Sign language recognition using competitive learning in the HAVNET neural network

Vivek A. Sujan* and Marco A. Meggiolaro

Dept. of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

An optical modeless Sign Language Recognition (SLR) system is presented. The system uses the HAusdorf-Voronoi NETwork (HAVNET), an artificial neural network designed for two-dimensional binary pattern recognition. It uses an adaptation of the Hausdorff distance to determine the similarity between an input pattern and a learned representation. A detailed review of the architecture, the learning equations, and the recognition equations for the HAVNET network are presented. Competitive learning has been implemented in training the network using a nearest-neighbor technique. The SLR system is applied to the optical recognition of 24 static symbols from the American Sign Language (ASL) convention. The SLR system represents the target images in a 80x80 pixel format. The implemented HAVNET network classifies the inputs into categories representing each of the symbols, using an output layer of 24 nodes. The network is trained with 5 different formats for each symbol and is tested with all 24 symbols in 15 new formats. Results from the SLR system without competitive training show shape identification problems, when distinguishing symbols with similar shapes. Implementation of competitive learning in the HAVNET neural network improved recognition accuracy on this task to 89%. The hand gestures are identified through a window search algorithm. Feature recognition is obtained from edge enhancement by applying a Laplacian filter and thresholding, which provides robustness to pose, color and background variations.

Keywords: neural networks, HAVNET, Hausdorff distance, American sign language

1. INTRODUCTION

Sign language recognition (SLR) systems can contribute significantly to the communication of hearing impaired people, either by supplementing the shortage of sign language interpreters or by providing an* aid for their training. Such systems need not be limited to communication for the hearing impaired, but can be extrapolated to form a basis for human-machine interfacing. Because neural networks specialize in handling ambiguous data, they are well suited for SLR tasks. Previous work has achieved high recognition rates in SLR, however such systems require model-based tracking[7] or special wearable hardware such as instrumented gloves[6]. In this work, an optical modeless SLR system is presented. The system uses the HAusdorf-Voronoi NETwork (HAVNET), an artificial neural network originally designed for two-dimensional binary pattern recognition. Several artificial neural networks such as Self-organizing maps, Adaptive Resonance theory and Neocognitron neural networks have been presented in the literature[2,3,4,5]. A brief outline of these three methods is presented. It can be seen that these methods require the 2D images be converted into multidimensional vectors before training and recognition can be done. Further, due to the choice of the comparison metric it can produce counterintuitive results.

The HAVNET architecture has been introduced in previous papers[1,8], featuring enough flexibility to incorporate self-organization and unsupervised learning, desirable properties in a pattern recognition context. It uses an adaptation of the Hausdorff distance to determine the level of similarity between an input pattern and a learned representation. This similarity metric, called truncated inverted pointwise directed Hausdorff distance, has been shown to mimic closely the psychological behavior of the human brain. In order to improve recognition performance, competitive learning is implemented in the trained network using a nearest-neighbor technique. This supervised learning process differentiates similar symbols by familiarizing the network with a training set.

The SLR system presented in this paper is applied to the optical recognition of a finger alphabet of 24 static symbols. This excludes the representation of J and Z as these are non-static symbols. The digitized images used in the demonstration are extracted from a person communicating in American Sign Language. Feature recognition is obtained from edge enhancement by applying a Laplacian filter and thresholding, which provides robustness to pose, color and background

* Correspondence: Email: vasujan@mit.edu; Telephone: 617 225 9882

variations. Other than the search process, no other pre-processing of the input images is done. The SLR system represents the final binary target images in an 80x80 pixel format, and then employs the HAVNET neural network for symbol training/recognition. The implemented HAVNET network classifies the inputs into categories representing each of the symbols, using an output layer of 24 nodes. The network is trained with 5 different formats for each symbol and is tested with all 24 symbols in 15 new formats. Results from the SLR system without competitive training show shape identification problems, especially when distinguishing symbols with similar shapes. Implementation of competitive learning in the HAVNET neural network improved recognition accuracy on this task to 89%.

