A Multi-Stage Neural Network for Automatic Target Detection

Ayanna Howard, Curtis Padgett, Carl Christian Liebe
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, California 91109-8099

Abstract - Automatic Target Recognition (ATR) involves processing two-dimensional images for detecting, classifying, and tracking targets. The first stage in ATR is the detection process. This involves discrimination between target and non-target objects in a scene. In this paper, we shall discuss a novel approach which addresses the target detection process. This method extracts relevant object features utilizing principal component analysis. These extracted features are then presented to a multi-stage neural network which allows an overall increase in detection rate, while decreasing the false positive alarm rate. We shall discuss the techniques involved and present some detection results that have been implemented on the multi-stage neural network.

Keywords - Automatic Target Recognition, Neural Network, Detection

I. Introduction

There has been much work involved in the process of automatic target recognition (ATR). This process involves automatic detection, classification, and tracking of a target located, or camouflaged, in an image scene. The typical procedure utilized for recognition involves a three-stage process - segmentation, feature extraction, and classification. The segmentation process is useful for dividing the image space into separate regions of interest. The feature extraction process allows the ATR system to identify and classify targets based on relevant features. Classification involves detecting and identifying the target in question.

An ATR system must be invariant toward vantage point differences. These differences include illumination changes, shadowing, noise, and occlusion. In Aerial ATR applications, the input image is typically an on-line aerial image acquired by video camera. Such real world imagery is affected by climate, season, weather, and time of day. An aerial image is also subject to geometric distortions, such as position, orientation, and scale variations.

There are many problems facing ATR systems. Normally, the target recognition process is highly data dependent. Most systems are only able to recognize a pre-specified number of targets and are unable to expand their object database. In addition, many ATR systems are encoded with predetermined tolerances in that they tend to be very sensitive to scale and orientation changes.

II. Background

MODALS [9] is a 3-D multiple object detection and location system which utilizes a neural network to simultaneously segment, detect, locate, and identify multiple targets. The neural network in MODALS is used to find discriminating features between targets and background images. The neural network is trained on 11x11 pixel-sized features. From this learning phase, a set of features is extracted and an associative mapping phase associates features with images. MODALS is able to provide robust detection, high classification, identification and location accuracy as well as a low false alarm rate. The main limitation with this approach is that the system is not rotation and scale invariant. The system does not yield the same response to an identical target if there are changes in its size or orientation. In addition, the assumed feature size of 11x11 pixels does not ensure that all relevant and necessary features can be extracted.

SAHTIRN [1] performs automatic target recognition through a three-stage process. It initially binarizes target boundary information and separates it from background clutter by using an edge detector algorithm. The object image is then fed into a multi-layer feedforward neural network that associates the input image with a cluster of feature characteristics. The final stage then utilizes a neural network to classify an object based on these associated features. SAHTIRN is able to successfully classify objects with varying scale and orientation parameters. One of the problems associated with SAHTIRN is that it is assumed a simple edge detector algorithm is adequate for segmenting objects from background. Based on this assumption, not only is the system not robust when faced with changes in lighting conditions, but objects which are not targets, such as rocks and trees, would also be processed as possible targets. In addition, the Neocognitron architecture used to associate input images with features requires additional layers if more information is required for classification. The more distinguishing features

0-7803-4859-1/98 $10.00©1998 IEEE

required, the more computationally complex the system will become.

Turk and Pentland's work [10] utilizes eigenvectors to capture the greatest variation of a set of faces to form "eigenfaces" – the significant features of a face. Face recognition is accomplished by using the eigenvector set to project an input image onto a face space and finding the face class nearest to its projection through a simple correlation technique. The system works well in identifying faces given changes in lighting and orientation. The main limitation with this approach is the system assumes the face image is segmented from the background and therefore does not address face recognition in cluttered background images. The system performance also degrades when faced with alterations in face size.

