Detection and orientation classifier for the VIGILANTE image processing system

Curtis Padgett
Jet Propulsion Laboratory
California Institute of Technology

Michael Zhu
Jet Propulsion Laboratory
California Institute of Technology

Steven Suddarth
Ballistic Missile Defense Organization
Department of Defense

VIGILANTE is an automated recognition and tracking system that closely integrates a sensing platform with a very large processing capability (over 2 TeraOPS). The architecture currently consists of an optical bench with multiple sensors, a large parallel analog pre-processor, and a digital 512 processor, parallel machine. Preliminary results on target detection and orientation are presented for an algorithm that is suitable for the VIGILANTE architecture. The technique makes use of eigenvectors calculated from image blocks (size 32x32) drawn from video sequences containing rocket targets. The eigenvectors are used to reduce the dimensionality of frame-lets (size 32x32) from the larger sensor images. These frame-lets are projected on to the eigenvectors and the resultant values are then used as an input pattern to a feed forward neural network classifier, A description and evaluation of this algorithm (including precision limitations) with respect to VIGILANTE is provided. Experiments using this technique have generated near 99target and non-target images and close to 97% identification of the rocket type.

Keywords: automatic target recognition, neural processor, eigenvectors, machine vision, system architecture

1 Introduction

The VIGILANTE project seeks to develop an integrated sensor, image processing system with the ability to detect, recognize and track a target object in real time (30 frames per second). To accomplish this, an analog neural network processor (3DANN), designed to perform 64 concurrent vector inner product operations (1x4096 dimensions) every 250 nanoseconds is used to process a 64x64 sub-window (frame-let) of a larger sensor image. A column or row loading, digital to analog input device (CLIC) can grab 1 x64 pixels in an image frame and provide the neural processor with an analog 64x64 window of the image at operating speeds. A total of 64 separate convolutions of a 256x256 image ant] the 3DANN's 64x64 masks can be performed in about 1 6 milliseconds. A large, digital SIMD computer provides post-processing support and image classification for the analog processor.

The algorithms implemented to perform the detection and recognition functions need to make use of the large processing bandwidth provided by the analog processor to effectively exploit its advantages over other processing systems. Fortunately, a large number of image processing and understanding algorithms employ the inner product as a key component in image evaluation. Image pre-processing algorithms such as sharpening, blurring, and edge detection; template matching or spatial correlation aucl convolution algorithms; and dimensionality reduction for classification; employ the inner product in deriving their respective results. In this paper we describe a detection and recognition neural network algorithm that is easily implemented on the VIGILANTE architecture. We provide preliminary detection results using this classifier on a sequence of rocket imagery provided by Ballistic Missile Defense Organization (BMDO).

2 VIGILANTE Description

The sensing component of VIGILANTE, calls for two visual sensors (one with cent rollable zoom lens), and a sensor each for infra-red (IR) and ultra-violet ([J\'). The sensing platform is to be mounted on a gimbal to allow real time tracking to be performed. Initial target acquisition and pointing is via the detection system or external to VIGILANTE (e.g. radar). An image (size 256x256) from a selected sensor is read into a frame buffer at video rates for usc by the rest of the system. The control of the gimbal and the selection of the sensor is performed by the control logic that sits on a host P6 machine.

The processing path of VIGILANTE consists of digital loading device (CLIC) that transforms the digital sensor image to 64x64 analog windows that are then presented to the analog processor (3DANN) in parallel. The analog processor performs 64 concurrent inner products with its stored templates and outputs these values every '250 nano-seconds. These values are transformed back to digital and feel into a 512 processor SIMD parallel computer (1'01'). It is here where the interpretation aucl classification algorithms are performed. These results are then sent to a host processor which makes appropriate state changes for the VIGILANTE machine.

The central component of VIGILANTE is the 3DANN module. It has 64 64x64 digitally specified weight templates. The inner product of each template and an analog conversion of a 64x64 frame-let from the current sensor window is evaluated every 250 nanoseconds resulting in a 64 dimensional output vector, **v**-

$$v_i = \sum_{j=1}^{64x64} c_j * T_{i,j}$$

where c is a 64x64 input image and T is the matrix defined by the templates. The templates in the 31) ANN module are specified with 8 bit precision. A full set of templates (64) can be loaded in approximately 1 millisecond.

The 31) ANN is supplied an analog 64x64 frame-let each operation by the CLIC. The CLIC loads a single 1x64 column or row of pixels from the buffered image each time step and converts 64x64 pixels to analog values which are then dumped to the 31) ANN in parallel for evaluation. The pixels are retained in the CLIC circuitry in digital format until shifted out. All 64 convolutions of a 256x256 image with the 64x64 3DANN templates requires approximately 16 milliseconds.