1.1. Self-Organizing Maps (SOM)

In the past a great deal of organization has been experimentally observed in the human brain. Physical and mental functions can be mapped into specific areas of the brain, and these mappings are essentially constant across human beings. Furthermore, similar functions are mapped to neighboring physical areas of the brain. With this approach a family of artificial neural networks (ANNs) have been developed and called SOM. These ANNs are capable of observing input patterns and organizing them into neighborhood groups internal to the network based on the similarity found in the inputs. Typically these self-organizing maps are composed of a two layer neural network with horizontal connections in the output layer indicating interactions between neighboring neurons (see figure 1). The input layer simply distributes the inputs, via weighted connections, to the output layer neurons. The output neurons compute the classical sum-of-products, followed by a "winner-take-all" strategy where the output neuron with the largest value is set to one and the others are reduced to zero. The SOM model is by far the simplest approach to unsupervised learning and self organization in neural networks.

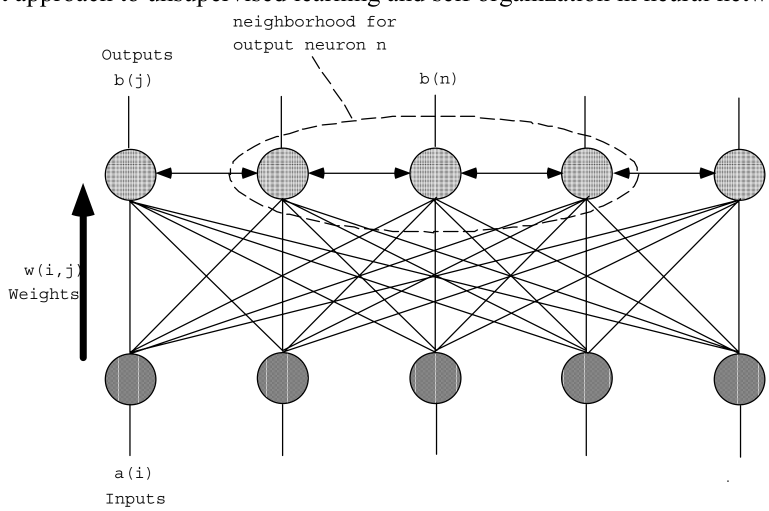


Figure 1 : Self - organizing map neural network architecture

The SOM model of unsupervised learning and self organization are ideas that are biologically feasible and have many practical applications, especially in machine vision. Additionally the SOM learns in ways that allow people to visualize what has been learned by observing the neighborhoods that have been formed.

1.2. Adaptive Resonance Theory (ART)

Humans are capable to remember something learned a long time ago, while still being able to learn something new. Typical ANNs tend to forget i.e. overwrite previously learned information, by training on the new data. This is known as the stability-plasticity dilemma and resulted in the development of the ART, where one combines the competitive learning strategy of the SOM model with a feedback and control mechanism that regulates memory coding. Weights are not changed unless the input pattern is sufficiently close to a pattern already stored by a particular output neuron. This is called resonance. Learning only occurs when resonance occurs and uncommitted neurons are recruited when a novel input pattern is encountered. ART retains the advantages of SOMs, while adding the ability to encode novel inputs without disturbing previously trained responses, but adds on the issue of mathematical complexity in implementation.

1.3. Neocognitron Neural Network

The next step in developing a neural network analogue of the human vision system we find what is known as the Neocognitron Neural Network. Unlike ART and its derivatives, the Neocognitron is a cross between a neural network and a model of the human vision system. The Neocognitron was designed specifically as a pattern recognition device, so the inputs are two dimensional images representing the retinal array. The Neocognitron is a multilayered architecture with hierarchical organization, with weights lower in the hierarchy performing more complex and abstract functions. A typical layout is shown in figure 2. Each layer in the Neocognitron architecture is made up of a number of cells which are divided into two groups of simple and complex cells. Simple cells receive inputs from the previous layer and is responsive to a certain feature in a specific position in the visual field. The simple cells are further divided into planes with all of the cells in a given plane being responsive to the same feature but in different positions on the retina. All of the simple cells in a given plane usually have the same input weights.

The complex cells in a given layer receive inputs from the simple cells of that layer. Complex cells effectively group the primitive features recognized by the simple cells, and therefore respond to more elaborate forms. Complex cells receive inputs from groups of simple cells in the current layer, and are therefore less position sensitive than the simple cells. Like the simple cells, complex cells are grouped into planes with the cells in a given plane responding to similar objects. Complex cell input weights are usually fixed. As the layers become further removed from the retina, the complex cell response becomes increasingly abstract, till the point when a complex cell in the highest layer responds to a certain complex feature or pattern on the retina.