Murase and Nayar [5] performed the task of 3-D object identification and pose estimation through the use of eigenvectors. The system is able to deal with changes in lighting and orientation by incorporating these parameters into the database itself. The approach is able to successfully recognize an object and its relevant orientation in less than 1 second. However, there are some inherent problems with this technique. The object images must initially be segmented from the background in order for the recognition process to occur. In order to accomplish this, it is assumed that the system is able to distinguish differences between background and object pixels. Another problem is that the system is unable to work for an arbitrary number of rotations and illuminations. By including these parameters into the learning database, all combinations of an object must be seen before it can be recognized.

III. Technique

In our ATR application, the input is an image acquired by video camera. We first partition this input image into smaller regions of interest (ROIs). These ROIs give us the ability to systematically search for targets in an incremental fashion. Once these ROIs are extracted, we look for relevant features in the segmented image space which will signify that an object is located in the region. We accomplish this task by projecting the extracted image onto an object eigenspace. The resultant vector is then fed into a multi-stage neural network which allows the detection of targets/non-targets. The multi-stage neural network contains two neural networks which train in serial, but run in parallel. The use of a dual neural network allows for a very high target recognition rate while still maintaining an almost zero false positive alarm rate.

A. Segmentation

The segmentation process is useful for dividing the image space into separate regions of interest. These regions of interest represent a portion of the image space and are used to search for targets in a specified location. A region of interest is created though utilizing a moving window. This allows us to systematically search an image in an incremental fashion and allow for multi target detection.

Let I represent the input image, I^s represent the segmented image, N^2 represent the size of the input image and M^2 represent the area of the segmented image. The chosen value for M^2 allows for the recognition of an object scaled from ε to $M^2+\varepsilon$ pixels, where ε is chosen such that the number of relevant features needed to recognize the object are prevalent in I^s. The procedure thus utilized is invariant to variations in object size. Based on this segmentation method, the system is able to simultaneously detect N^2/M^2 target objects.

B. Feature Selection

In many image-processing applications, a classical technique used to determine the similarity between a pair of images is to estimate a correlation measure between the images. A pattern that is close to the reference image pattern produces a higher signal in the correlation plane than do other patterns. Based on this fact, we want to utilize a correlation scheme which tightly clusters together related targets while excluding non-target objects. We can accomplish this task by implementing a procedure which cleverly extracts the features needed for characterizing an object as a target. An object can thus be identified based on these extracted features. To ensure that we only select the most relevant features, we implement a process called principal component analysis (PCA).

i. Principal Component Analysis

Principal component analysis (PCA) is an optimal linear method of reducing data dimensionality by identifying the axis about which the prototype image data varies the most. PCA computes a set of orthonormal eigenvectors of an image set that captures the greatest correlation between images. Using this set of eigenvectors, the prototype images are projected onto an eigenspace. Subsequently, when a new input image is presented to the system, its projection onto the eigenspace is used to classify the input as a member of the prototype images. In this way, we can construct an

eigenvector set which represents the class of target objects. A segmented image, once projected can thus be classified as belonging (or not) to the object set.

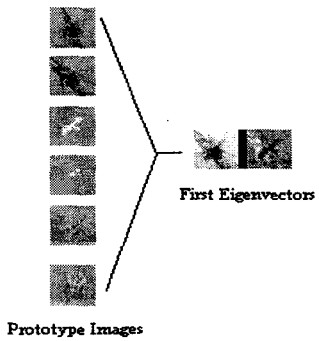


Figure 1:Orthonormal Eigenvectors of a Sample Image Set

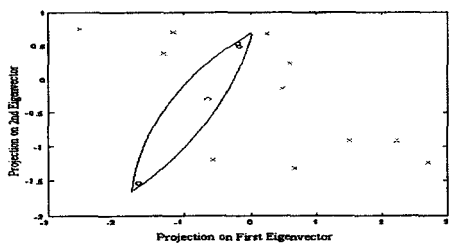
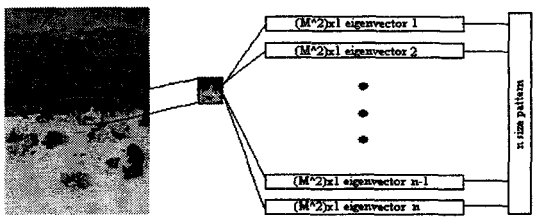


