Optical disk implementation of radial basis classifiers

M. A. Neifeld, S. Rakshit, A. A. Yamamura, S. Kobayashi and D. Psaltis

California Institute of Technology, Department of Electrical Engineering
Pasadena, California 91125

ABSTRACT

We describe an optical disk based system for handwritten character recognition. The recognition scheme is based on a radial basis function approach to pattern classification. The optical system computes the Euclidean distance between an unknown input and 650 stored patterns at a demonstrated rate of 26,000 pattern comparisons per second.

1. INTRODUCTION

The optical disk is a computer controlled binary storage medium with a capacity of more than 10^{10} bits. The 2-D storage format of the optical disk makes parallel access to data an attractive possibility. The optical disk can be thought of as a computer addressed, 2-D binary spatial light modulator with a space bandwidth product of 10^{10} pixels. A number of potential applications which take advantage of these characteristics exist and have been discussed elsewhere in the literature[1,2].

In this paper we describe an optical disk based implementation of a handwritten character recognition system. A neural network architecture based on Radial Basis Function (RBF) interpolation schemes is trained to recognize the handwritten numerals 0-9. The RBF approach is one in which a hypersurface that separates the patterns may be learned from examples as the sum of radially symmetric basis functions[3]. In contrast with techniques that utilize image projections onto radially symmetric functions in the 2-D *feature space*, these basis functions are defined in the N-dimensional *input space* of the image patterns. The resulting surface can be considered to be a discriminant surface for some classification task. In order to classify an input vector using an RBF network, the distance between the unknown input and the array of basis function *centers* must be computed. This computation is well suited to an optical disk based processor when the number of centers is large. The optical implementation we describe here takes advantage of the storage capacity and the fast parallel access capabilities of the optical disk.

2. RADIAL BASIS FUNCTIONS

The RBF approach to pattern recognition lies between supervised output error driven learning algorithms like Back Error Propagation (BEP) and memory intensive sample based systems such as K-Nearest Neighbor (KNN) classifiers, in terms of learning time[4,5]. This approach results in a network whose processing units are slightly more complicated than simple linear thresholding devices; however, the overall network computation required to perform a single classification can be much less time consuming than in the KNN approach if a reduced representation can be found. The motivation for using a RBF network to perform pattern recognition tasks comes from the relatively well established mathematical framework associated with regularization theory and hypersurface reconstruction[6]. In hypersurface reconstruction the problem is to construct a *machine* or function $\hat{F}(\underline{W}, \underline{X})$, which takes a vector $\underline{X}$ into a prescribed output $F(\underline{X})$. The vector $\underline{W}$ is a parameter vector used to tune the estimate $\hat{F}$. For simplicity, we will consider only one dimensional outputs. In order to construct $\hat{F}$ a set of training samples taken from the mapping to be learned (ie. the underlying hypersurface to be approximated) is provided $\{\underline{x_i} \rightarrow F(\underline{x_i}); i = 1, ..., M\}$ and the problem becomes to choose the form of $\hat{F}$ and the appropriate parameters $\underline{W}$, such that $\hat{F}(\underline{W}, \underline{x_i}) = F(\underline{x_i})$ for $i = 1, ..., M$. This problem is very similar to the pattern recognition problem where one is given a set of training patterns and is asked to find a classifier $\hat{F}$ with the appropriate parameters $\underline{W}$, such that the resulting machine classifies the training set correctly. In both cases we desire that future samples be mapped correctly and that the system behave well in the presence of noise. In order to obtain these desirable characteristics in hypersurface reconstruction, a criterion of smoothness is often placed on the estimator $\hat{F}$. One approach for specifying a smooth $\hat{F}$ is the RBF approach.

If we choose $\hat{F}$ to be of the following form :

$$\hat{F}(\underline{W}, \underline{X}) = \sum_{i=1}^{\tilde{M} \leq M} a_i exp(-|\underline{X} - \underline{t^i}|^2/\sigma_i^2),$$

where $\underline{W} = \{\underline{t^i}, \sigma_i, a_i : i = 1, ..., M\}$, then the resulting machine is called a radial basis function interpolator/classifier. We see that the RBF classifier seeks to approximate the underlying function as a sum of gaussian "bumps". According to the above expression, the approximation $\hat{F}$ comprises $\tilde{M}$ of these bumps each centered at $\underline{t^i}$ with width σ_i and weighted by a_i in the final output. Given a set of training vectors $\{\underline{x_i} : i = 1, ..., M\}$, we may estimate the parameters $\underline{W}$ such that $\hat{F}(\underline{x_i}) \approx F(\underline{x_i})$ using any number of supervised and/or unsupervised algorithms.