Among the advantages of the Neocognitron are its ability to recognize patterns, based on extracted features, invariant of scale and translation. Additionally, the ability to use either supervised or unsupervised training based on data availability is also useful. The primary disadvantage lies in its mathematical complexity and immense size for implementation involving several thousand neurons, each with hundreds of weights for even a modestly sized retina.

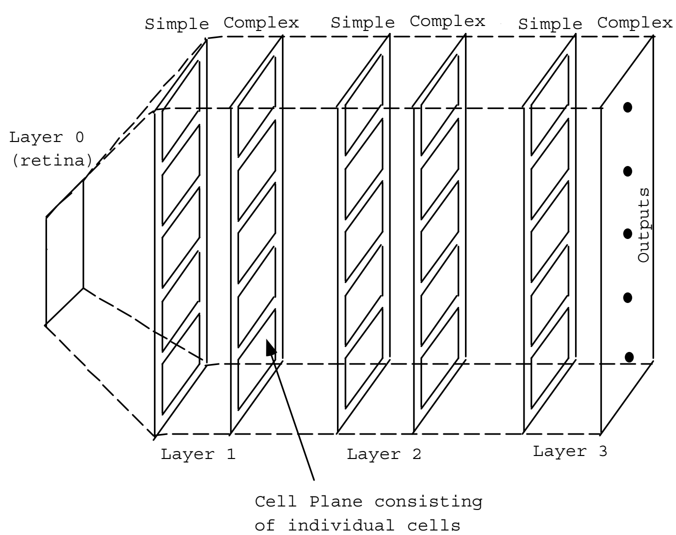


Figure 2 : Neocognitron neural network architecture

2. HAVNET NEURAL NETWORK

2.1. Similarity metrics

The choice of the Hausdorff distance as the metric of similarity between the input patterns and the learned patterns is what makes the HAVNET different from most other neural network paradigms. Most current neural networks require 2D input patterns to be converted into multidimensional vectors before training and recognition can be carried out. Learned patterns in these networks are represented as multidimensional vectors of trained weights and the measure of similarity between the presented input and the learned pattern is based on some similarity metric. Three common similarity metrics used in neural networks for binary vectors, with input vector **A** and a weight vector **W** are :

$$\text{Hamming distance : } 1 - \frac{d}{n} \tag{1}$$

$$\text{Vector dot product : } \frac{A.W}{|A|} \tag{2}$$

$$\text{Euclidean distance : } 1-\sqrt{\frac{d}{n}} \tag{3}$$

where d is the number of mismatched elements between **A** and **W**, n is the number of total elements in **A**, and |A| is the number of ones in vector **A**. Unfortunately, transforming a 2D input pattern into a multidimensional vector and then comparing that vector to learned vectors can produce behavior that is counterintuitive. 2D input patterns that appear very similar (to humans) to a particular learned pattern can generate very poor results when compared to that patterns using these metrics.

2.2. Hausdorff distance

The Hausdorff distance, when used as a measure of similarity between 2D binary patterns, has been shown to agree closely with human performance. The Hausdorff distance measures the extent to which each point of an input set lies near some point of a model set. Given two finite point sets A= {a1,.....,ap} and B = {b1,.....,bq}, the Hausdorff distance is defined as :

$$H(A,B) = \max\{h(A,B), h(B,A)\} \tag{4}$$

where the function h(A,B) computes the directed Hausdorff distance from A to B as follows:

$$h(A,B) = \max_{a \varepsilon A}\left\{\min_{b \varepsilon B}\{\|a-b\|\}\right\} \tag{5}$$

where ||a-b|| is typically the Euclidean distance between points a and b. The directed Hausdorff distance identifies that point in A that is furthest from any point in B and measures the distance from that point to its nearest neighbor in B. Hence if h(A,B) = d, then all points in A are within distance d of some point in B. The (undirected) Hausdorff distance is the maximum of the two directed distances between two point sets A and B, hence if the Hausdorff distance is d, then all the points of set A are within distance d of some point in set B and vice versa. The Hausdorff distance exhibits many desirable properties for pattern recognition.