Figure 2: Eigenspace Projection
o–target image projection; x–non-target image projection

To compute the eigenvector set, we must first compute an image matrix M_i which represents all of the prototype images. The prototype images are target images containing little noise, relatively homogenous background, and no occlusions. We construct the image matrix by first transforming each prototype image into a one-dimensional vector. Each vector will correspond to a row in the image matrix. We then use the image matrix to calculate the covariance matrix of all prototype images. The mean vector of all the prototype images will be denoted m_x. The covariance matrix of all the prototype images, is defined as:

$$Cov = \left(M_i - m_x\right)\left(M_i - m_x\right)^T$$

From this, we derive a set of orthonormal eigenvectors which captures the greatest correlation between prototype images. If there are x prototype images, x eigenvectors will be created. The highest variations will stem from projecting the image onto the first vector of the eignevector set. Statistically, as we progress through the eigenvector set, each eigenvector will have decreasing variance. The majority of information is therefore contained in the first eigenvectors, and only a small amount of information is contained in the smaller eigenvector projections. We therefore will not project an image using the entire eigenvector set, and thus we reduce the number of features used to identify a target object.



Segmentation Feature Extraction

C. Multi-Stage Neural Network

Our desire is to associate as small a number of features as possible with a desired target object. By utilizing principal component analysis, we effectively reduce the number of input features required for detection. We must now construct a way to cluster together associated points located in the constructed eigenspace. Given the points derived from projecting the prototype images onto the eigenvector set, a priori knowledge can be utilized for associating target objects to eigenspace points. We also need an effective procedure for discriminating targets from background objects such as trees and rocks. This can effectively be accomplished by projecting the associated background images into the eigenspace and ensuring that their associated points are not included in the clustering algorithm.

To accomplish this clustering and discrimination process a multi-stage neural network system is utilized. The multi-stage neural network contains two neural networks which are trained in a serial fashion. Both networks possess 45 input nodes, one hidden layer with 30 hidden nodes, and one output node. The number of input parameters fed into the network correlates with the number of eigenvectors required to classify an object. The first neural network is initially trained on a pattern set created from projecting a set of prototype images onto the eigenvector set. The input images consist of a combination of regions of interest containing targets against background, targets without background, and background with no targets. The neural network output value ranges from -1.0 to 1.0. Based on this output, a discrimination threshold for detection specifies whether a region of interest contains a target (1) or non-target (-1). We utilize a high threshold value for object detection to ensure that we maintain a low false positive alarm rate.

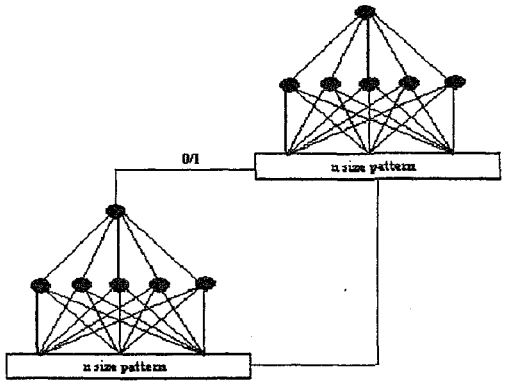


Figure 3: Multi-Stage Neural Network

Once the first network is trained, the objects which were incorrectly classified are parsed and unified into another training set. The second network is trained with the set of misclassified targets and the set of background images. We reuse the set of background images in order to maintain a low false positive alarm rate. This allows the second network to increase the system's detection rate while still ensuring non-targets are not mistakenly classified.

Learning in the multi-stage neural network is accomplished using standard backpropagation. This backpropagation algorithm allows the propagation of the network's output error to traverse back through the network. We ensure that learning is implemented in a time efficient manner by parsing the input images into three sets – a training set, a testing set, and a verification set. If the number of input images is X_n, the training set contains $.25X_n$ images, the testing set consists of $.40X_n$ images, and the verification set contains $.35X_n$ images. Instead of training the neural network until the least squares error converges to zero, we stop training when the error derived from the testing set starts to increase significantly. This ensures that we do not over train the network.