The RBF approach may be considered to be a neural network architecture as shown in figure 1. We define the RBF unit in Fig. 1a as a "neuron" with response given by:

$$y_i = exp(-|\underline{X} - \underline{t^i}|^2/\sigma_i^2),$$

where $\underline{t^i}$ is called the center or template associated with neuron i and σ_i is called the width. These units are depicted in the second layer of figure 1b. The output layer of the RBF network consists of a single linear unit whose output is simply the weighted sum of its inputs. The overall network mapping then is defined by :

$$\hat{f}(\underline{W}, \underline{X}) = \sum_{i=1}^{\tilde{M} \leq M} a_i exp(-|\underline{X} - \underline{t^i}|^2/\sigma_i^2)$$

as desired. In figure 2a we show a RBF network for estimating a function of two variables and in figure 2b we depict an example of an input space configuration of the resulting mapping. The points in fig. 2b represent the training data and the broken circles represent the four gaussian basis functions used to construct the RBF network. In order to generate fig. 2b we utilized a k-means algorithm with k=4 to determine the centers of the basis functions[7]. This procedure results in determination of the four centers shown. In order to determine the widths associated with each center, a KNN algorithm was used. The five nearest neighbors to each center were chosen and the average of these five distances was used as σ_i for the associated bump. Note that these procedures result in the determination of the centers $\underline{t_i}$ and the widths σ_i in a completely unsupervised fashion. Specifically, the first layer of a RBF network may be trained without using an error driven procedure thereby reducing training time. Training of the output layer can be accomplished through the use of either a mean squared error minimization procedure or a relatively simple perceptron learning algorithm[8].

3. RBF BASED HANDWRITTEN CHARACTER RECOGNITION

In this section we will describe the implementation of the RBF classifier in the solution of a handwritten character recognition problem. We will consider the 10 class problem of identifying handwritten digits 0-9. Using a SUN3/60 workstation, several authors were asked to draw the numerals 0-9 on a 16×16 grid. The resulting database of 950 images (95 per class) was randomly separated into a 300 element testing set and a 650 element reference library. Examples of characters from the training and testing sets are shown in figure 3.

In order to provide shift and scale invariance to our system we first preprocessed both training and testing sets so that each 16×16 image was centered and scaled to a 10×10 window. Following this preprocessing, the 10×10 pixel input field is *unrastered* to form a 100 bit binary vector and each such vector $\underline{t^i}$ corresponding to each of the 650 preprocessed training or reference images, is stored on the optical disk as a radial line. For each vector $\underline{t^i}$, we also store its complement $\underline{\bar{t}^i}$ in an adjacent sub area. This method of encoding allows us to simulate bipolar templates thus eliminating the need to remove signal dependent bias. The pixel size in this experiment was chosen to be 177 tracks by 116 pixels along track. Storing one vector per sub area allows us to record up to 43 vectors per sector or 1376 templates per disk.

The actual architecture we have implemented using the optical disk is represented schematically in Figure 4. A preprocessed 100 bit binary vector $\underline{x}$, is presented to the system shown in Fig. 4 and the first layer of RBF units

compute the RBF projections $y_i = exp(-|\underline{x} - \underline{t^i}|^2/\sigma_i^2)$. We have chosen to use as RBF centers $\{\underline{t^i}\}$, all 650 reference images of the training set. This choice of centers also facilitated an earlier KNN based handwritten character recognition system which has been reported elsewhere[9]. After the RBF projections are calculated in the middle layer, this 650 dimensional intermediate representation is then transformed using the interconnection matrix $\underline{\underline{W}}$ to arrive at a 10 dimensional output representation as shown. Each output neuron corresponds to one of the classes 0-9 and a winner take all network then computes the classification. Since we have chosen to use the entire 650 template training set as RBF centers, the only learning required for the first layer of this network is for the widths $\{\sigma_i\}$. The second layer of course must be trained to perform the desired classification on the resulting RBF representations.

