two dimensions by replicating the basic unit shown in Figure 15 (with one soma cell) in a 2-D array. Second, an adaptive model is created by adding long-term memory to the original model. The neurotransmitter and cell activity
equations are the long- and short-term memory equations defined by Grossberg (1988), modified to allow for an asynchrony in the opposing receptive regions of the neuron (Öğmen & Gagné, 1990). This asynchrony has a natural tendency to dampen tonic elements of the input field while enhancing phasic elements (Öğmen & Gagné, 1990; Swanson, 1992).

These models were tested using FLIR imagery sequences, a sample of which is shown in Figure 7. In the sequence, a tank moves initially from right to left against a scintillating background, then turns towards the camera and moves to the right where it leaves the frame. In the last three frames, a plume of exhaust can be seen traveling to the left. Figures 16 and 17 illustrate the same frames after processing through the extended O&G and adaptive models. The background is removed in both figures by the models and neither show the exhaust plume.

The extended O&G model shows more of the tank’s detail while the adaptive model picks out the leading edge of the moving tank (treads and front of...
gun barrel for example). The results indicate that the adaptive model with added long-term memory achieves better performance than the original O&G model. It removes stationary information better than the original O&G model and produces a heightened response to leading edges. The direction of motion is clearly identified and stationary, constant intensity objects are completely removed (Swanson et al., 1993).

3.5. Retina/Cortical Wavelet Model

Past and ongoing research clearly points toward the existence of a biological motion detection system in mammals that responds to localized spatial and temporal frequency stimuli (Watson & Ahumada, 1983; Adelson & Bergen, 1984; Adelson & Moshvon, 1982; McKee, 1981; Robson, 1966). This subsection presents a retina/cortical model for velocity estima-
tion (motion segmentation) which uses nonhomogeneous 3D wavelet analysis (Burns, 1996) to model this localized spatial and temporal frequency response, and a cooperative/competitive methodology to fuse the information from the various scales.

Figure 18a shows four frames of a 64 × 64 × 64 simulated image sequence in which a BRL-CAD tank moves across the field of view at a constant velocity, passing in front of the tree on the right and partially behind the tree on the left. Figure 18b shows a flow map obtained by decomposing the sequence with a 3D wavelet filter constructed from a 23 tap truncated cubic spline in space and a compact Daubechies 4 tap filter in time. A smaller order temporal filter was used to generate more decomposition levels in time. This increases the number of "speed bins" in the directionally selective motion-oriented multiresolution wavelet filter bank, enhancing the velocity estimation capabilities of the system. Erroneous flow vectors are primarily attributable to pixel scintillations on the tank's surface which violate a constant intensity assumption of the algorithm.

Using a methodology that combines a modified version of Grossberg's gated dipole filter with a cooperative-competitive strategy (Grossberg & Mingolla, 1987), these erroneous flow vectors are eliminated. This methodology is represented graphically in Figure 19. In the cooperative stage, the vector orientations are discretized, and the flow map is decomposed into a set of oriented flow maps by placing all the vectors of each orientation into a separate map. Each orientated flow map is then filtered with an identically oriented modified gated dipole filter to boost vectors which lie in groups along the same orientation as the map. The magnitude of the gated dipole filter response is computed and recorded at each location in every oriented flow map. Then the composite flow map is reconstructed in the competitive stage. Under a winner-take-all rule, the vector at each location in the composite map is assigned the orientation of the oriented map with the highest filter response at that location. The magnitudes are then determined by computing the average magnitude of the non-zero flow vectors contained under the gated dipole filter in the winning oriented flow map.

The wavelet/gated dipole filter velocity estimation system performs well in the presence of modest amounts of additive Gaussian noise. Figure 20 shows four image sequences obtained by adding increasing amounts of Gaussian noise to the sequence shown in Figure 18. The same 3D wavelet filter described earlier was used to generate a flow map for each of the sequences, and the cooperative-competitive flow restoration technique was applied to the resulting flow maps. Figure 21 shows the results of this process. Notice in the top flow map (corresponding to the original "noiseless" sequence) that the
4. FEATURE EXTRACTION, SALIENCY, AND SELECTION

The extraction of a set of numbers to accomplish ATR using neural networks and the subsequent selection of the discriminantly relevant subset of those features is addressed in this section. Early work in neural networks suffered from criticisms that it was often not apparent why a particular decision was made. Hypothetically for ATR, questions may be raised regarding the most important features which resulted in classification of critical targets. One obvious reason for reduced feature sets is the timeliness of model training from smaller numbers of parameters. In addition, the curse of dimensionality states that the number of training patterns grows exponentially as fast as the number of features. Others have also related the impact on required training exemplars to the dimensionality of the input features (Foley, 1972; Cover, 1965; Baum & Haussler, 1989).