(a) It is known to be a metric over the set of all closed bounded sets.
(b) It is everywhere non-negative
(c) It obeys the properties of identity, symmetry and triangle inequality

In the context of pattern recognition this means that a shape is identical only to itself, that the order of comparison of two shapes does not matter and that if two shapes are highly dissimilar they cannot be similar to a third shape, which proves to be a very important feature. Three adaptations of the Hausdorff distance is required in order to make it appropriate for use in a neural network.

(1) The directed Hausdorff distance was computed on a point wise basis for each point in A

$$h(a,B) = \min_{b \varepsilon B}\{\|a-b\|\} \tag{6}$$

(2) Each of pointwise distances was truncated to limit the maximum distance (limits noise) and inverted so that a closer match generated a larger number.

$$h_\delta(a,B) = \begin{Bmatrix} \delta - h(a,B) & if \quad h(a,B) < \delta \\ 0 & otherwise \end{Bmatrix} \tag{7}$$

where δ is the truncation distance.

(3) The truncated inverted pointwise distances are averaged for the point set A as follows :

$$\overline{h_\delta}(A,B) = \frac{\sum_{a \epsilon A} h_\delta(a,B)}{p} \qquad (8)$$

where p is the number of points in set A. This now gives us the average truncated inverted pointwise directed Hausdorff distance. Averaging has a normalizing effect on the measurement, making it invariant to the number of points present in the set and contributing to the noise immunity.

2.3. HAVNET Architecture

Figure 3 shows an overview of the HAVNET architecture. The neural network acts as a binary pattern classifier. It takes as inputs a 2D binary pattern and employs feed forward processing to produce an analog output value per node, indicating the level of match between the input pattern and the class represented by the node. The HAVNET neural network consists of the following :

- Plastic layer - contains several planes of neurons with the weights that are adjusted during training . One plastic layer plane is required for each pattern learned by the network.
- Voronoi layer - utilizes a matrix of fixed weights (i.e. one unique plane), serves to measure the distance between individual points in the input and learned patterns.
- Hausdorff layer - uses information generated by the Voronoi layer to compute the overall level of similarity between the input and learned patterns. One output node is required in the Hausdorff layer for every unique pattern learned.
- Normalizing subsystem - has the effect of making the response of the network invariant to the number of 1's in the input pattern and negates the effect of uneven training of the nodes.

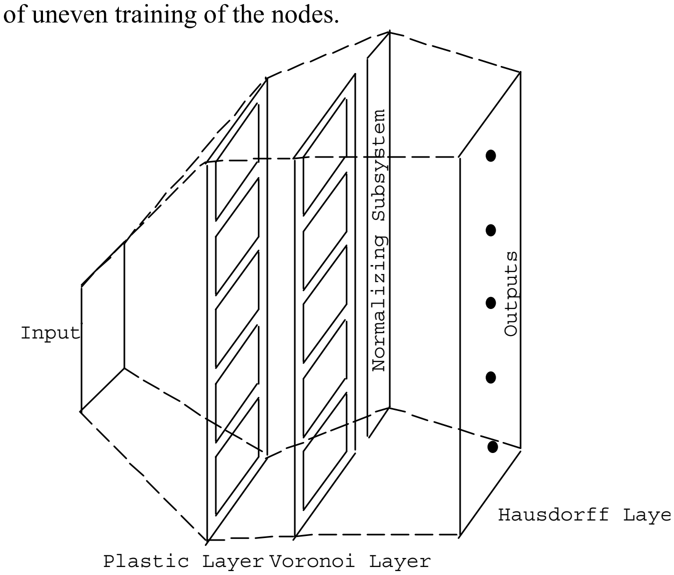


Figure 3 : HAVNET Architecture

2.3.1. HAVNET learning

Learning in this version of the HAVNET network is done off-line and in a supervised manner. A new node is added for every new pattern encountered, which involves the creation of a new plane in the plastic layer, a new plane in the Voronoi layer and a new output node in the Hausdorff layer. The network is informed of the class of the object presented (supervised learning). The following steps are involved in training the network :