There are many potential training algorithms for $\{\sigma_i\}$ and $\underline{\underline{W}}$. The most successful algorithm we found for computing the widths $\{\sigma_i\}$, was to make σ_i proportional to the distance between template $\underline{t^i}$ and its nearest neighbor. That is

$$\sigma_i = \tilde{\sigma} \min_{j \neq i} |\underline{t^j} - \underline{t^i}|.$$

Training of the output layer was most successful when $\underline{\underline{W}}$ was initialized with a binary address algorithm and then trained using the perceptron learning algorithm. The binary address algorithm does not require specific knowledge of the intermediate representations generated during training, it only requires knowledge of the class assignment of each of the 650 RBF centers. This reduces second layer computation time and improves network performance. The binary address algorithm defines the initial $\underline{\underline{W}}$ as

$$w_{ij} = \begin{cases} 1 & if \quad \underline{t^j} \in \Omega_i \\ -1 & otherwise. \end{cases}$$

Following this initialization, the perceptron algorithm is used to incorporate detailed knowledge of the training representations into the output layer weights. Using these procedures for training the RBF network, we have a best RBF performance of 89% as shown in Table 1. Although the trend with increasing $\tilde{\sigma}$ is an improvement in network performance, in general, the broader the basis functions the longer the perceptron algorithm will take to converge. For this reason, table 1 does not contain any entries for $\tilde{\sigma} > 1.2$. We note here that the best RBF network performance of 89% compares with a best KNN system performance of 83% using the same template library. This performance can also be compared with a single layer of 10 neurons, each trained with the perceptron algorithm using the 650 image reference library. The recognition rate in this case is 75% on the 300 element testing set. In general, we would expect an improvement in RBF network performance with variable centers where both the number and location of those centers are optimized. This case was not studied here as we are primarily interested in the performance of the optical implementation.

In Figure 5 we show the RBF widths computed using the procedure described above. Each row in the figure represents the widths associated with centers in a single class. There are therefore 65 blocks per row and 10 rows in Figure 5. Each block in the figure is a grey scale coding of the width associated with the corresponding template with dark = 0 width. Using this encoding, each row of the figure corresponds to the values of σ_i for templates in a single class. It is interesting to notice that the second row in Figure 5 corresponding to handwritten ones, is particularly dark indicating that these vectors tend to be well clustered or in general located close to other vectors. Also in Figure 5 we can see that the width associated with one particular template representing a handwritten six, is quite broad indicating that this vector is basically isolated in the input space. Using the same display format as in Figure 5, Figure 6 shows the second layer weights generated for the 'best' RBF network. The single bright row in each weight vector indicates that the weight vector is tuned to intermediate representations from essentially one class. The 89% recognition rate achieved using the RBF network corresponds to 33 misclassifications out of 300. An example of one misclassification from each class along with the intermediate representation associated with the misclassified input is shown in Figure 7. Next to each input character in the figure is the erroneous class to which it was assigned. Most of these misclassifications, although they would most likely not be made by a human classifier, can be easily understood by inspection of Figure 7.

4. OPTICAL RBF CLASSIFIER

A schematic of the optical system used to compute the distance between some unknown preprocessed input image and each template stored on the disk in the format described above, is shown in figure 8. In this architecture, an Epson LCTV is used as a 1-D SLM to present the preprocessed input character to the system. An image of the input vector is formed as a radial line on the disk as shown, and the total diffracted intensity is collected by the output lens and measured using a Photodyne 1500XP detector. The detector output represents the inner product between the input vector and the illuminated reference vector. The postprocessing system for this experiment consists of two parts. First, a sample/hold (S/H)·circuit is used to detect the peaks of the raw detector output. These peaks represent the desired inner products. The S/H circuit is clocked by a signal which is phase locked to the sector markers in the data stream which appear as 32 bright radial lines on the disk and provide a strong diffracted signal. The second stage of postprocessing consists of an A/D converter board in an IBM PC followed by software which implements the desired recognition algorithm.

The 650 reference images were preprocessed as described above and stored on the disk along with their complements, as 100 bit binary vectors. Using a disk rotation rate of 20Hz, these 1300 vectors were processed at a rate of 26,000 inner products per second equivalent to 2,600,000 binary operations per second. The 300 testing images were preprocessed and stored in an IBM PC which drove the LCTV and provided input vectors to the system. An example of the raw detector output for the *all one* input vector is shown in figure 9. The two tallest peaks in this trace correspond to sector markers on the disk and represent the inner product between the *all one* vector and itself. From this data we can calculate the effective brightness per input pixel as measured at the detector as 0.6nW. This value is in good agreement with the known optical losses in the system. The other peaks in figure 9 provide normalization data which is stored in normalization memory and read out during postprocessing. The PC samples the inner product signal once per peak, averages 4 rotations worth of data (total acquisition time $\approx$0.2s) and computes the Euclidean distances from the inner products as :

$$|\underline{x} - \underline{t^i}|^2 = |\underline{x}|^2 + |\underline{t^i}|^2 - 2\underline{x} \cdot \underline{t^i},$$