D. Implementation

After learning, the two neural networks are operated in parallel. A video image is presented to the system. After segmentation and feature extraction the projected image is presented to both networks. Because of the low false positive alarm rate, only one network will output a positive value if a target is identified. The centroid pixel representing the region of interest is correlated with a positive detection value. After the entire image is processed, the system response consists of the pixel locations of any detected objects.

IV. Results

The images used for the detection algorithm were obtained using scale target models with various backgrounds and different settings of the following independent variables: luminance, translation, scale, azimuth, and elevation. The segmented images were 30x30 pixel dimensions. The eigenvector set was computed from three separate targets possessing different orientations, scales, and luminance. All targets were centered in their respective regions of interest. The neural networks were then trained on images projected with the first 45 eigenvectors.

After learning, the system retrieves input images from the camera at a rate of 30 frames/sec. These images are of size 256x256 pixels. ROIs are constructed by incrementally traversing through the image pixel by pixel. This gives a total accumulation of 226^2 regions of interest. Since the neural network was trained on centered targets, the detection process only outputs a positive value for a ROI possessing such a target.

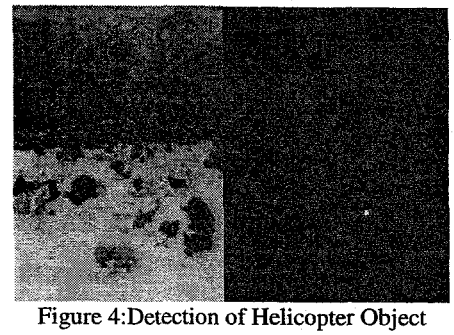


Figure 4:Detection of Helicopter Object

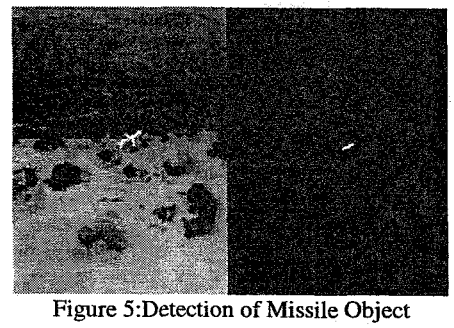


Figure 5:Detection of Missile Object

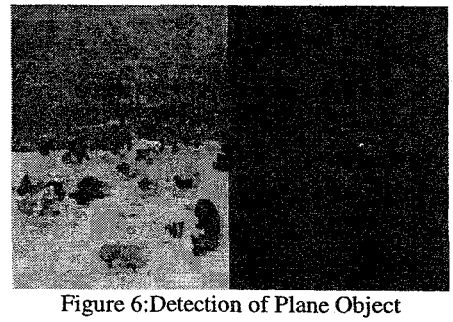


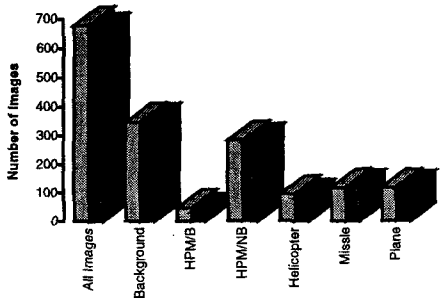
Figure 6:Detection of Plane Object

Table 1:Centroid Location Error Table

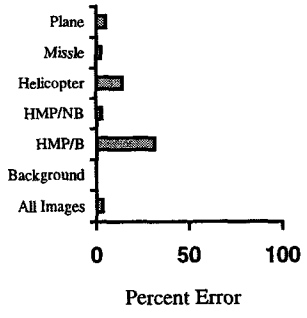
	Actual	Estimated	Error
Missle	(62, 90)	(62, 89)	1 pixels
Helicopter	(97, 154)	(97, 149)	5 pixels
Plane	(76, 95)	(76, 95)	0 pixels

The Bayes error rate is the fundamental statistical upper bound on the performance of any pattern classifier. The leave-one-out estimate is obtained by training the network with a sample from the input data. The network is trained using all the data except the one held-out sample, the verification set, which is used to test the network. The percentage of samples misclassified by this procedure is takes as a leave-one-out estimate of the Bayes error rate.