Feature extraction can make use of shape information of the segmented image or it may transform or project the raw data into a smaller dimension. Representative FLIR shape features are
provided in Table 2. Other features include high order moments. The Karhunen-Loeve transformation (KLT) or principal components analysis is an optimal method of determining the minimum number of linear combinations of the input data to permit accurate reconstruction of the data. Though reconstruction is not the goal, KLT and the neural extensions may project data into a space with better class distinguishing properties. Lee and Landgrebe (1993) propose a transformation based on decision-boundaries that AFIT has extended for use with neural networks. Other spectral transformations, such as discrete cosine, low frequency Fourier and linear predictive coding have demonstrated success on ATR problems.

4.1. Saliency

Saliency will be defined as a measure of a feature's ability to impact classification. Saliency will be shown an effective tool on selecting features for FLIR ATR. A benchmark to compare feature saliency methods is the single probability of error criterion, denoted $P_e$. This technique computes the probability of error separately for each individual feature. These error rates can be rank ordered to choose an appropriate size subset.

Le Cun proposed a saliency metric which made use of the second derivative of the MLP outputs with respect to the input features (Le Cun et al., 1990). A diagonal Hessian matrix assumption was made to define the saliency metric, $\tilde{s}_i$, for each feature.
averaged over all training exemplars. Another saliency metric for feature $i$, $\Lambda_i$, measures the effect of the MLP outputs, sampled over an appropriate range of allowable input values (Ruck, 1990). Empirical results indicate that $\Lambda_i$ provides similar rankings to $P_e$ (Ruck et al., 1990a, p. 46).

A relationship exists between the class specific Bayesian probability of error and the derivative-based feature saliency metric, $\lambda_i$ (Priddy et al., 1993). The relationship relies on the assumptions necessary for feedforward neural networks to approximate a Bayes optimal discriminant function (Kanaya & Miyake, 1991; Richard & Lippmann, 1991; Ruck et al., 1990b; Shoemaker, 1991; Wan, 1990; Lippmann, 1989, p. 50). When these assumptions are met, the trained feedforward neural network output can be interpreted in the limit as $P(\omega_k|x)$, which is the posterior probability of class $k$ for $x$.

A weight-based metric, $\Upsilon_i$, for measuring feature $i$'s saliency has also been developed (Tarr, 1991) as the sum of squared lower weights. Computationally, this metric is much simpler than the other available feature saliency metrics. The Minkowski$_1$ (taxi-cab) and Minkowski$_\infty$ (infinity) norms of the weights

FIGURE 20. Synthetic tank image sequence with four levels of additive Gaussian noise.
No noise

\[
\begin{align*}
\text{S/N} &= 10 \\
\text{S/N} &= 7 \\
\text{S/N} &= 5 \\
\text{S/N} &= 1
\end{align*}
\]

FIGURE 21. "Flow restored" optical flow maps from "noiseless" and noise-added four tank sequences.

provide similar feature rankings to \( \mathcal{T}_i \) weight saliency, which is defined using the squared Minkowski_2 norm (Stepp et al., 1994). These will be denoted by \( \mathcal{T}_i^l \) and \( \mathcal{T}_i^c \). Experimentally, \( \mathcal{T}_i \) provides rankings similar to \( \Lambda_i \) (Tarr, 1991, p. 49).