- The binary input pattern of dimensions X and Y, is presented to the network during the learning phase as : $a_{x,y} = (0,1)$ where x=1...X; y=1...Y
- Prior to learning each weight matrix w^n of the N nodes of the network in the plastic layer is initialized to zero as follows

$$w^n_{(x+\delta),(y+\delta)} = 0 \qquad n = 1...N \qquad (9)$$

$$w^n_0 = 0$$

- The quantity δ is defined as the span of the Voronoi layer, and is identical to the truncation distance defined earlier in section 2.2. The truncation distance, as employed in the network, is a positive integer value that is somewhat less than the dimensions of the input pattern. The weight w^n_0 is defined as the averaging weight for a node (indicating the amount of training a node has received), and is trained during each training pass regardless of the input pattern.
- Given $0<\alpha<1$ (learning rate) and the training pattern for node N, then node n is trained as follows (where t is the number of training iterations)--see (Rosandich, 1997) for proof of convergence for stability:

$$\Delta w^n_{(x+\delta),(y+\delta)} = a^m_{x,y}\alpha(1 - w^n_{(x+\delta),(y+\delta)}) \tag{10}$$

$$\Delta w^n_0 = \alpha(1 - w^n_0) \tag{11}$$

$$and$$

$$w^{n(t+1)}_{(x+\delta),(y+\delta)} = w^{n(t)}_{(x+\delta),(y+\delta)} + \Delta w^{n(t)}_{(x+\delta),(y+\delta)} \tag{12}$$

$$w^{n(t+1)}_0 = w^{n(t)}_0 + \Delta w^{n(t)}_0 \tag{13}$$

- Once the network has been trained, the trained weights can be adjusted using nearest neighbor competition, where the excitation levels of the network output nodes that represent patterns other than the one presented for response is within a neighborhood parameter μ of the output of the correct node, then the plastic layer weights corresponding to the incorrect node are reduced as follows :

$$if \qquad net^c < net^i + \mu \qquad then$$

$$\Delta w^i_{(x+\delta),(y+\delta)} = a^m_{x,y}\alpha_c w^i_{(x+\delta),(y+\delta)} \tag{14}$$

$$w^{i(t+1)}_{(x+\delta),(y+\delta)} = w^{i(t)}_{(x+\delta),(y+\delta)} - \Delta w^{i(t)}_{(x+\delta),(y+\delta)} \tag{15}$$

- Competitive learning has the advantage that it provides for improved recognition by the network so as to be able to differentiate between similar characters such as 3 and 8. The training cycle can be repeated till satisfactory performance is obtained or no further improvement is seen.

2.3.2. HAVNET recognition

In the recognition mode the network attempts to classify an arbitrary input pattern into one of the classes represented by the trained network nodes. To clarify how the previously mentioned Hausdorff distance is employed here, the Voronoi layer is introduced. A Voronoi surface is constructed for a 2D set of points A, by first locating the members of A in the x-y plane, and then plotting in the z dimension the distance from any point in the x-y plane to the nearest point that is a member of A. When this distance is not allowed to exceed some value δ then the surface is defined as a truncated Voronoi surface. This surface can be used conveniently to compute the directed Hausdorff distance between two point sets A and B. The maximum of these values is the directed Hausdorff distance h(B,A). While computing the pointwise distance between a particular input pattern and the learned pattern, the network generates a truncated Voronoi surface for the point of interest. When the plastic layer weights are projected through this surface a response is generated. The maximum of this response represents the contribution of that particular point in the input pattern to the total calculation, which will be computed by averaging and normalizing similar responses over the entire input pattern. Mathematically this is broken down as follows :

- Response of a node n to an input pattern A^m is determined by computing the output of the plastic layer first, where $i,j = -\delta...\delta$ is the Voronoi layer span.

$$b^n_{(x+i),(y+j)} = w^n_{(x+i),(y+j)} a^m_{x,y} \tag{16}$$

- Given the plastic layer outputs, the outputs from the Voronoi layer for node n, is computed as follows:

$$c^n_{x,y} = a^n_{x,y} \max_j \{\max_i \{v_{i,j} b^n_{(x+i),(y+j)}\}\} \tag{17}$$