where $\underline{x}$ is the unknown input image and $\underline{t^i}$ is a stored template. Since our optical system actually measures $\underline{t^i} \cdot \underline{x}$ and $\overline{\underline{t^i}} \cdot \underline{x}$, we may form the distance for binary vectors as :

$$\begin{aligned} |\underline{x}|^2 = (\underline{x} \cdot \underline{1}) \qquad & \underline{1} = (1, 1, ..., 1) \\ = \underline{x} \cdot (\underline{t^i} + \overline{\underline{t^i}}) & \end{aligned}$$

so that

$$|\underline{x} - \underline{t^i}|^2 = |\underline{t^i}|^2 + \underline{x} \cdot \overline{\underline{t^i}} - \underline{x} \cdot \underline{t^i}.$$

Once again, $|\underline{t^i}|^2$ for $i = 1, ..., 650$ is stored in normalization memory and read out during the postprocessing stage.

The optical disk based inner product calculations are collected by the postprocessing system which computes the required gaussian weighting and simulates the output layer where a classification is made. These postprocessing steps were carried out in software for our experiments. The classification rate for the optical RBF system was 83%. This is compared with a recognition rate of 79% using an optical KNN network based on the same template data. A comparison between the performances of the optical system and a computer simulation is shown in Table 2. The table entries indicate the number of correct classifications out of 30 for each of the 10 classes 0-9. The behavior of the optical system is encouraging considering the various noise sources inherent to this implementation. In order to better understand the effect of these imperfections on the RBF network performance, a computer model was constructed which incorporates error sources such as finite contrast, nonuniform illumination profile, detector noise, and quantization noise. Using values for the error variables as measured from the optical apparatus, we found that nonuniformity of the illumination profile was the limiting factor in our experiment. A plot of classification rate vs. log of the $1/e^2$ gaussian profile width is given in figure 10. We can see from Fig. 10 that for a measured profile parameter of 1.8, the expected recognition rate drops to 86%. This rate then is the noise limited optical system performance and agrees well with the experimentally demonstrated 83%. The cumulative effect of these errors can be measured a second way, directly from the distance calculations. In figure 11 we show the 650 distances computed for a single input image (a handwritten 3) using both the ideal computer simulation and the optical system. From

the figure we see that there is a substantial variation between these two plots. This variation can be quantified by computing the RMS distance error over the entire testing set as :

$$\Delta D_{RMS} = \frac{\sqrt{\frac{1}{M}\sum_{i=1}^{M}(d_i^{Opt} - d_i^{Sim})^2}}{\frac{1}{M}\sum_{i=1}^{M} d_i^{Sim}},$$

where d_i^{Sim} and d_i^{Opt} are the Euclidean distances between the 300 input images and the 650 templates calculated from simulation and the optical system respectively. There are M=195,000 such measurements in our case. For the results presented here, the RMS distance error was found to be ΔD_{RMS} =28.5% . Although this error is quite large, the recognition rate obtained using the optical system is in satisfactory agreement with the expected rate attesting to the robustness of the RBF approach.

5. CONCLUSION

We have demonstrated an optical system which can implement a RBF pattern classifier. Our system achieved a processing rate of 2,600,000 binary operations per second corresponding to the computation of 13,000 Euclidean distances per second. The capability of the optical disk based system is limited by the maximum length of template vectors (10^4 bits), the maximum number of template vectors (10^5) and the maximum disk rotation rate (100Hz). These upper bounds correspond to a processing rate of 10^{11} binary operations per second. The accuracy of this optical system has been measured with respect to its performance on a handwritten character recognition task. The optical system realized an 83% recognition rate which compares well with an expected rate of 89% as predicted using a computer simulation.

6. ACKNOWLEDGEMENTS

This work is supported by the U.S. Army Research Office and the Defense Advanced Research Projects Agency. Seiji Kobayashi is with Sony Corporation. Alan Yamamura is supported by a fellowship from the Fannie and John Hertz Foundation.

7. REFERENCES

[1] D. Psaltis, M. A. Neifeld, A. A. Yamamura, "Image Correlators Using Optical Memory Disks," *Optics Letters*, Vol. 14, pp. 429–431, 1989.

[2] D. Psaltis, M. A. Neifeld, A. A. Yamamura, and S. Kobayashi, "Optical memory disks in optical information processing," *Applied Optics*, Vol. 29, No. 14, May 1990.

[3] John Moody and Christian Darken, "Fast Learning in Networks of Locally Tuned Processing Units," *Neural Computation*, Vol. 1, pp. 281–294, 1989.

