HIGHER-ORDER NEURAL NETWORK SOFTWARE FOR DISTORTION INVARIANT OBJECT RECOGNITION

Max B. Reid and Lilly Spirkovska

NASA Ames Research Center
Mail Stop 244-4
Moffett Field, CA 94035

ABSTRACT

The state-of-the-art in pattern recognition for such applications as automatic target recognition and industrial robotic vision relies on digital image processing. Digital image processing for automatic pattern recognition is very computationally intensive, involving feature extraction performed via large matrix operations. Digital techniques for recognizing objects regardless of their position, scale, and angular orientation are even more computationally intensive and cannot run at real time. They also are not readily adaptive due to the long time required to compute the matrix equations in digital algorithms.

We present a higher-order neural network model and software which performs the complete *feature extraction - pattern classification* paradigm required for automatic pattern recognition. Using a third-order neural network, we demonstrate complete, 100% accurate invariance to distortions of scale, position, and in-plane rotation. In a higher-order neural network, feature extraction is built into the network, and does not have to be learned. Only the relatively simple classification step must be learned. This is key to achieving very rapid training. The training set is much smaller than with standard neural network software because the higher-order network only has to be shown one view of each object to be learned, not every possible view.

The software and graphical user interface run on any Sun workstation. We also present results of the use of the neural software in a autonomous robotic vision system. Such a system could have extensive application in robotic manufacturing.

I. INTRODUCTION

Neural networks have been applied to various domains including speech recognition, trend analysis and forecasting, process monitoring, robot control, and object recognition. We present work in the position, scale, and rotation invariant (PSRI) object recognition domain. The objective in this domain is to recognize an object despite changes in the object's position in the input field, size, or in-plane orientation, as shown in Figure 1.

Pattern recognition may be viewed as a two part process of feature extraction followed by object classification[1-2]. First, a preliminary mapping from an image to a representation space is made, generally resulting in a significant degree of data reduction. A second mapping then operates on this reduced data to produce a classification or estimation in an interpretation space. Historically, these steps have required mathematical mappings operating directly on a detected image. However, digital image processing techniques are very computationally intensive, require extensive computer calculations, and have difficulty handling full in-plane distortion invariance.

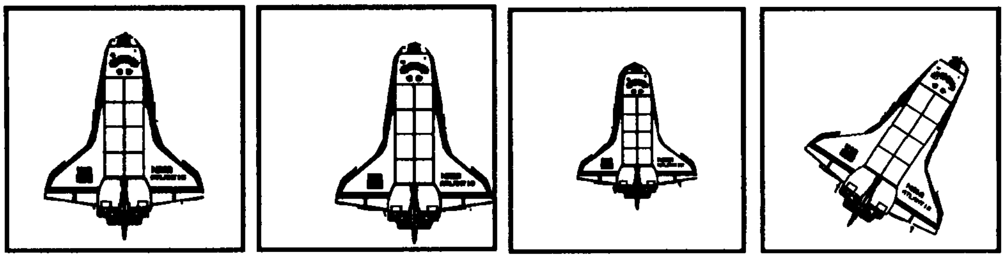


Figure 1: PSRI object recognition. In the PSRI (position, scale, and rotation invariant) object recognition domain, all four of these objects would be classified as a single object. Three distortions of the prototype in (a) are shown. The object in (b) is a translated view, (c) is scaled, and (d) is rotated in-plane.

In this paper we discuss higher order neural networks as implementations of the complete pattern recognition operation. Higher-order neural networks can be designed to implement the extraction of simple but effective features suitable for in-plane distortion invariance. Known geometric relationships are exploited and the desired invariances are built directly into the architecture of the network. Building such domain specific knowledge into the network's architecture results in a network which is pre-trained and does not need to learn invariance to distortions. For each new set of training objects, a HONN only needs to learn to distinguish between one view of each training object; it does not need to be trained on all distorted views. Therefore, training time is reduced significantly from that typically required for other neural models. Moreover, 100% recognition accuracy is guaranteed for noise-free images characterized by the built-in distortions.

We explain how known relationships can be exploited and desired invariances built into the architecture of higher-order neural networks, discuss some limitations of HONNs and how to overcome them, present simulation results demonstrating the usefulness of HONNs with practical object recognition problems, discuss the performance of HONNs with noisy test data, and present laboratory results of using a HONN to control a robot performing a manufacturing task.