A method for reducing gradient calculations and

<table>
<thead>
<tr>
<th>Feature Number</th>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Length/width</td>
<td>Ratio of object length to width</td>
</tr>
<tr>
<td>2</td>
<td>Standard deviation</td>
<td>Standard deviation of pixel values on object</td>
</tr>
<tr>
<td>3</td>
<td>Maximum brightness</td>
<td>Maximum brightness on object</td>
</tr>
<tr>
<td>4</td>
<td>Compactness</td>
<td>Ratio of number of pixels on object to number of pixels in rectangle which bounds object</td>
</tr>
<tr>
<td>5</td>
<td>Complexity</td>
<td>Ratio of border pixel to total object pixels</td>
</tr>
<tr>
<td>6</td>
<td>Mean contrast</td>
<td>Contrast ratio of object's mean to local background mean</td>
</tr>
<tr>
<td>7</td>
<td>Contrast ratio</td>
<td>Contrast ratio of object's highest pixel to its lowest</td>
</tr>
<tr>
<td>8</td>
<td>Bright pixel ratio</td>
<td>Ratio of number of pixels on object within 10% of maximum brightness to total object pixels</td>
</tr>
<tr>
<td>9</td>
<td>Difference of means</td>
<td>Difference of object and local background means</td>
</tr>
</tbody>
</table>
approximating $\lambda_i$ is defined by sampling only the known data, $\lambda_i^{\text{data}}$. For this known data saliency, the network’s feature sensitivity is evaluated in a manner proportional to the total likelihood function of the data. Note that regions of maximum total likelihood may not correspond to regions of maximum feature saliency.

These measures were applied to FLIR imagery where targets consisted of tanks, trucks, and armored personnel carriers using the nine features given in Table 2. The saliency values are provided in Table 3. Details regarding these features for ATR can be found in (Roggemann, 1989; Roggemann et al., 1989a,b; Ruck, 1990).

### 4.2. Feature Selection Criterion

Statistical pattern recognition offers many methods related to feature subset selection (and the related dilemma of model order selection). Within statistical regression models, criterion can be defined for reduced parameter subsets, most relating MSE of the reduced model to the original model or to the actual samples (Steppe, 1994). Akaike information criterion (AIC) or final prediction error (FPE), often used in autoregressive modeling, provide criterion for model order selection (Kay, 1988).

Subset selection of features can be accomplished using a partial $F$-test, where the $F$ statistic is taken to be the ratio of MSE errors when a subset of the features is hypothesized to be zero (not used). Having this statistic allows one to grow subsets (forward selection) or contract subsets (backward selection). These statistical iterative methods require the final size feature dimension to be specified and suffer from the inability to add features once removed (Steppe, 1994; Stepp et al., 1994).

When using saliency, a rank ordering can be accomplished to remove unimportant features, or ones exhibiting small saliency. In Table 4, rankings of saliency values are shown for all metrics using the FLIR data set. A ranking of "1" indicates the best feature and a ranking of "9" indicates the worst feature. With the exception of the metric $\tilde{s}_i$, similar features receive similar rankings across all metrics. In Figure 22, the mean accuracy and a 95% confidence interval error band (plotted as horizontal bars about the means) are plotted for $\lambda_i^{\text{data}}$. Results are shown

![Figure 22](image-url)
for the $\hat{A}_{i}^{\text{data}}$ saliency. The results from all of the elevated metrics indicate it is possible to remove one to three features based on saliency metric rankings with little or no degradation in the average network error. However, further reduction of the feature set requires a trade-off in classification error.

### 5. CLASSIFICATION

Having received data from one of several sensors, applied suitable preprocessing and segmented regions of interest, then reduced these regions via projections and saliency, recognition of objects is the final step in the ATR process. These prior decisions which lead up to classification will impact the best results achievable. This last section describes bounds on this best performance and highlights some complete ATR systems, comparing both statistical and neural paradigms for target image and sequence recognition.

#### 5.1. Bayesian Analysis

The Bayes error rate has long been established as the fundamental statistical upper bound on the performance of any pattern classifier (Duda & Hart, 1973). This limit is achieved by the application of the Bayes, or maximum a posteriori (MAP), decision rule, which assigns each unknown sample to the class with the MAP probability, given the sample. In addition to providing a measure of the separability of the classes in a given problem, the Bayes error rate is a benchmark against which classifier performance may be judged.