In order to do template matching, the CLIC also calculates the energy of each column or row it loads. The value is sent to the and provides this value to the 1'0 1' which keeps track of the total energy in the window. This is important when trying to determine the image location with least mismatch energy, M, for image I, and template T. It is defined as follows-

$$M(p,q) = \sum_{m,n} (I(m + p, n + q) - T(m,n))^2$$

expanding the right side gives-

$$M(p,q) = \sum_{m,n} \left(I(m+p, n+q)^2 - 2I(m+p, n+q)T(m,n) + T(m,n)^2 \right)$$

As we are generally interested in the template with least mismatch energy, finding the maximum for the inner product between T and I would be sufficient provided that all templates were normalized (we could ignore the T^2 term) and the energy of each window (the I^2 term) could be determined. If the window energy term is not available concurrently with the inner product results from the 3DANN, the convolution outputs from each template would need to be stored and updated after the normalization terms were calculated. This would significantly delay processing and limit the effectiveness of template matching in VIGILANTE.

The 64 analog values generated by the 3D ANN and the energy term calculated in the CLIC are converted to 8 bit digital values and are passed into a 512 processor SIMD machine. It consists of 4, 128 processor CNAPS boards,

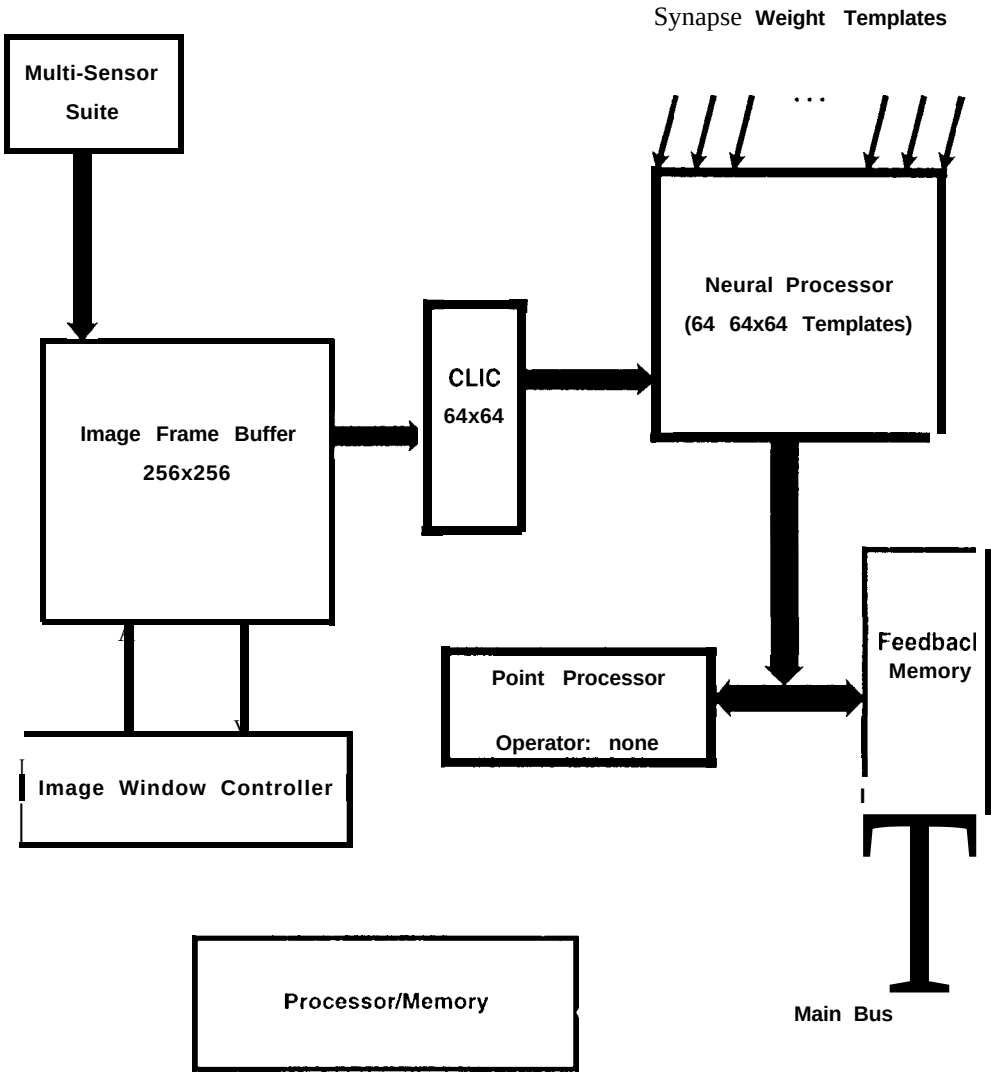

Figure 1: VIGILANTE architecture

distributed by Adaptive Solutions in Beaverton, Oregon. The set of processors aud memory provide storage for partial results, perform post-processing operations on the 3D ANN output, and evaluate the processed information for system control and image understanding. Figure 1 shows the architectural layout of the VIGILANTE image processing system