II. HIGHER-ORDER NEURAL NETWORKS

The output of a node, denoted by y_i for node i, in a general higher-order neural network is given by

$$y_i = \Theta\,(\Sigma_j\, w_{ij}\, x_j + \Sigma_j\, \Sigma_k\, w_{ijk}\, x_j\, x_k\ + \Sigma_j\, \Sigma_k\, \Sigma_l\, w_{ijkl}\, x_j\, x_k\, x_l + ...) \qquad (1)$$

where $\Theta(f)$ is a nonlinear threshold function such as the hard limiting transfer function given by

$$\begin{aligned} y_i &= 1, \text{ if } f > 0, \\ y_i &= 0, \text{ otherwise,} \end{aligned} \qquad (2)$$

the x's are the excitation values of the input nodes, and the interconnection matrix elements, w, determine the weight that each input is given in the summation. Using information about relationships expected between the input nodes under various distortions, the interconnection weights can be constrained such that invariance to given distortions is built directly into the network architecture [3-7].

For instance, consider a second-order network as illustrated in Figure 2. In a second-order network, the inputs are first combined in pairs and then the output is determined from a weighted sum of these products. The output for a strictly second-order network is given by the function

$$y_i = \Theta\,(\Sigma_j\, \Sigma_k\, w_{ijk}\, x_j\, x_k). \qquad (3)$$

Pattern recognition invariant to geometrical distortions in the object are achieved by constraining the values which the weights w_{ijk} are allowed to take on.

As an example, each pair of input pixels combined in a second-order network define a line with a certain slope. As shown in Figure 3, when an object is moved or scaled, the two points in the same relative positions within the

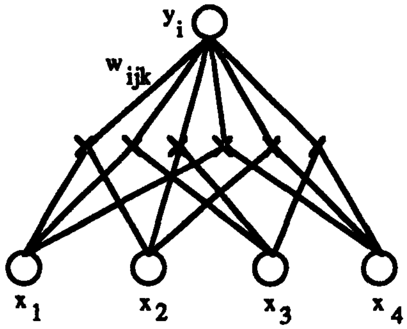


Figure 2: Second-order neural network. In a second-order neural network, the inputs are first combined in pairs (at X) and the output is determined from a weighted sum of these products.

object still form the endpoints of a line with the same slope. If all pairs of points which define the same slope are connected to the output node using the same weight, the network will be invariant to distortions in scale and translation. In particular, for two pairs of pixels (j, k) and (l, m), with coordinates (x_j, y_j), (x_k, y_k), $(x_l\ y_l)$, and (x_m, y_m) respectively, the weights are constrained according to

$$w_{ijk} = w_{ilm}, \text{ if } (y_k - y_j) / (x_k - x_j) = (y_m - y_l) / (x_m - x_l). \tag{4}$$

Alternatively, the pair of points combined in a second-order network may define a distance. As shown in Figure 4, when an object is moved or rotated within a plane, the distance between a pair of points in the same relative position on the object does not change. If all pairs of points which are separated by equal distances are connected to the output with the same weight, the network will be invariant to translation and in-plane rotation distortions. The weights for this set of invariances are constrained according to

$$w_{ijk} = w_{ilm}, \text{ if } \|d_{jk}\| = \|d_{lm}\|. \tag{5}$$

That is, the magnitude of the vector defined by pixels j and k (d_{jk}) is equal to the magnitude of the vector defined by pixels l and m (d_{lm}).

To achieve invariance to translation, scale, and in-plane rotation simultaneously, a third-order network can be used. The output for a strictly third-order network, is given by the function

$$y_i = \Theta (\Sigma_j \Sigma_k \Sigma_l w_{ijkl} x_j x_k x_l). \tag{6}$$

Each set of input pixel triplets forms a triangle with some included angles (α, β, γ), as shown in Figure 5. When an object is translated, scaled, or rotated in-plane, the three points in the same relative positions on the object still form the included angles (α, β, γ). In order to achieve invariance to all three distortions, all sets of triplets forming similar triangles are connected to the output with the same weight. That is, the weight for the triplet of inputs (j, k, l) is constrained to be a function of the associated included angles (α, β, γ) such that all elements of the alternating group on three elements (group A3) are equal

$$w_{ijkl} = w(i,\alpha,\beta,\gamma) = w(i,\beta,\gamma,\alpha) = w(i,\gamma,\alpha,\beta). \tag{7}$$

The fact that HONNs are capable of providing nonlinear separation using only a single input layer and a single output layer, with no hidden layer of nodes required, allows them to be trained using a simple rule of the form

$$\Delta\ w_{ijkl} = (t_i - y_i)\ x_j\ x_k\ x_l, \tag{8}$$

where the expected training output, t, the actual output, y, and the inputs, x, are all binary.

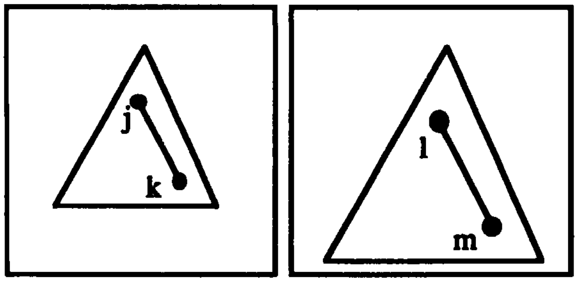


Figure 3: Translation and scale invariance in a second-order network. By constraining the network such that all pairs of points which define equal slopes use equal weights, translation and scale invariance are incorporated into a second-order neural network.