Because the Bayes error rate is a measure of the best classification accuracy that may be obtained with a given set of features, it provides the benchmark against which any classification system based on those features may be judged, whether statistical or neural. It has been proven many times in the literature that the common neural techniques perform as approximators to the Bayes optimal discriminant functions (minimum probability of error) (Ruck et al., 1990b). This means there exist no statistical techniques which will outperform correctly engineered neural algorithms with respect to accuracy. Nevertheless, the comparison of the neural classification algorithms with statistical techniques is useful to ensure the neural technique is correctly engineered.

Before designing a classifier around a chosen set of features, the Bayes error rate may be computed (or estimated) to determine whether or not the features will yield adequate separation between the classes. In most practical cases, the Bayes error rate is estimated using a finite set of labeled samples from the various classes. This is typically done by estimating the $a \text{ posteriori}$ probabilities of each class for each sample, then assigning each sample to the class with the MAP probability estimate. The percentage of samples misclassified by applying the MAP decision rule to the $a \text{ posteriori}$ estimates is taken as an estimate of the Bayes error rate.

In estimating the expected error rate of any system, care must be taken in the use of the labeled samples to arrive at the estimate. Toussaint enumerates many different techniques of using a finite set of data to estimate the accuracy of a classifier (Toussaint, 1974). Two techniques of particular interest are referred to as resubstitution and leave-one-out. In resubstitution, all available samples are used in both the training set and the test set. In this technique the estimates of the $a \text{ posteriori}$ probabilities favor the true class for each sample, because the knowledge of the true class of each sample is included in the estimate. In leave-one-out, a separate classifier is designed for each sample, using all the available data except the one test sample. In this way, the class membership of each sample is not included in the $a \text{ posteriori}$ estimates for that sample. This technique has been characterized as "almost unbiased" (Lachenbruch, 1967).

On average, resubstitution and leave-one-out error estimates bound the Bayes error rate (Fukunaga, 1990, p. 220). Fukunaga and Hummels have demonstrated nonparametric statistical techniques for obtaining bounds on the Bayes error rate which use $a \text{ posteriori}$ estimates derived from Parzen and $k$-nearest neighbor ($k$-NN) density estimates (Fukunaga & Hummels, 1987). Using multilayer perceptron $a \text{ posteriori}$ estimates, a similar neural approach may be implemented (Martin et al., 1994). The resubstitution estimate is obtained by training a multilayer perceptron with all available labeled data, then testing its performance using the same set of data. The leave-one-out estimate is obtained by training a separate network for each labeled sample. Each network is trained using all the data except the one held-out sample, which is used to test the network. The percentage of samples misclassified by this procedure is taken as a leave-one-out estimate of

### TABLE 4.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$\hat{A}_{i}$</th>
<th>$\hat{A}_{i}^{\text{data}}$</th>
<th>$\hat{s}_{i}$</th>
<th>$T_{i}^{(1)}$</th>
<th>$T_{i}^{(2)}$</th>
<th>$T_{i}^{(\infty)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>3</td>
<td>9</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Neural Networks for ATR 1175

the Bayes error rate. This procedure is repeated with networks of various sizes, then the minimum $L$ error and the corresponding $R$ error are taken as the upper and lower bounds on the actual Bayes error rate.

Figure 23 shows plots obtained by applying the statistical (Parzen and $k$-NN) and neural network techniques to a UHRR target identification problem. The 35-dimensional feature vectors in this problem represent UHRR range profiles of two aircraft at various aspect angles. In the Parzen and $k$-NN techniques, the plots are generated by varying a parameter in the density function estimator: the window width $h$ in the Parzen case, and the number of neighbors $k$ in the $k$-NN case. The bounds on the Bayes error rate are taken as the minimum $L$ error rate, together with the corresponding $R$ error rate. In the neural approach, there is a tradeoff between accurately modeling the Bayes decision boundary and preventing the networks from memorizing the training data.

5.2. Single Look Classification System

Bayesian analysis applies to single-look classifiers where radar returns or single images require classification. FLIR images for target/nontarget classification are the most common data available. A system which uses histogram-based segmentation, followed by salient ranked feature selection is considered. Targets consisted of tanks, trucks and armored personnel carriers. Eight possible FLIR features were extracted for each “blob” with the best three selected using probability of error and saliency (Ruck, 1990). A comparison of an MLP classifier to both a one nearest neighbor and Parzen window classifier is provided in Table 5, demonstrating their statistical similarities—statistically equivalent at the 95% confidence level. Results are based on the leave-one-out method.