3 Rocket Imagery

To develop and evaluate the algorithms used to accomplish real time tracking of targets, the VIGILANTE team is currently collecting a suitable database of digital video images containing potential targets. Although the VIGILANTE system should be capable of tracking a wide range of target objects, this report concentrates on airborne rocket targets. The database used for these preliminary tests was made available by BMDO and consists of relatively long range, up looking, missile sequences

Typically, a frame contains a single target object in a clutter free sky. In general, no details of the missiles can be made out in any of the frames, the target object consists of a few bright pixels (1 0-20) and in some cases, an extensive plume. The images in the database suffer from a small dynamic range and considerable noise which can be attributed to digitization and the poor quality of the video tape. There are three distinct image sequences, each with a different type of missile. The sequences are from 10-30 seconds in duration with 30 individual frames for each second. Figure 2 shows examples of the three missile sequences at the beginning, middle, and end of the digitized portion of video.

All images were linearly stretched over the entire 8 bit intensity range and cropped to 256x256. Care was taken to insure that the target was still present in each of the images. The individual images were labeled with the location of the target witbin the frame and the direction of the plume with respect to the image. The plume direction values could take any of the eight major compass headings (N,NW,W,SW,S,SE,E,NE) where N indicates the top of the image. The data from the sequences were divided into training and test sets.

The goal of the detection algorithm is to determine whether or not a target exists in a given window and if it does, provide the direction of the plume. The training data is provided to set classifier parameters (by learning for example) while the testing data is used to evaluate the classifier (i.e. determine how well it generalizes to novel data). To facilitate this process the training and test set images were further modified by extracting 32x32 non-target (selected randomly) and target patches from all of the images. In addition, rotations were performed on the target data to provide a more comprehensive test of both plume direction and sensitivity of the classifier to orientation. The size of the test ancl training sets are 400 and 1200 images respectively with equal numbers of target and non-target patches. Examples of the targets from each of the missile types along with]Ion-targets can be found in Figure 3.

4 Algorithm Description

To demonstrate the flexibility of the VIGILANTE design and architecture with respect to image processing, a detection algorithm suitable for the system is described and tested on the rocket imagery. The algorithm was developed and tested initially using full floating point precision (4 bytes). In subsequent experiments, the precision of the templates and the resultant inner product were restricted to 8, 6, and 4 bits to provide a more realistic appraisal of the expected performance of such an algorithm used on VIGILANTE. The algorithm we evaluated on the rocket database consists of two stages:

1. The image window to be evaluated is projected onto each of n distinct masks of the same dimensionality.

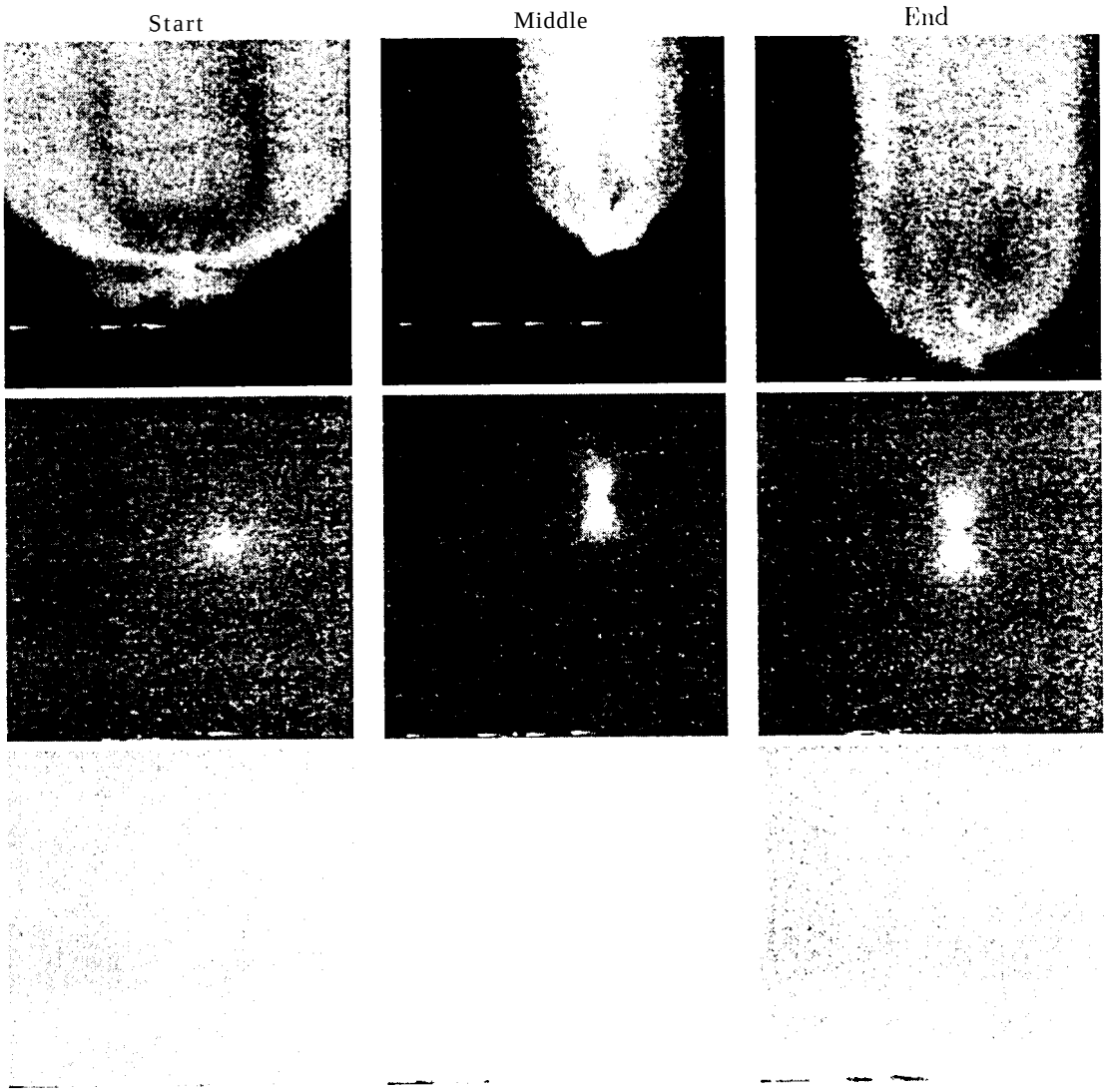

Figure 2: Examples of image sequence frames in the BM 1)0 rocket data

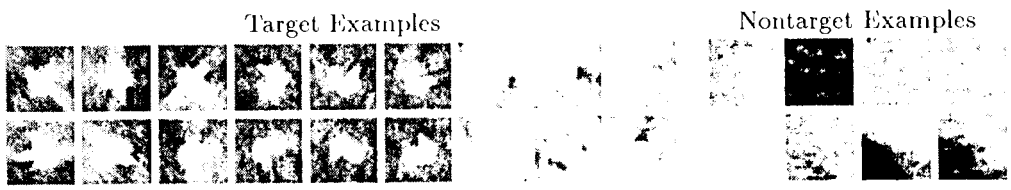

Figure 3: Examples of 32x32 windows extracted form the database for training and evaluation.