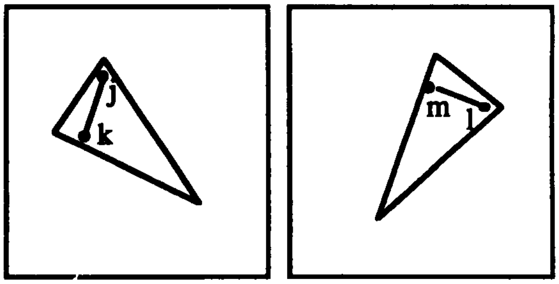


Figure 4: Translation and rotation invariance in a second-order network. By constraining the network such that all pairs of points which are equal distances away use equal weights, translation and rotation invariances are incorporated into a second-order network.

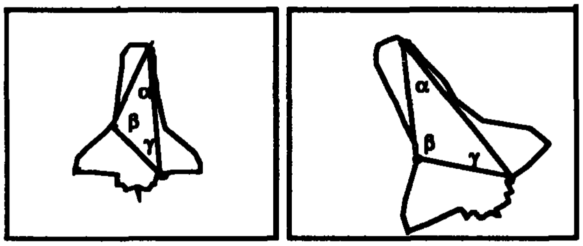


Figure 5: PSRI recognition with a third-order neural network. As long as all similar triangles are connected to the output with the same weight, a third-order network will be invariant to scale, in-plane rotation, and translation distortions.

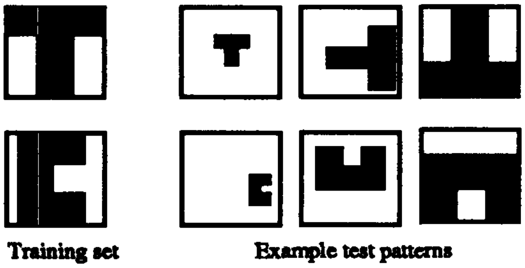


Figure 6: Training set and sample test patterns for distinguishing a "T" and a "C", invariant to translation, scale, and rotation.

The main advantage of building invariance to geometric distortions directly into the architecture of the network is that the network is forced to treat all distorted views of an object as the same object. Distortion invariance is achieved before any input vectors are presented to the network. Thus, the network needs to learn to distinguish between just one view of each object, not numerous distorted views, which leads to rapid convergence.

Software Results: Fully-connected Networks

We developed third-order network software using a Sun 3/60 workstation, where the third-order network was designed for scale, translation, and in-plane rotation invariance in a 9x9 pixel input field, giving 81 input nodes. The network had just one output node and one input layer. To build in invariance to distortions in scale, translation, and in-plane rotation, the weights were constrained according to Eq. (7) and the network was trained using the rule in Eq. (8).

The network was trained on just one view of each of the objects it was required to learn. In particular, we trained the network on the T/C recognition problem. As explained in Rumelhart [8], in the T/C problem, both objects are constructed of 5 squares, as illustrated in Figure 6, and the problem is to discriminate between them independent of translation or 90 degree rotations. In our work, the network was also required to distinguish between the objects invariant to distortions in scale.

The network learned to distinguish between all distorted views of a "T" and a "C" after just 10 passes through the training set, requiring less than 60 seconds on a Sun 3/60. The network was trained on just one view of a "T" and one view of a "C", as shown in Figure 6. Nevertheless, because the invariances are built into the architecture of the network, it was able to distinguish between the two characters regardless of their position in the input field, 90 degree rotations, or changes in size over a factor of three. In principle, recognition is invariant for any rotation angle, given sufficient resolution to draw the objects accurately.

III. EXPANDING TO PRACTICAL IMAGE SIZES

The advantages of HONNs stem from the fact that known relationships are incorporated directly into the architecture of the network. The network weights are constrained by this domain specific knowledge. Thus, fewer training passes and a smaller training set are necessary to learn to distinguish between the training objects.

The assumption behind incorporating specific knowledge into a network is that the weight values determined by the learning process result in the same output for one view of an object and a distorted view of the same object. Specifically, in our work, we assumed that the relationship expressed by Eq. (7), that all similar triangles have the same weight, constrained the network sufficiently so that an object and a distorted view of the same object would produce the same output. Using this relationship, we demonstrated that a third-order network can achieve simultaneous invariance to translation, in-plane rotation, and scale on the T/C recognition problem in a 9x9 pixel input field. Unfortunately, due to the finite resolution of actual images [7], Eq. (7) constrains the network adequately only in this limited domain but not when using a more general set of objects or a larger input field. Invariance to object scale changes can be lost when using larger image field sizes.