- The Voronoi weights $v_{i,j}$ are the same for all nodes and are computed as follows :

$$v_{i,j} = 100\left(1 - \frac{\sqrt{i^2 + j^2}}{\delta + 1}\right) \tag{18}$$

- Once the outputs from the Voronoi layer are determined, the responses of the Hausdorff layer neurons are computed as follows :

$$net^n = \frac{1}{w_0^n p_a^m p_w^n} \sum_{y=1}^{Y} \sum_{x=1}^{X} c_{x,y}^n \tag{19}$$

$$where$$

$$p_a^m = \sum_{y=1}^{Y} \sum_{x=1}^{X} \Phi(a_{x,y}) \tag{20}$$

$$p_w^n = \sum_{y=1}^{Y} \sum_{x=1}^{X} \Phi(w_{(x+\delta),(y+\delta)}^n) \tag{21}$$

$$and$$

$$\Phi(x) = \begin{Bmatrix} 1 & if & x > 0 \\ 0 & otherwise & \end{Bmatrix} \tag{22}$$

- The final outputs net^n indicate the similarity of the input pattern to the patterns that have been learned by that node. These values have also been normalized number of ones in the input vector and the past training history of the node.

3. EXPERIMENTAL IMPLEMENTATION OF HAVNET ARCHITECTURE

A HAVNET network was designed to accept binary 2D input patterns in an 80x80 pixel format. A monochrome CCD camera with a 510x492 pixel array and a 42 degree field of view was used in acquiring the digital images. The hand signs were generated at a distance of approximately 5 ft from the camera. This ensures that the hand signal would be contained in an 80x80 pixel window for an average sized individual. The images were taken against a dark back ground and dark clothes with frontal illumination for contrast enhancement. Each image is filtered with a Laplacian filter and processed with a blurring kernel. The result is slightly different than that obtained from a LOG filter, and more effective in reducing "noise" caused by texture, reflections, soft edges (such as skin fold lines on the hands), etc. Finally, the image is converted into a binary image by thresholding. This threshold value for the binary image is selected directly from the image histogram after the blurring operation.

A window search algorithm traces the region of interest. The system is assisted by an input clue of initial search coordinates. Such a search technique has the potential of simplicity and shorter processing times, as no specific differentiation is attempted in identifying the hand generating the signal and other objects in the field of view (e.g. the face, feet and other hand). All additional searches are initiated at these coordinates until the system is prompted otherwise. Additionally, such a search methodology is easily modifiable to account for more complex signals generated by multiple hand motions, while keeping the search time relatively short. Using this technique, it is seen that the search settles into a local maxima around the hand generating the signal at an average of 0.3 seconds on a Pentium 166 MHz system. This places the target within the 80x80 window and prepares it for neural network processing. Figure 4 shows a subset of input signals and the processed binary images in an 80x80 pixel grid.

The network is designed to classify the inputs into 24 categories, with one category representing each of the 24 alphabets (excluding J and Z). The output layer of the network consisted of 24 nodes, one for each category. Other than the search process mentioned above, no other pre-processing of the input images was done. The network used a Voronoi layer span of 2, so that the plastic layer weights a distance of 3 or more units from an input point would generate zero output, where a unit is defined as a pixel in the input pattern. The Voronoi layer weights were fixed for all the nodes and using the equations above we get a Voronoi distribution seen in figure 5.

These weights are used by the network to compute the truncated inverted pointwise directed Hausdorff distance from an input pixel at the center of the pattern to any position in the plastic layer. Since that span of the Voronoi layer was 2, the effect of projecting the pointwise Voronoi surface onto the plastic layer extended two pixels beyond the 80x80 giving plastic layer weight matrices of 82x82. Whenever a new image was presented to the network another weight matrix was added to the plastic layer. A normalizing weight was also created for each image class and was trained each time regardless of the configuration of the input pattern.