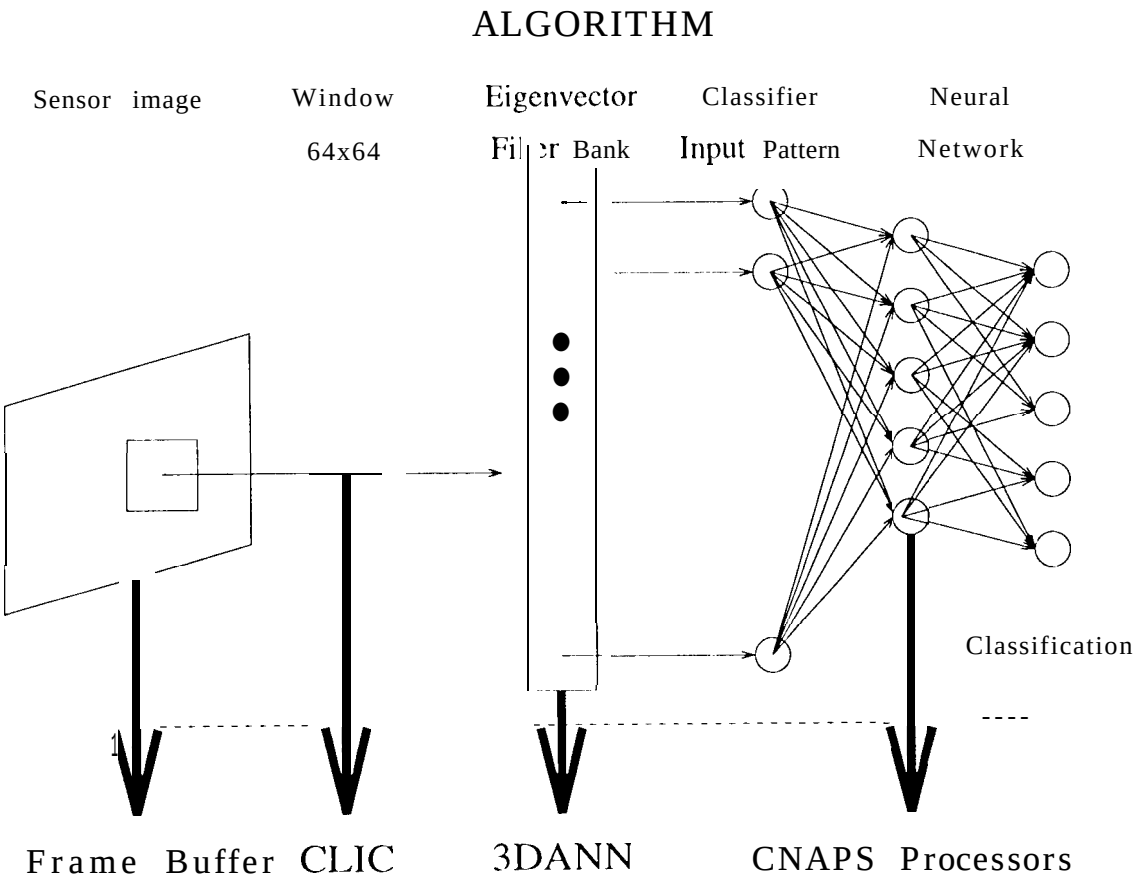

Figure 4: Algorithm to VIGILANTE architecture.

2. The n dimensional output vector serves as input to a classifier that determines if a target is present and if it is, the direction of the plume.

Figure 4 shows the mapping of the algorithm to the VIGILANTE architecture. The projections, a simple inner product, are implemented by the 3D ANN. The CLIC and Frame Buffer are used to select the window to be evaluated and finally, the 'DI' implements the classifier. In this study, a simple feed forward neural network is used to classify the windows. A single hidden layer employing a sigmoid activation function in its units (the number of units in this layer was allowed to vary) feeds 5 output variables labeled target, N, W, S, E. Similar classifiers have been developed and tested on the CNAPS processors providing satisfactory results [7,11]

Each sub-image block is evaluated independently by first projecting the 32x32 image patches on each of n masks and then providing these outputs to the neural network for classification. The mask are used to reduce the dimensionality of the data from 32x32 to n. Such a reduction should simplify the underlying statistical problem by reducing the number of free parameters and enhancing the ability of the classifier to generalize. Dimensionality reduction has been used reliably in a number of image recognition problems (where the "curse of dimensionality" is a particularly pressing problem[2]) including view invariant object recognition,[12,8,6] face recognition[9,16,3,5] and emotion classification of faces. [1,14,13]