Problems arising from finite image resolution can be largely overcome by using edge-only images, as shown in Figure 7, and by restricting the resolution to which the angles α, β, and γ are calculated. We have shown that for a 36x36 pixel input field, angles need to be rounded to the nearest 20° in order for test objects to be recognized when scaled down to 50% of the training image size. As the input field is increased to 80x80 pixels, the angle resolution can be increased to the nearest 10°. Further increasing the input field resolution to 127x127 pixels allows the angle resolution to increase to 5°. Thus, with larger input fields, both the image resolution and the resolution to which α, β, and γ are calculated can be increased.

A greater constraint on increasing the size of images which can be evaluated using HONNs is the amount of storage required to implement the network. A network with M inputs and one output using only rth order terms requires M-choose-r interconnections. For large M, this number, which is on the order of M^r, is clearly excessive, as some storage must be used to associate each triplet of pixels with a set of included angles. In an NxN pixel input field, combinations of three pixels can be chosen in N^2-choose-3 ways. Thus, for a 9x9 pixel input field, the number of possible triplet combinations is 81-choose-3 or 85,320. Increasing the resolution to 128x128 pixels increases the number of possible interconnections to 128^2-choose-3 or 7.3×10^{11}, a number too great to store on most machines. On our Sun 3/60 with 30 MB of swap space, we can store a maximum of 5.6 million (integer) interconnections, limiting the input field size for fully connected third-order networks to 18x18 pixels. Furthermore, this number of interconnections ($\sim 10^{12}$) is far too large to allow a parallel implementation in any hardware technology that will be commonly available in the foreseeable future.

A coarse coding algorithm [7,9] can be used to permit an input field size practical for object recognition problems. The coarse coding algorithm involves overlaying fields of coarser pixels in order to represent an input field composed of smaller pixels, as shown in Figure 8. Figure 8a shows an input field of size 10x10 pixels. In Figure 9b, we show two offset but overlapping fields, each of size 5x5 "coarse" pixels. In this case, each coarse field is composed of pixels which are twice as large (in both dimensions) as in Figure 8b. To reference an input pixel using the two coarse fields requires two sets of coordinates. For instance, pixel (x=7, y=6) on the original image would be referenced as the set of coarse pixels ((x=D, y=C) & (x=III, y=III)), assuming a coordinate system of (A, B, C, D, E) for coarse field one and (I, II, III, IV, V) for coarse field two. This is a one-to-one transformation. That is, each pixel on the original image can be represented by a unique set of coarse pixels.

This transformation of an image to a set of smaller images can be used to greatly increase the resolution possible in a higher-order neural network. For example, a fully connected third-order network for a 10x10 pixel input field requires 10^2-choose-3 or 161,700 interconnections. Using 2 fields of 5x5 coarse pixels requires just 5^2-choose-3 or 2300 interconnections, accessed once for each field. The number of required interconnections is reduced by a factor of ~70. For a larger input field, the savings are even greater. For instance, for a 100x100 pixel input field, a fully connected third-order network requires 1.6×10^{11} interconnections. If we represent this field as 10 fields of 10x10 coarse pixels, only 161,700 interconnections are necessary. The number of interconnections is decreased by a factor of ~100,000.

The relationship between number of coarse fields, n, input field size, IFS, and coarse field size, CFS, in each dimension is given by [7,9]

$$IFS = (CFS * n) - (n - 1). \qquad (9)$$

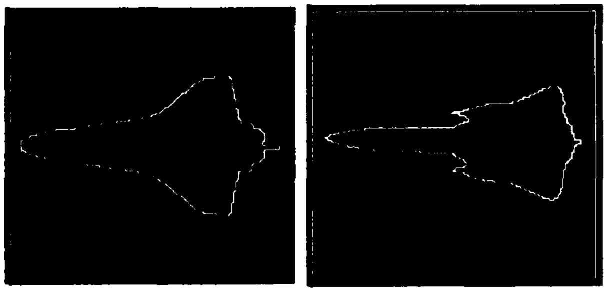


Figure 7: A binary edge-only representation of a Space Shuttle orbiter and an SR-71 aircraft, drawn in a 127x127 pixel window.

Training of the network proceeds in the usual way with one modification: the transfer function thresholds the value obtained from summing the weighted triangles over *all* coarse images associated with each training object. That is,

$$y = 1, \text{ if } \{\Sigma_n (\Sigma_j \Sigma_k \Sigma_l w_{jkl} x_j x_k x_l)\} > 0, \qquad y = 0, \text{ otherwise,} \tag{10}$$

where *j*, *k*, and *l* range from one to the coarse pixel size squared, *n* ranges from one to the number of coarse fields, the *x*'s represent coarse pixel values, and w_{jkl} represents the weight associated with the triplet of inputs (*j*, *k*, *l*). During testing, an input image is transformed into a set of coarse images. Each of these "coarser" vectors are then presented to the network and an output value determined using Eq. (10).