Classification using multiple sensors, multisensor fusion, can also be accomplished with neural or statistical techniques. Table 6 demonstrates the classification accuracies of using absolute range imagery (Roggemann, 1989) combined with FLIR. This approach separately segments each image, using a histogram-based approach for FLIR and a small-scale planarity mechanism for the range data (Roggemann, 1989). The additional range imagery consists of 32 bits per pixel range information, corresponding to distances of 860–1700 m for the

<table>
<thead>
<tr>
<th>Table 5. Estimated True Classification Accuracies of Two FLIR Feature Subsets (95% Confidence Interval is Shown)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classifier</strong></td>
</tr>
<tr>
<td>Bayesian with Parzen windows ($\sigma = 0.1$)</td>
</tr>
<tr>
<td>$k$-Nearest neighbor ($k = 1$)</td>
</tr>
<tr>
<td>Multilayer perceptron (3-5-2, $\eta = 0.3$, $\alpha = 0.8$)</td>
</tr>
</tbody>
</table>

FIGURE 23. $R$ and $L$ error estimates of the Bayes error rate for ultra high-range radar target identification problem. The asterisks indicate minimum average $L$ error and corresponding average $R$ error.
training and test samples. Again, these results show no significant difference between classifiers.

### 5.3. Sequence Classification Systems

The Bayes bounding by Fukunaga (1990) and the neural extensions (Martin, 1993), assumed the training and test patterns were independent in time. Duda and Hart review the extensions of Bayes decision theory to sequences (Duda & Hart, 1973, pp. 34–36). Two ways for classification of sequential patterns is to make decisions as each feature provided, *sequential decision theory*, or wait until some $T$ vectors have passed before classifying all $T$, *compound decision theory*.

Application of sequence processing for ATR dates back some 15 years. Le Chevalier was among the first to examine the structure in sequences of radar returns (Le Chevalier, et al., 1978). The key point of this research was that aspect angle information for aircraft targets produced by a tracker could be fused with radar cross-section measurements for improved identification. Thus, evolutionary constraints from kinematic sources can be combined with non-kinematic features to classify sequences of returns. Further developments can be found in Waxman et al. (1993); Libby (1993); and Libby & Maybeck (1993). This latter work exploits this kinematic and nonkinematic joint density for improved dynamic target recognition by incorporation of dynamic programming and Kalman filter trackers.

Since many ATR systems can acquire a series of imagery, the classification of objects within these images should extract information present in the entire series. Given a Markov dependency in the input sequence, a pattern recognizer designed for this dependency will have greater or equal classification accuracy than single look methods (Fielding, 1994). This increase in accuracy occurs since the entropy of the sequence is less than the entropy of the individual frames, and a reduction in entropy is related to an increase in information. Hidden Markov models have recently been shown to be an effective ATR algorithm by taking advantage of the reduction of uncertainty in processing a data sequence (Fielding, 1994; Fielding & Ruck, 1994). Neural methods such as recurrent neural networks, time delay neural networks and predictive neural architectures are presented for time series modeling and classification.

#### 5.3.1. Hidden Markov Modelling

Hidden Markov Model techniques have been used extensively in the area of speech recognition over the past 15 years and have become the technique of choice among researchers because of their ability to successfully learn the time varying characteristics of the spoken word. The reader is referred to the excellent reviews of the Markov process and its extension to the hidden Markov modeling that are given by Rabiner (1989), Rabiner and Juang (1986), and Poritz (1988). For a multiple class problem, decisions are based on arg max $P(X/\lambda_i)$ shown in Figure 24, where $X$ is the $T$ observation and $\lambda_i$ is the $i$th trained hidden Markov model. The hidden Markov model is trained by updating the model parameters using the Baum–Welsh re-estimation algorithm, a hill climbing maximum likelihood procedure. This method iteratively updates $\lambda$ to increase $P(X/\lambda_i)$.