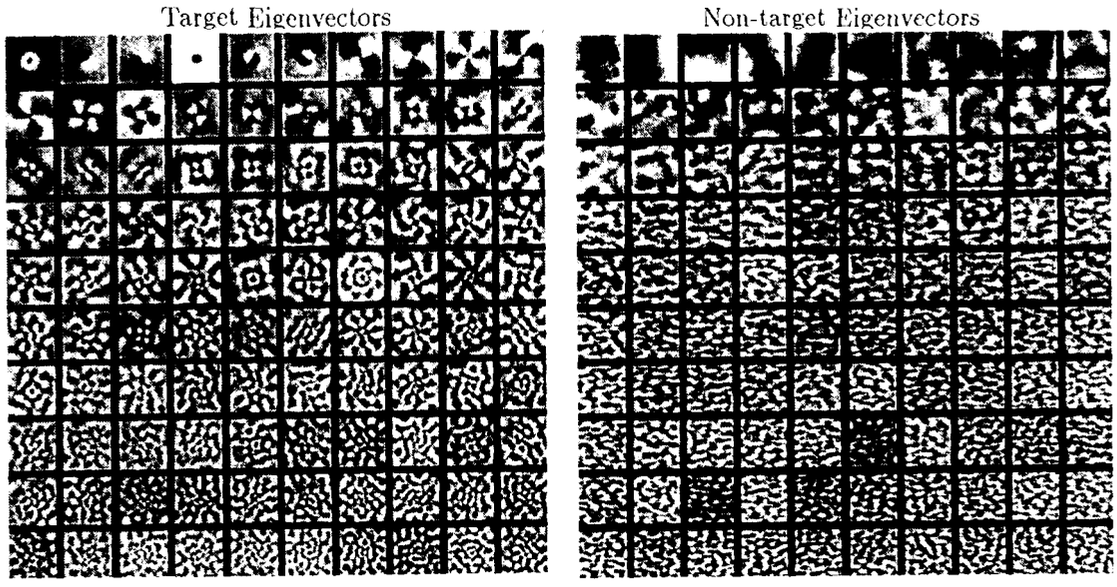

Figure 5: Eigenvectors developed from target ancl non-target 32x32 patches.

Of course the selection of the masks used to reduce the dimensionality of the data is of great importance. A poor set of masks will decrease tile effective signal to noise ratio resulting in poor discrimination between targets and nontargets. However, finding a good set of filters almost invariably depends on the set of images being evaluated. Clutter, occlusion, sensor noise, lighting dependencies, views, etc. make detection of even well modeled objects extremely difficult. One approach used to minimize some of these problems is to design the masks from actual sensor views of the object.[15] Typically interesting features are located on the object or the object itself is centered (as was the case for the rocket data) and principal components is used to extract a set of eigenvectors ordered with respect to the variance (by eigenvalues).

The eigenvectors of a set of images, I, can be obtained using the technique outlined by Turk and Pentland.[16] They found the eigenvectors, E_M of the autocorrelation matrix M ($M = I'I$), by using SVD. A set of eigenvectors E_I, of suitable dimension can then be obtained by matrix multiplication- $E_I = IE_M$. Two such sets of eigenvectors are presented in Figure 5. The set labeled "Target Eigenvectors" was generated using the 32x32 targets in the train ing set. The other set was generated frorlltllcll oll-target examples in the training set. Both sets of eigenvectors have been used successfully in different classification tasks. One advantage of the non-target eigenvectors is that they could serve as the masks for a large number of different classifiers whereas the target eigenvectors are probably only suitable for the particular targets from which they were generated. in these tests, the input patterns to the network were derived from the projections of the image on the top k eigenvectors of a *single* mask set (either target or non-target).

A simple two layer, feed-forward network is used to classify the projections of the image blocks onto the eigenvectors sets. Its hidden layer employs a non-linear sigmoid function and the network is trained with back propagation.[10] A portion of the training set (a hold out set) is reserved to stop network learning in order to enhance generalization.[4] The simple design of the feed forward network allows for efficient implementation of the algorithm on the post-processing machine. The network is required to map the mask outputs for the image blocks onto the associated 5 dimensional output vector. For non-target patterns, no error is back propagated in any of the direction fields- a simple don't care state is incorporated into the network to facilitate this aspect of the mapping.

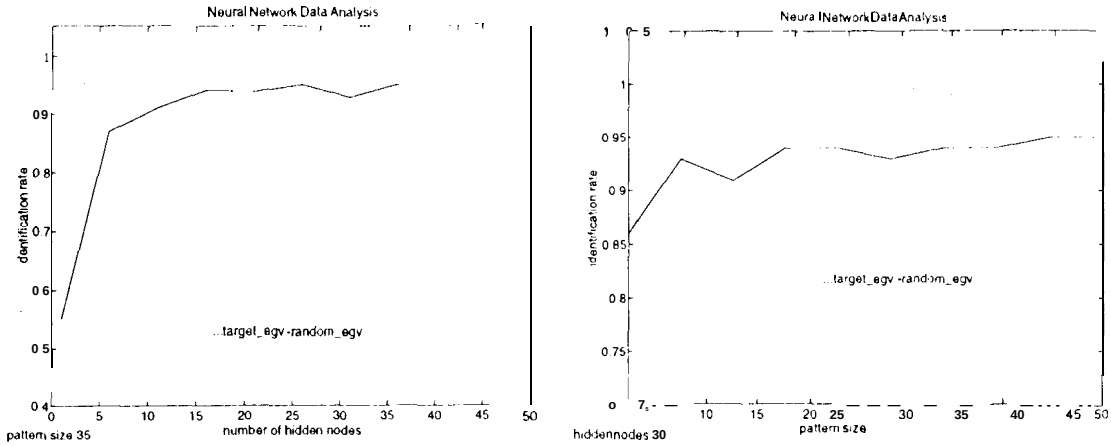

Figure 6: The graph on the left presents the fraction of test cases successfully identified vs. the number of nodes in the hidden layer. The two curves represent individual classifier results with either the eigenvectors generated by the targets or the non-targets. The dimensionality of the input pattern was 30. The graph on the right shows the same fraction vs. the size or number of projections on the two eigenvector sets. Tile size of the hidden layer was fixed at 30nodes.