Software results: coarse-coded networks

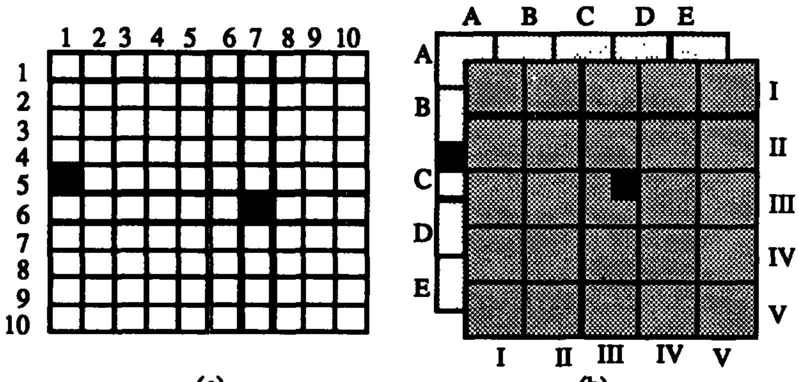
We evaluated the coarse coding technique using an expanded version of the T/C problem. Implementing coarse coding, we increased the input image resolution for the T/C problem to 127x127 pixels using 9 fields of 15x15 coarse pixels. The network was trained on just two images: the largest "T" and "C" possible within the input field, and training took just five passes.

A complete test set of translated, scaled, and one degree rotated views of the two objects in a 127x127 pixel input field consists of ~135 million images. Assuming a test rate of 200 images per hour, it would take about 940 computer-months to test all possible views. Accordingly, we limited the testing to a representative subset consisting of four sets:

(1) All translated views, but with the same orientation and scale as the training images.
(2) All views rotated in-plane at 1° intervals, centered at the same position as the training images but only 60% of the size of the training images.
(3) All scaled views of the objects, in the same orientation and centered at the same position as the training images.
(4) A representative subset of approximately 100 simultaneously translated, rotated, and scaled views of the two objects.

The network achieved 100% accuracy on all test images in sets (1) and (2). Furthermore, the network recognized, with 100% accuracy, all scaled views, from test set (3), down to 38% of the original size. Objects smaller than 38% were all classified as C's. Finally, for test set (4), the network correctly recognized all images larger than 38% of the original size, regardless of the orientation or position of the test image.

A third-order network also learned to distinguish between practical images such as a Space Shuttle Orbiter versus an SR-71 aircraft (Figure 7) in up to a 127x127 pixel input field. In this case, training took just six passes through the training set, which consisted of just one (binary, edge-only) view of each aircraft. As for the T/C problem, the network achieved 100% recognition accuracy of translated and in-plane rotated views of the two images. Additionally, the network recognized images scaled to almost half the size of the training images, regardless of their position or orientation.



(a) (b)

Figure 8: Example of a coarse-coded input field. (a) A 10x10 pixel input field. (b) Two fields of 5x5 coarse pixels.

The maximum input field resolution possible with coarse coded HONNs has not yet been reached. We ran simulations on the T/C problem coded with a variable number of 3x3 coarse pixels. A third-order network was able to learn to distinguish between the two characters in less than ten passes in an input field size of up to 4095x4095 pixels using 2047 fields. We expect a resolution of 4096x4096 is sufficient for most object recognition tasks. Notwithstanding, we also expect greater resolution is possible.

IV. TOLERANCE TO NOISE

All the demonstrations discussed so far showed the performance of HONNs in a noise-free environment. In this section, we discuss the recognition accuracy of HONNs with non-ideal test images. We consider white Gaussian noise and occlusion.

We evaluated the performance of HONNs with noisy images on two object recognition problems: an SR-71/U-2 discrimination problem and an SR-71/Space Shuttle discrimination problem. All simulations used a coarse-coded third-order network designed for a 127x127 pixel input field. We used 9 fields of 15x15 coarse pixels and a resolution of 10° for the angles α, β, and γ in Eq. (2), which allowed scale invariance over the range between 70% and 100% of the original image size. Each instantiation of the network was trained on just one binary, edge-only view of each object, as shown in Figure 10a, and training required less than ten passes through the training set.

The training sets were generated from 8-bit gray level images of actual models of the aircraft. The images were thresholded to produce binary images, and then edge detected using a digital Laplacian convolution filter with a positive derivative to produce the silhouettes shown in Figure 9a. For rotated and scaled views of the objects, the original gray level images were first scaled, then rotated, and then thresholded and edge-detected. Test images were positioned arbitrarily to validate the translation invariance of the network. Notice that the profiles of the SR-71 and Space Shuttle are somewhat similar whereas those of the SR-71 and U-2 are very different.