To demonstrate the pattern identification capability of the hidden Markov model classification technique, a five class problem for identifying the tactical military ground vehicles shown in Figure 25 was examined (Fielding, 1994; Fielding & Ruck, 1994). The imagery is $256 \times 256$ pixel 8-bit grayscale generated using a constructive solid geometry-based computer-aided design package known as BRL-CAD (U.S. Army, 1991). Both discrete and continuous left-right and ergodic model architectures were investigated. Results using the continuous left–right hidden Markov model are discussed here. The left–right hidden Markov model only allows self- or forward-state transitions and requires multiple sequences from the observed process for training. The left–right sequence data set for each vehicle consists of 200 randomly generated sequences constrained to the area defined by azimuth angles $0–180^\circ$ and elevation angles $0–75^\circ$. Each sequence started at a low azimuth
angle (front view) and moved toward high azimuth angles as shown in Figure 26.

The object features used in the investigation are 28 low-frequency Fourier magnitude coefficients taken from a $7 \times 4$ rectangle. Each feature vector component is statistically normalized over all classes. The hidden Markov model was compared to a single look classifier, the one nearest neighbor algorithm-(1-NN), as well as a multiple frame 1-NN classifier. A classification decision is made at the sequence level according to which class wins a plurality of the frames for a given sequence. The error rate for each classifier is determined using the leave-one-out method. To further test the robustness of the hidden Markov model technique, classification rates of image sequences to which varying degrees of correlated Gaussian noise is investigated. Image sequences are generated for signal-to-noise ratios (SNR) of 20, 15, 10, and 5. The correlation level of the noise used here is 2 pixels. The correlation level refers to the full width of the autocorrelation of the noise at the half maximum amplitude point. A more extensive evaluation of SNR and correlation level effects is found in Fielding (1994). Table 7 illustrates the performance comparison between the continuous left–right hidden Markov model and the single-look and multiple frame nearest neighbor classifiers for the four SNRs at a correlation level of 2.

5.3.2. Recurrent Neural Networks. Recurrent neural networks (RNN) can perform as associative memories (Tank & Hopfield, 1987), time-series predictors,
sequence generators (Pineda, 1989; Pearlmutter, 1989) and models of finite state grammars (Elman, 1990; Servan-Schreiber et al., 1991; Cleeremans et al., 1989). Most recently, these architectures have been proven to be approximations to models of both discrete and continuous time-invariant nonlinear dynamic systems (Seidl & Lorenz, 1991; Funahashi & Nakamura, 1993). Their methods of training (Pearlmutter, 1990; Pineda, 1987; Zipser, 1989; Williams & Zipser, 1989) and their theoretical convergence and stability characteristics have been refined continually. Several researchers have demon-
strated relations to hidden Markov models. Bridle (1990) experimented with recurrent architectures which propagate forward probabilities in time. Keihagies (1991a,b) used a recurrent network for prediction and classification of time series trained by a method based on the Baum forward–backward algorithm used for hidden Markov models. Likewise, Niles and Silverman (1990), Hochberg et al. (1991) have also cast an hidden Markov model within a recurrent architecture.

Real time recurrent learning (RTRL) (Williams & Zipser, 1989) was successfully applied to a five class aircraft identification problem. Other feedforward approaches to classification of HRR sequences have been demonstrated by Brown et al. (1990) and Farhat (1989). Simulated high range resolution radar signatures, corrupted by 5 dBsm scintillation noise, were generated using the RCS_TOOLBOX (Bramley et al., 1991) for the Mig-21, F-4, F-15, F-16 and F-18 aircraft. Estimates of the target azimuth and width, along with amplitudes from the polarized range bins were used as features to a fully recurrent network.

Results are for 2100 test sequences, evaluated at elevation offsets between $-5.0^\circ$ and $5.0^\circ$ (increments of $1^\circ$) and azimuth offsets between $0^\circ$ and $180^\circ$ also in increments of $1^\circ$. Azimuth estimates biases were randomly generated to match uncertainty of $\pm 5^\circ$ as quoted for typical radar tracking algorithms (Libby, 1993). The classification accuracy increased monotonically with increasing number of frames. After 10 frames, results using six features (4 peak range bins, azimuth and target width estimates) are 88.7–91.3 “right” and 95.2–96.8 “good” at the 95% confidence level. “Right” refers to all class actual outputs are within 20% of desired, while “good” refers to a correct classification for the maximum output.