1st Neural Network Classification Rate

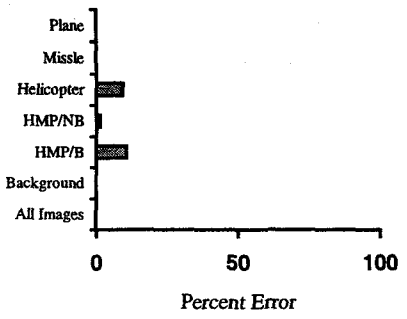


1st Neural Network Bayes Error Rate

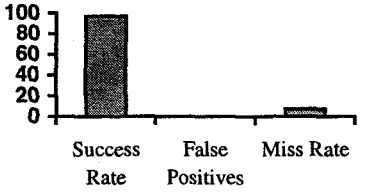


As observed, the false positive rate is effectively kept at 0%. Given the misclassified targets, we then train another network with the set of misses and the set of background images. This effectively increases our overall recognition rate, while still keeping a low false positive alarm rate.

2nd Neural Network Bayes Error Rate



Overall Detection Results



Although the overall detection result is not at 100%, we make note of the fact that the images are input at a rate of 30 frames/sec. This ensures that by keeping the number of misses at a minimum, the importance of the miss rate diminishes since the system is provided with more than one opportunity for detecting the target.

V. Conclusions

We have presented a novel approach for solving the target detection process by utilizing a multi-stage neural network. We require that a small number of input features are feed into the neural network since the more input parameters there are, the more timely and computationally expensive is the training and detection process. By utilizing principal component analysis, we ensure that the detection process adheres to this consideration. In the future, we desire to take real world aerial imagery and train the network on this data.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the Ballistic Missile Defense Organization through an agreement with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

References

[1] C.E. Daniell, D.J. Kemsley W.P. Lincoln, W.A. Tackett, G.A. Baraghimian, "Artificial neural networks for automatic target recognition", Optical Engineering, Vol. 31, No. 12, pp. 2521-2530, Dec. 1992.

[2] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks, John Wiley and Sons, New York, 1996.

[3] J.F. Gilmore, "Knowledge-based target recognition system evolution", Optical Engineering, Vol. 30, No. 5, pp. 557-570, May 1991.

[4] S. Greenberg, J. Guterman, "Neural-network classifiers for automatic real world aerial image recognition", Applied Optics, Vol. 35, No. 23, pp. 4598-4608, Aug. 1996.

[5] H. Murase, S. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance", Int. Journal of Computer Vision, Vol. 14, No. 1, pp. 5-24, Jan. 1995.

[6] S.K. Rogers, J.M. Colombi, C.E. Martin, J.C. Gainey, K.H. Fielding, T.J. Burns, D.W. Ruck, M. Kabrisky, M. Oxley, "Neural Networks for Automatic Target Recognition", Neural Networks, Vol. 8, No. 7/8, pp. 1153-1184, 1995.

[7] C. Padgett, G. Woodward, "A hierarchical, automated target recognition algorithm for a parallel analog processor", to appear in Proc. of 1997 IEEE Int. Symposium on Computational Intelligence in Robotics and Automation '97, Monterey Bay, CA, July 10-12.

[8] C. Padgett, M. Zhu, S. Suddarth, "Detection and orientation classifier for the VIGILANTE image processing system", Proc. of SPIE Int. Symposium on Aerospace/Defense Sensing and Controls, Vol. 3077, April 1997.

[9] W.A. Thoet, T.G. Rainey, D.W. Brettle, L.A. Slutz, F.S. Weingard, "ANVIL neural network program for three-dimensional automatic target recognition", Optical Engineering, Vol. 31, No. 12, pp. 2532-2539, Dec. 1992.

[10] M. Turk, A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86, 1991.

[11] S. Udomkesmalee, A. Thakoor, C. Padgett, T. Daud, W-C Fang, S. Suddarth, "VIGILANTE: An Advanced Sensing/Processing Testbed for ATR Applications", Proc. of SPIE Int. Symposium on Aerospace/Defense Sensing and Controls, Vol. 3069, April 1997.