White Gaussian Noise

To test the tolerance of higher-order neural networks to white noise, each instantiation of the network (one for the SR-71/U-2 problem and one for the Shuttle/SR-71 problem) was tested on 1200 images generated by modifying the 8-bit gray level values of the original images using a Gaussian distribution of random numbers with a mean of 0 and a standard deviation of between 1 and 50. The noisy images were then geometrically distorted, binarized, and edge-enhanced. Typical test images which were correctly identified are shown in Figure 9b.

The results are summarized in Figure 10. The network performed with 100% accuracy for our test set for a standard deviation of up to 23 on the SR-71/U-2 problem and 26 on the Shuttle/SR-71 problem. For the similar images of the Shuttle and SR-71, the recognition accuracy quickly decreased to 75% at a σ of 30 and to 50% (which corresponds to no better than random guessing) for σ greater than 33. The SR-71/U-2 remained above 75% accuracy up to a σ of 35 (or ~14% of the gray level range) and gradually decreased to 50% at a σ of 40 (or ~16% of the gray level range). If we define "good performance" as greater than 75% accuracy, HONNs have good performance for σ up to 35 (or ~14% of the gray level range) for images with very distinct profiles and σ up to 30 (or ~12% of the gray level range) for images with similar profiles.

Occlusion

To test the tolerance of HONNs to occlusion, the two instantiations (one for the Shuttle/SR-71 problem and one for the SR-71/U-2 problem) of the third-order network built to be invariant to scale, in-plane rotation, and translation as described above were tested on occluded versions of the image pairs. We started with binary, edge-only images and added automatically-generated occlusions based on four variable parameters: the size of the occlusion, the number of occlusions, the type of occlusion, and the position of the occlusion. Objects used for occlusion were squares with a linear dimension between one pixel and twenty-nine pixels. The number of occlusion objects per image varied from one to four, and the randomly chosen type of occlusion determined whether the occlusion objects were added to or subtracted from the original image. Finally, the occlusions were randomly (uniform distribution) placed on the profile of the training images. The test set consisted of 10 samples for each combination of scale, rotation angle, occlusion size, and number of occlusions for a total of 13,920 test images per training image or 27,840 test images per recognition problem. Typical test images are shown in Figure 9c.

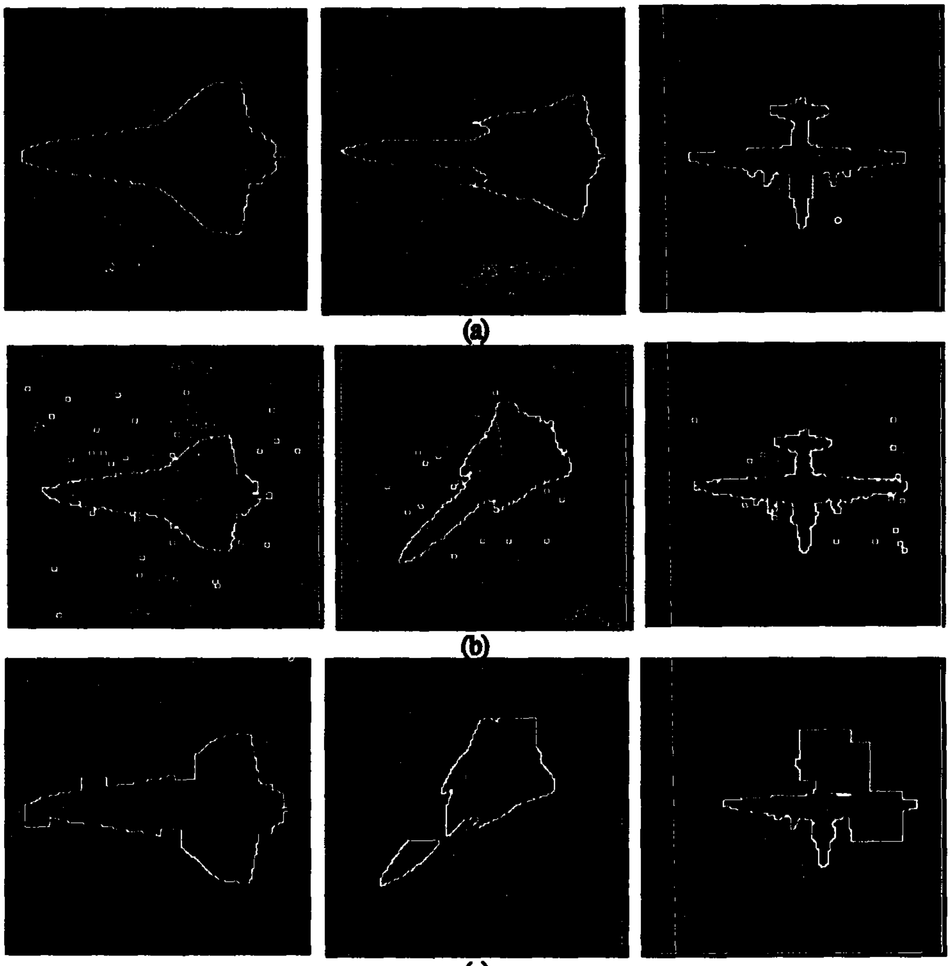


Figure 9: Training images in 127x127 pixel fields. (a) Binary edge-only training images of Space Shuttle Orbiter, SR-71, and U2. (b) Geometrically distorted and noisy test images correctly identified.
(c) Geometrically distorted and occluded test images correctly identified.