While recurrent architectures may be capable of modeling a very large class of dynamic systems, the alternative is to present time spatially using delay elements. Such a time-delay approach (TDNN) has been demonstrated successfully for speech phonetic classification (Lang & Waibel, 1990; Waibel et al., 1989).

5.3.3. Adaptive Time-Delay Neural Networks (ATDNN). TDNN handle both spatial and temporal information for both prediction and classification applications. A modification to the original TDNN algorithm presented by Wan (1993) as a finite impulse response (FIR) neural network won the Sante Fe Institute’s 1992 time series prediction competition. This algorithm, herein called the adaptive time-delay neural network (ATDNN), derives a temporal generalization of the familiar error backpropagation algorithm for learning optimum time delays for each weight as well as synaptic weights themselves. ATDNN proved quite successful in predicting Mackey–Glass chaotic time series (Lin et al., 1993) as well as ballistic trajectories (Lin et al., 1992). The ATDNN algorithm, both theory and interpretation, is fully described in Gainey (1993).

The exploded portion of Figure 27 shows the interconnection scheme for ATDNN nodes including $n$ connections, each with a different time delay, between previous node $i$ and the next node $j$. Each connection contains its own time delay and associated weight. In their article, Lin et. al., (1992) call this interconnection scheme between nodes a “delay block” where $\tau_{jn}$ and $\omega_{jn}$ are the independent time delays and weights for each $n$th interconnection of the block.

The same data set shown in Figure 25 is used, where all interframe times are uniform around the target. A general ATR might desire an accurate classification system independent of the time map (i.e., the speed with which the recognizer platform approaches the target should not matter to the classification system). Assumptions for this application include that the dimensionality of the random walk generated data to be used here is relatively low compared to chaotic time series data. For that reason, the number of time delays should be small — on the order of 5 or less. Rules for bounding the temporal history for a process of this random walk type are not defined. Talkiens rule for similar chaotic data generated using a Lorentian attractor was shown by Stright (1994) to have an embedology dimension of 3–4 which equates, by Talkiens rule (2d+1, where $d$ is the embedology dimension), to nine past time samples required for accurate prediction. The data used here, though, are much less stochastic (i.e., of lower dimensionality) than Lorentian data.

Both TDNN and ATDNNs were trained and tested with several architectures. The different number of hidden nodes and time delays were trained separately, each as a separate network. Hidden nodes of 5, 10, 14, and 20 were examined with both 3 and 5 maximum time delays allowed. The 95% confidence intervals for the best network architecture (20 hidden nodes with a maximum of 5
time delays) is 75.2–82.4 for the TDNN and 75.4–82.6 for the ATDNN. Thus, the TDNN and ATDNN both achieved the same accuracy in this multilook (plurality of frames) classifier. However, a sample confusion matrix, Table 8, illustrates the ATDNN learned the class discriminant functions more quickly. Note the high confusion between the two wheeled vehicles — BTR60 and M35.

5.3.4. Neural Prediction and Markov Modeling. By embedology (Stright, 1994), a wide class of deterministic nonlinear time series can be modeled and predicted. In contrast, hidden Markov models offer a stochastic method to explicitly capture the changing “state”, by the assumption of an underlining first-order Markov state process. By combining these methods, nonlinear modelling and discrete state processes, a potentially better model is created where periods of nonlinearity are modeled within the Markov state chain. This model was first examined for speech recognition in Levin (1990, 1993).

Let the source be modeled by a nonlinear autoregressive process with an exogenous state variable, $c_t$,

$$x_{t+1} = F_{\omega, c_t}(x_t, \ldots, x_{t-p}) + e_t,$$

where the $\omega$ describes a set of learned weights describing the nonlinearity occurring during state $c_t$ and $e_t$ is Gaussian white noise, initially assumed zero mean with unit variance. This can easily be extended to generalized normal densities dependent on the current state. This function can be approximated by a feedforward MLP using $p$ past samples and a particular hidden state or control input, $c_t$. This model is shown in Figure 28.