5 Results

Initial tests on the rocket imagery were used to determine the architecture of the classifier and the number of projections on the eigenvector sets needed to obtain good generalization. In these experiments, full floating point precision was used in both the eigenvectors and the actual patterns presented to the neural network. We felt this was sufficient to provide a reasonable starting point for the architecture and subsequent experiments tended to bear this out. To determine if the neural network classifier could perform this detection and direction test, we initially used full floating point precision and varied the number of nodes in the hidden layer to select an architecture suitable for the problem. This was done with both sets of eigenvectors. In addition, we examined the performance of the classifier using different numbers of projections, which changes the size of the input pattern presented for classification. For the varying pattern size, the number of hidden nodes was fixed at 30. The size of the pattern was fixed at 35 when the number of hidden nodes was allowed to vary.

Figures 6 present the results of these tests. The graphs show the detection rate achieved by the classifiers which is simply the number of correct observations (target or non-target) divided by the total number of test cases. For the eigenvectors derived from tile actual target data of the training set, detection rates approaching 100% are achieved while the non-target eigenvectors produce results at around 92%. Plume direction (not shown) was correctly identified in about 90% of the target cases. This is a more difficult task for the network in that most of the imagery is without clutter and the target is quite bright with respect to the background so that detection is quite straight forward. Determining the plume direction of the 32x32 images of Figure 2 is quite difficult however, even for the human eye. Surprisingly, the identification rates are quite high for the classifiers with both eigenvector sets.

The second set of experiments looks at the how noise in the system, as reflected by reduced precision in the analog processing compo cuts, impacts the detection and direction identification rates. Figure 7 examines the impact of reduced precision in both the output of the neural processor (the inner product of the eigenvectors and the image; io in the chart) and in the actual values of the eigenvectors themselves (the template values stored in the 3D ANN). In these set of tests, precision was only examined for the target set of eigenvectors. The tests were conducted at full, 8 bit, 6 bit, and 4 bit precision.

Eigenvectors	Precision: Weights & io	Total Id Rate	Dir Id Rate
Non-target	float & float	93	92
Target	float & float	100	93
Target	float & 8bit	100	94
Target	8bit & 8bit	100	93
Target	6bit & 6bit	100	89
Target	4bit & 4bit	99	54

Figure 7: Classification rates for both non-target and target eigenvector sets using 30 projections (input dimensions) and 30 hidden nodes. The precision term indicates the number of bits used to represent both the eigenvectors and the projection values.

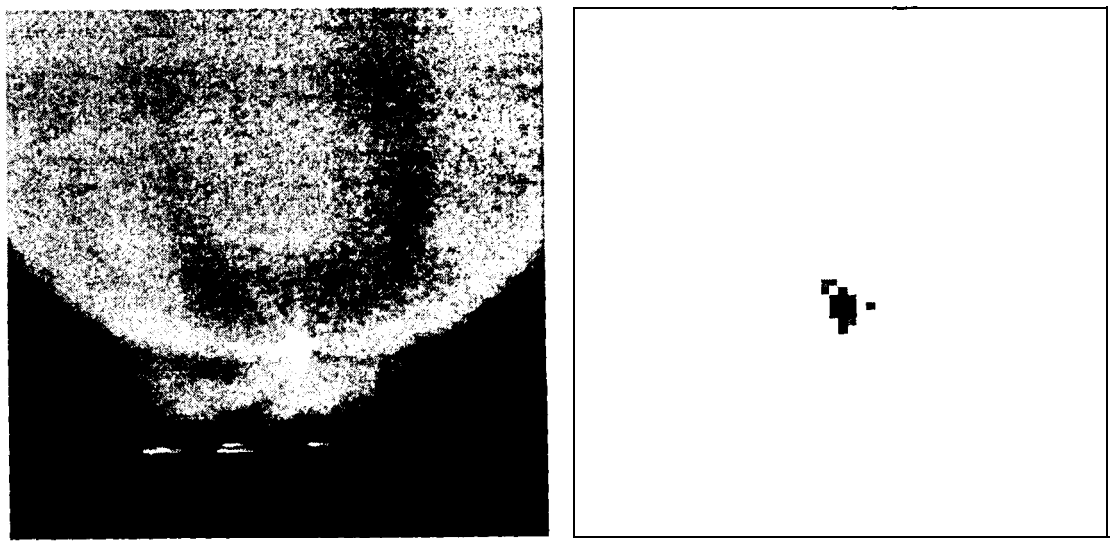

Figure 8: The left panel presents the original image. Each 32x32 block is first projected on the top 30 target eigenvectors (8 bit) and the resultant values (8 bit) are presented to the mural network for classification. The right panel shows the output of the network (in reverse video). Values below 0 are mapped to () (white in the panel).