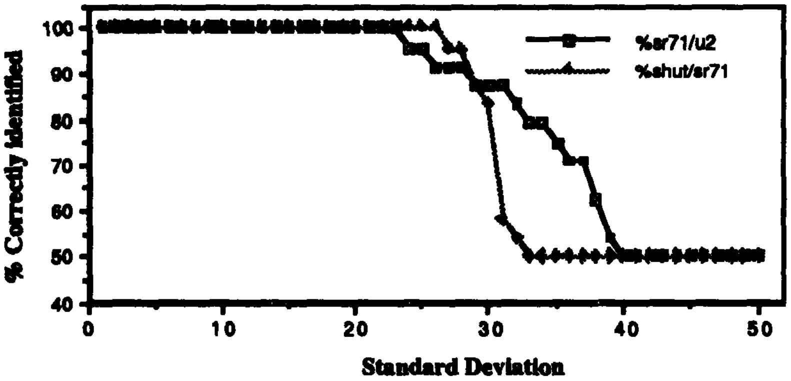


Figure 10: Tolerance of HONNs to white Gaussian noise. Each instantiation of a third-order network (one for the SR-71/U-2 problem and one for the Shuttle/SR-71 problem) was tested on 1200 test images generated by modifying the 8-bit gray level values of scaled, rotated and translated versions of the original training images using a Gaussian distribution of random numbers with a mean of 0 and a standard deviation between 1 and 50.

The performance of HONNs with occluded test images depends mostly on the number and size of occluding objects and to a lesser degree on the similarity of the training images. In the case of the Shuttle/SR-71 recognition problem, the network performed with 100% accuracy for our test set for one 16 pixel occlusion and up to four 10 pixel occlusions. It performed with better than 75% accuracy ("good performance") for up to four 19 pixel occlusions, three 21 pixel occlusions, two 24 pixel occlusions, and one 29 pixel occlusion.

For the SR-71/U-2 problem, the network exhibited good performance for the entire test set but achieved 100% accuracy only for one 4 pixel occlusion and up to four 3 pixel occlusions.

V. APPLICATION TO ROBOTIC VISION FOR MANUFACTURING

Vision processing is one of the most computationally intensive tasks required of an autonomous or semi-autonomous robot. A vision system based on a parallel implementation of a higher-order neural network can be used to perform one of the most difficult functions required of a general robotic vision system, distortion invariant object recognition, and can perform fast enough to keep pace with incoming sensor data. At Ames Research Center we have developed a robotic vision processing system to test concepts and algorithms for autonomous construction, inspection, and maintenance of space based habitats.

The benchmark task of the system is to allow a robot arm to identify and grasp an arbitrary tool moving in space with all six degrees of freedom without using any kind of cooperative marking techniques for the vision system. This is representative of one task required from the Flight Telerobotic Servicer (FTS) or the EVA Retriever, both of which are robots designed to operate in a weightless environment. A higher-order neural network can satisfy the first system task of object identification, after which other image processing sub-systems perform the tasks necessary to allow grappling.

We have tested a HONN-based vision system in the control of a Microbot robotic arm. The task was a subset of the benchmark task of allowing a robot arm to identify, track, and grasp an arbitrary tool without using any kind of cooperative marking techniques for the vision system. The robot arm carries a camera to observe the workspace below it, as shown in Figure 11.

The vision system task was to find one of a set of tools, as shown in Figure 12. The object set consists of five common tools and a structural component designed for automated in-space assembly. The work area is draped in black cloth to control the amount of background clutter. The robot was directed to look at each "bin" space in the work area, and to identify the tool located there. The tool could be located at any location within the bin, could be rotated in-plane. The camera height was not held constant, so the tools had varying apparent size. When the desired tool was found, a grappling operation was initiated.

This system also demonstrates the capability of HONN-based vision for a part/product identification task on a manufacturing assembly line. For example, parts on an assembly line passing below a camera could be quickly identified, regardless of their position, orientation, and (if need be) size.