(a)

(b)

Figure 4: Sample alphabets A, E, I, O (a) hand signal, (b) processed network input image

6	25	33	25	6
25	53	67	53	25
33	67	100	67	33
25	53	67	53	25
6	25	33	25	6

Figure 5 : Voronoi weight distribution for $\delta = 2$

3.1 Discussion

- The network was trained with 5 different hand signals. There were small modifications in the signals such as position, orientation and distance from camera. The network was tested on all 24 nodes with 15 new signals.
- Without competitive training the networks performs at about 83% success rate with a winner-take-all evaluation of the network's response. By incorporating competitive training the network performance improves to 89%. It is conceivable that the network performance may be enhanced by larger training sets, larger images, better image filtering and enhancement. Table 1 summarizes these results. Further, increasing the Voronoi layer distance, δ, results in decreased performance. In more extreme cases competitive training is not sufficient to compensate for these effects.
- Training with competitive compensation proves to give improved results yet the choice of competitive training can be questionable. Competitive training is supervised; consequently there are many ways to competitively train such networks. Further work should be done in exploring the possibilities of different approaches in competitive training.
- The choice of the input pattern proves to be a difficult and challenging one as the figures are broad and consume a significant fraction of the 80x80 grid, thus giving a high degree of similarity regions in the nodes. Further work may be done by selecting a larger pixel grid and larger signals, thus making them more distinct.
- Typical shape identification problems occur when distinguishing between similar patterns, such as A, E, M, N, S, T. Such problems are typically resolved with the competitive training. Table 2 and 3 demonstrate such examples.
- As expected distinct figures such as U have very little problem in being identified. See table 4.

- Although the ASL set used was essentially generic, training may be expanded to incorporate a variety of ASL characters including presenting the signals in significantly differing orientations. This would provide for a larger training set and universe of identification. Such an approach may include additional network nodes for signals differing greatly in orientation/position/distance from camera. This would make the network robust to position, orientation and scale.

Table 1: HAVNET testing results

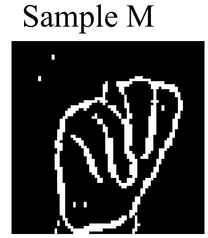
Character	**Number presented**	**Correctly identified (w/o competitive training)**	**Percentage correct (w/o competitive training)**	**Correctly identified (w/ competitive training)**	**Percentage correct (w/ competitive training)**
A	15	13	86.67	14	93.34
B	15	15	100	15	100
C	15	14	93.34	15	100
D	15	12	80.0	13	86.67
E	15	12	80.0	14	93.34
F	15	13	86.67	14	93.34
G	15	12	80.0	13	86.67
H	15	12	80.0	13	80.0
I	15	13	86.67	13	86.67
K	15	12	80.0	14	93.34
L	15	12	80.0	13	86.67
M	15	11	73.34	12	80.0
N	15	10	66.67	12	80.0
O	15	12	80.0	13	86.67
P	15	12	80.0	12	80.0
Q	15	11	73.34	13	86.67
R	15	11	73.34	12	80.0
S	15	12	80.0	13	86.67
T	15	12	80.0	13	86.67
U	15	13	86.67	14	93.34
V	15	12	80.0	13	86.67
W	15	15	100	15	100
X	15	14	93.34	14	93.34
Y	15	15	100	15	100
Total	**360**	**300**	**83.34**	**322**	**89.44**

Table 2: HAVNET recognition of "M"

Before competitive training (failure)	**After competitive training (success)**
Network value for A = 0.054357	Network value for A = 0.057023
Network value for E = 0.032557	Network value for E = 0.047345
Network value for I = 0.053060	Network value for I = 0.046329
Network value for M = 0.082472	**Network value for M = 0.093045**
Network value for N = 0.086271	**Network value for N = 0.088664**
Network value for O = 0.045787	Network value for O = 0.046083
Network value for S = 0.050392	Network value for S = 0.054366
Network value for T = 0.041136	Network value for T = 0.072770
Network value for U = 0.045423	Network value for U = 0.051997

***Note that only a 9 of 24 network responses are shown for brevity**

Sample M



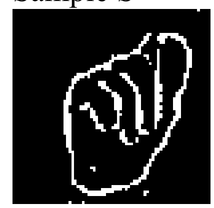
Sample N

Table 3: HAVNET recognition of "T"

Before competitive training (failure)	After competitive training (success)
Network value for A = 0.054562	Network value for A = 0.060460
Network value for E = 0.060454	Network value for E = 0.055776
Network value for I = 0.057189	Network value for I = 0.044715
Network value for M = 0.047659	Network value for M = 0.065850
Network value for N = 0.047774	Network value for N = 0.066967
Network value for O = 0.058678	Network value for O = 0.050496
Network value for S = 0.086787	**Network value for S = 0.059965**
Network value for T = 0.074582	**Network value for T = 0.072685**
Network value for U = 0.044439	Network value for U = 0.045623