While many neural prediction methods use MSE as the basis for classification, one may also perform a residual analysis. Similar to multiple model adaptive estimation (MMAE) used in Kalman filtering with unknown model parameters, examination of the residual statistics is performed and compared to expected statistics. Let the variable $\omega$ represent both a set parameter values and an object class. Classes or objects are, in an abstract sense, simply “sets of parameters” in some infinite dimensional parameter.

**TABLE 8.**

<table>
<thead>
<tr>
<th>ATDNN and TDNN Confusion Matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTR60</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>BTR60</td>
</tr>
<tr>
<td>M2</td>
</tr>
<tr>
<td>M35</td>
</tr>
<tr>
<td>M60</td>
</tr>
<tr>
<td>T62</td>
</tr>
<tr>
<td>TDNN</td>
</tr>
<tr>
<td>M2</td>
</tr>
<tr>
<td>M35</td>
</tr>
<tr>
<td>M60</td>
</tr>
<tr>
<td>T62</td>
</tr>
</tbody>
</table>
Parameter identification using sets of Kalman filters, or multiple model estimation, is accomplished by analyzing the residual sequences from the respective filters. For a linear filter with the classical Gaussian noise assumptions and proper model parameters \( \omega \), the residual sequence over time can be shown to be zero mean, white (independent in time), and Gaussian, with covariance defined by the measurement function, the measurement noise covariance and the current state covariance. The utility of residual analysis is that measurements from a system with parameters \( \omega_j \), processed by a filter designed for parameters \( \omega_i \neq \omega_j \), will not in general give the proper mean and covariance. This deviation should produce a correct classification. Using Bayes' rule, we can find an \textit{a posteriori} probability of the parameter set given the observation sequence.

\[
p(\omega | x_t, \ldots, x_1) = \frac{\prod_{m=1}^M p(x_{t-1}, \ldots, x_1, \omega_j) p(\omega_j)}{\sum_{j=1}^M \prod_{m=1}^M p(x_{t-1}, \ldots, x_1, \omega_j) p(\omega_j)}, \tag{2}
\]

for \( i = 1, 2, \ldots, M \) classes. An iterative version of the above expression is given in Maybeck (1979). Based on the predictive HCNN model, the likelihoods, \( p(x_t | x_{t-1}, \ldots, x_1, \omega_i) \) are normal, derived from the residual statistics of the training data. This method would allow incremental classification.

Experimentation using minimum MSE was performed on the five class military land vehicle data set, as shown in Figure 25. Three and five state left-right models were examined with varying prediction orders and hidden dimensions.

### 6. CONCLUSIONS

This paper has demonstrated the application of physiologically motivated processing and artificial neural networks to automatic target recognition. Where appropriate, it related non-neural techniques. Referring back to Figure 1, many alternatives exist and their choices affect final classification performance. An array of sensors have been reviewed as sources for generating many formats of ATR data: One-dimensional, images, image sequences, and hyperspectral. Segmentation of images to find regions of interests can be performed on single frames with feedforward MLP using scanning windows, unsupervised clustering or wavelet decomposition. Segmentation can also make use of motion, as evident within the flow diagram, using series of images. Both 3-D wavelets and cooperative-competitive strategies can be used to resolve motion ambiguities. Once regions have been defined, features need to be extracted and saliency allows reduction of the input space, which further effects model and training data sizes. Lastly, classification of these salient features should incorporate object characteristics observed over several frames, whereby more information or less uncertainty is present in the sequence. Both neural and conventional pattern recognition methods attempt to estimate class discriminant functions, and the unknown Bayes error rate will be best that can be expected. The true proof of any ATR method is ultimately obtained through field tests in real environments.

### TABLE 9.

<table>
<thead>
<tr>
<th>Control States</th>
<th>Past Samples</th>
<th>Hidden Nodes</th>
<th>Noise (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>20</td>
<td>99.2–100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>99.2–100.0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>20</td>
<td>99.2–100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>99.2–100.0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>20</td>
<td>99.2–100.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>99.2–100.0</td>
</tr>
</tbody>
</table>
REFERENCES


Toussaint, G. T. (1974). Bibliography on estimation of misclassi-