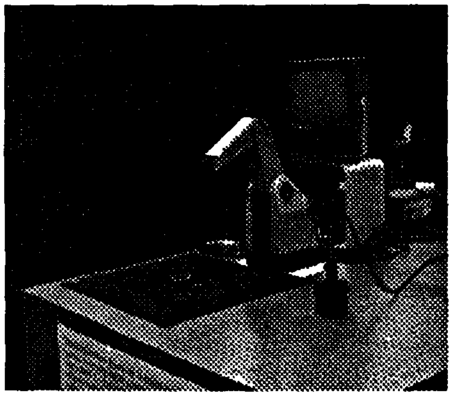


Figure 11: Photograph of the table top 5 degree-of-freedom Microbot arm and the work surface. The camera is attached to the writst of the arm

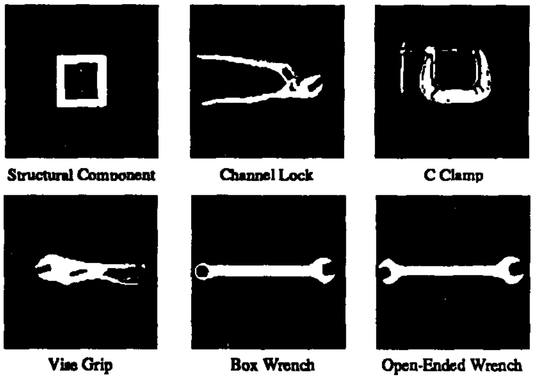


Figure 12: Binarized images of tools for recognition by the HONN vsion system. the images are edge-enhanced before being input to the HONN.

VI. CONCLUSIONS

We have shown that third-order neural networks can be trained to distinguish between two objects regardless of their position, angular orientation, or scale and achieve 100% accuracy on test images characterized by built-in distortions. Only one view of each object is required for learning and the network successfully learned to distinguish between all distorted views of the two objects in tens of passes, requiring only minutes on a Sun 3/60 workstation. In contrast, other neural network approaches require thousands of passes through a training set consisting of a much larger number of training images.

The major limitations of HONNs is that the size of the input field is limited because of the memory required for the large number of interconnections in a fully connected network. To circumvent this limitation, we developed a coarse coding algorithm which allows a third-order network to be used with a practical input field size of at least 4096x4096 pixels while retaining its ability to recognize images which have been scaled, translated, or rotated in-plane.

We explored the tolerance of higher-order neural networks (HONNs) to white Gaussian noise and to occlusion. We demonstrated that for images with an ideal separation of background/foreground gray levels, it takes a great amount of white noise in the gray level images to affect the binary, edge-only images used for training and testing the system to a sufficient degree that the performance of HONNs was seriously degraded. HONNs are also robust with respect to occlusion.

A third order neural network has been demonstrated in the laboratory for the control of a robot performing a typical manufacturing task of part identification. Our current research aims to extend the capabilities of this vision system by training a third order to recognize out-of-plane rotated versions of a training object. With scale, position, and in-plane rotation invariance built into the architecture, and out-of-plane invariance learned, a full six degree of freedom vision system can be achieved. In addition, we are working on a implementation of a third order network on a parallel processor, which will allow the identification of objects in a 128x128 pixel image at full video (60 Hz) rates.

All our current software runs on any Sun workstation, either a Sun 3/60 or a SPARC system. This software will soon by available through COSMIC, the U.S. Government's software distribution facility.

VII. REFERENCES

[1] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[2] Chen, C.H., Statistical Pattern Recognition. Washington: Hayden, 1973.

[3] G.L. Giles and T. Maxwell, "Learning, invariances, and generalization in high-order neural networks," Applied Optics, 26, 1987, pp. 4972-4978.

[4] G.L. Giles, R.D. Griffin, and T. Maxwell, "Encoding geometric invariances in higher-order neural networks," Neural Information Processing Systems, American Institute of Physics Conference Proceedings, 1988, pp. 301-309.

[5] M.B. Reid, L. Spirkovska, and E. Ochoa, "Simultaneous position, scale, and rotation invariant pattern classification using third-order neural networks," International Journal of Neural Networks, 1, 1989, pp. 154-159.

[6] M.B. Reid, L. Spirkovska, and E. Ochoa, "Rapid Training of Higher-Order Neural Networks for Invariant Pattern Recognition," Proceedings of Joint International Conference on Neural Networks, Washington, D.C., June 18-22, Vol. 1, 1989, pp. 689-692.

[7] L. Spirkovska and M.B. Reid, "Application of Higher-Order Neural Networks in the Position, Scale, and Rotation Invariant Object Recognition Domain," invited submission to Sixth Generation Computing, Wiley, New York, to be published, 1992.

[8] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Parallel Distributed Processing, Vol. 1, Ch. 8, pp. 348-352. Cambridge, Massachusetts: MIT Press, 1986.

[9] L. Spirkovska and M.B. Reid, "Coarse-Coded Higher-Order Neural Networks for PSRI Object Recognition," submitted to IEEE Transactions on Neural Networks.