***Note that only a 9 of 24 network responses are shown for brevity**

Sample S



Sample T

Table 4: HAVNET recognition of "U"

Before competitive training (success)
Network value for A = 0.045898
Network value for E = 0.046431
Network value for I = 0.047329
Network value for M = 0.055952
Network value for N = 0.062496
Network value for O = 0.040223
Network value for S = 0.048187
Network value for T = 0.048419
Network value for U = 0.091023

***Note that only a 9 of 24 network responses are shown for brevity**

Sample U

4. CONCLUSIONS

Pattern recognition is one of the most natural applications of artificial neural networks. Several neural network models were introduced, each of which has demonstrated considerable capability in the pattern recognition domain. The HAVNET neural network was the method that was finally selected and explored further because of it unique and new approach to artificial pattern recognition. The HAVNET network is the first known neural network paradigm to take advantage of the Hausdorff distance as a metric of similarity between two dimensional patterns. In doing so, the network inherits the desirable properties of the Hausdorff distance and therefore duplicates human performance more accurately than most previous neural network architectures. The network is fully developed and well behaved mathematically (i.e. convergence is guaranteed), and the network architecture is flexible enough to incorporate self-organization, unsupervised learning and nearest neighbor competitive learning in the plastic layer as required by specific applications. The network can also represent multiple aspects of a single object class, a feature that makes it much more useful in real world object recognition applications.

Certain aspects of the HAVNET neural network are built upon past work in the field. The architecture has much in common with that of Neocognitron and the learning process are similar to those employed in the ART and SOM networks. The use of aspects to represent characteristic views is also based on the approach used in the past. In a more general sense, even the cones generated by the Voronoi layer are similar to the fuzzy membership functions used in many applications and they also resemble the radial basis functions that have been also successfully employed in neural network applications.

In this work, an optical modeless SLR system is presented. The system uses the HAVNET, an artificial neural network originally designed for two-dimensional binary pattern recognition. The SLR system presented in this paper is applied to the optical recognition of a finger alphabet of 24 static symbols. The digitized images used in the demonstration are extracted from a person communicating in American Sign Language. Feature recognition is obtained from edge enhancement by applying a Laplacian filter and thresholding, which provides robustness to pose, color and background variations. Other than the search process, no other pre-processing of the input images is done. The SLR system represents the final binary target images in an 80x80 pixel format, and then employs the HAVNET neural network for symbol training/recognition. The implemented HAVNET network classifies the inputs into categories representing each of the symbols, using an output layer of 24 nodes. The network has been trained with 5 different formats for each symbol and is tested with all 24 symbols in 15 new formats. Results from the SLR system without competitive training show shape identification problems, especially when distinguishing symbols with similar shapes. Implementation of competitive learning in the HAVNET neural network improved recognition accuracy on this task from 83% to 89%.

REFERENCES

1. Rosandich, R.G. *Intelligent Visual Inspection using Artificial Neural Networks.* Chapman & Hall, 1997.
2. Taylor, J.G. *Neural Networks and their Applications.* Wiley, 1996.
3. Korn. *Neural Network Experiments on Personal Computers and Workstations.* The MIT Press, Cambridge, 1991.
4. Carpenter, G.A. and S. Grossberg. *Neural Networks for Vision and Image Processing*. The MIT Press, Cambridge, 1992.
5. Haykin. *Neural Networks.* MacMillan, 1994.
6. Erenshteyn, R., Laskov, P., Foulds, R., Messing, L. and Stern, G. "Recognition approach to gesture language understanding". Proceedings of the 13th International Conference on Pattern Recognition, IEEE Comput. Soc. Press. Part, vol.3, pp.431-435, Los Alamitos, CA, USA, 1996.
7. Huang, C.-L. and Huang, W.-Y. " Sign language recognition using model-based tracking and a 3D Hopfield neural network". Machine Vision & Applications, vol.10, no.5-6, pp.292-307, Springer-Verlag, USA, 1998.
8. Rosandich, R.G. "Visual learning and invariant recognition of complex two-dimensional objects using the HAVNET neural network". Intelligent Engineering Systems Through Artificial Neural Networks, Vol.4, pp.377-84, ASME, New York, NY, USA, 1994.