As Figure 7 snows, reduced precision had little effect on detection in the rocket imagery. Very high detection rates occurred at all levels of precision. Figure 8 shows the response of the classifier to each of the blocks of a novel image from the sequence (right panel) at 8 bit precision in both the eigenvectors and the pattern representation. Significant degradation in direction performance appears after 6 bits. Only the 4 bit representation of the eigenvectors and the 3D ANN output significantly impacts the classifiers ability with respect to direction

Both direction and detection have high classification rates with only a few dimensions which is beneficial in two ways. First, it frees up additional template resources in the 3D ANN for alternative mask for other classification tasks. Second, it reduces the complexity of the classifier and either frees resources in the POP or decreases the time needed to process the input pattern. Provided that other target sets can be represented *by* such a compact pattern, this allows VIGILANTE to conduct multi-way searches. In addition, if the detection rate achieved by the non-target eigenvectors is acceptable, multiple classifiers can be run on the same projection values which would also increase the flexibility of the VIGILANTE system and its ability to cope with novel situations.

6 Conclusion

We have described an algorithm that is easily implemented in the VIGILANTE architecture that provides very good detection rates on rocket imagery supplied by BMDO. Provided precision can be maintained, the algorithm also generates orientation information at rates over 90%. Obviously the imagery at this early stage of the project is quite simple. To provide detection, orientation, recognition information over a wider range of targets, views, and backgrounds will decrease these classification rates. To overcome the problems introduced by the additional variables, we are currently examining a hierarchical control and classification structure using a similar methodology that proceeds through the various image understanding goals in stages.

7 REFERENCES

[1] M Bartlett, P. Viola, T. Sejnowski, J. Larsen, J. Hager, aud P. Ekman. Classifying facial action. In *Advances in Neural Information Processing Systems 8*, Cambridge, MA, 1996. MIT Press.

[2] *R.* Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton [University Press, New Jersey, 1961.

[3] 1). Beymer. Face recognition under varying pose Technical Report Al Memo No. 1461, MIT Artificial Intelligence Lab, 1993.

[4] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 19$5.

[5] R. Brunelli and T. Poggio. Face recognition: Feature versus templates. *IEEE Trans. Patt. Anal. Machine Intell.*, 15(10), October 1993.

[6] M.C. Burl, U. Fayyad, P. Perona, P. Smyth, and Ml'. Burl. Automating the hunt for volcanoes on venus. In *Proceedings of the 1994 IEEE Computer Vision and Pattern Recognition Conference*, Seattle, WA, 1 994.

[7] D. Caviglia, M. Valle, and G. Bisio. Effect of weight discretization on the back propagation learning method: Algrothim design and hardware realization. In *International Joint Conference on Neural Networks*, San Diego, CA, 1990.

[8] S. Edelman. Class similarity and viewpoint invariance in the recognition of 3d objects. Technical Report CS-TR-92-17, Weizmann Institute of Science, 1992.

[9] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Patt. Anal. Machine Intell.*, 12(1), 1990.

[10] James L. McClelland, David E. Rumelhart, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* volume 2. The MIT Press, Cambridge, 1986.

[11] D. Mueller, D. an Hammerstrom. A neural network systems componenet. In *IEEE International Conference on Neural Networks*, San Francisco, CA, 1993.

[12] H. Murase and S. Nayar. Learning and recognition of 3d objects from appearance In *IEEE* 2nd *Qualitative Vision Workshop*, New York, June 1993. IEEE Press.

[13] C. Padgett and G. Cottrell. Representing face images for classifying emotions. In *Submitted: Advances in Neural Information Processing Systems* 9, Cambridge, MA, 1997. MIT Press.

[14] C. Padgett, G. Cottrell, and R. Ac101P1w. Categorical perception in facial emotion classification. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, Hillsdale NJ, 1996. Lawerence Erlbaum.

[15] A. Pentland, B. Moghaddam, and T. Starrier. View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision & Pattern Recognition*, 1994.

[16] Matthew Turk and Alexander Pentland. Eigenfaces for recognition. *The Journal of Cognitive Neuroscience*, 3:71–86, 1991.