[4] Rumelhart and McClelland, "Parallel Distributed Processing," Vol. 1, MIT Press, 1986.

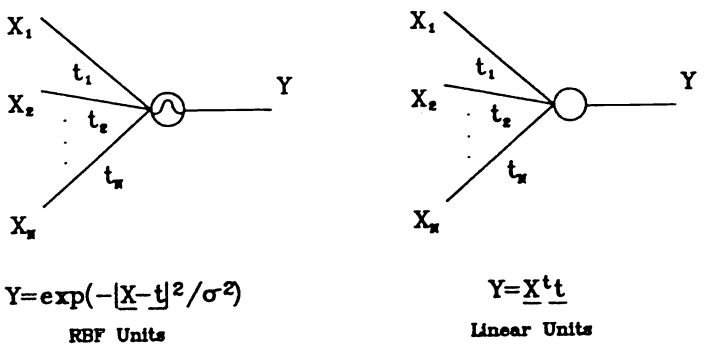
[5] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Vol. IT-13, No. 1, January 1967.

[6] T. Poggio, F. Girosi, "A Theory of Networks for Approximation and Learning," *MIT AI Laboratory and Center for Biological Information Processing Whitaker College*, AI Memo No. 1140, CPIB Paper No. 31, July 1989.

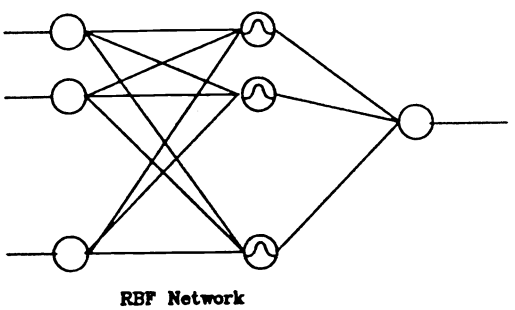
[7] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Berkeley Symposium on Math. Stat. and Prob. I.*, Berkeley and L. A. California, 1967.

[8] R. Duda and P. Hart, "Pattern Classification and Scene Analysis," pp. 141–147, John Wiley and Sons, 1973.

[9] M. A. Neifeld, S. Kobayashi, A. A. Yamamura and D. Psaltis, "Optical Disk Based Processor for Handwritten Character Recognition," *Presented at The International Neural Network Conference*, INNC-90-Paris, July, 1990.

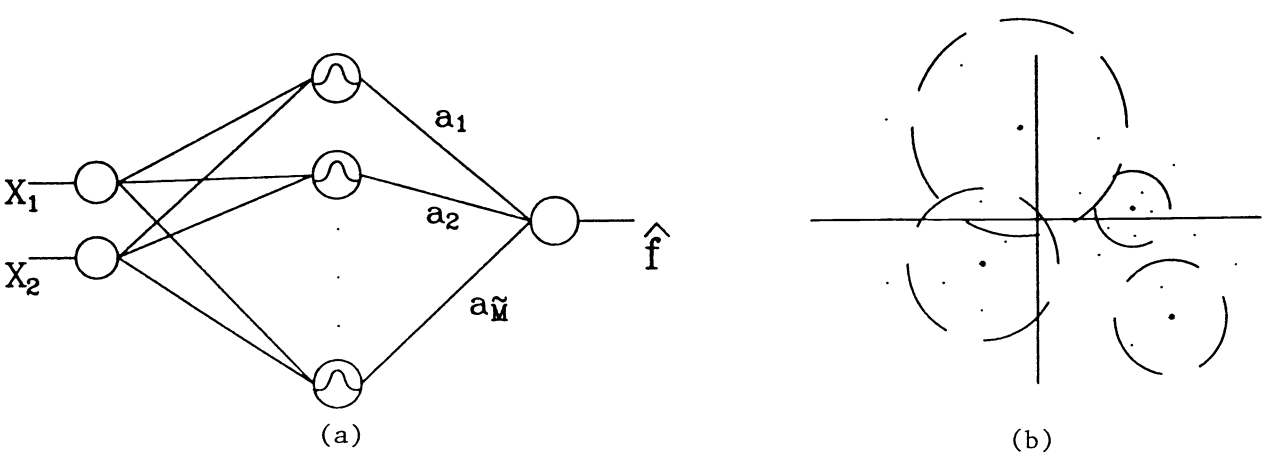


(a)



(b)

Figure 1 : (a) Definition of RBF units and linear units. (b) General RBF network.



(a) (b)

Figure 2 : (a) RBF network for estimating a scalar function of two variables. (b) Example input space configuration resulting from the network of (a).

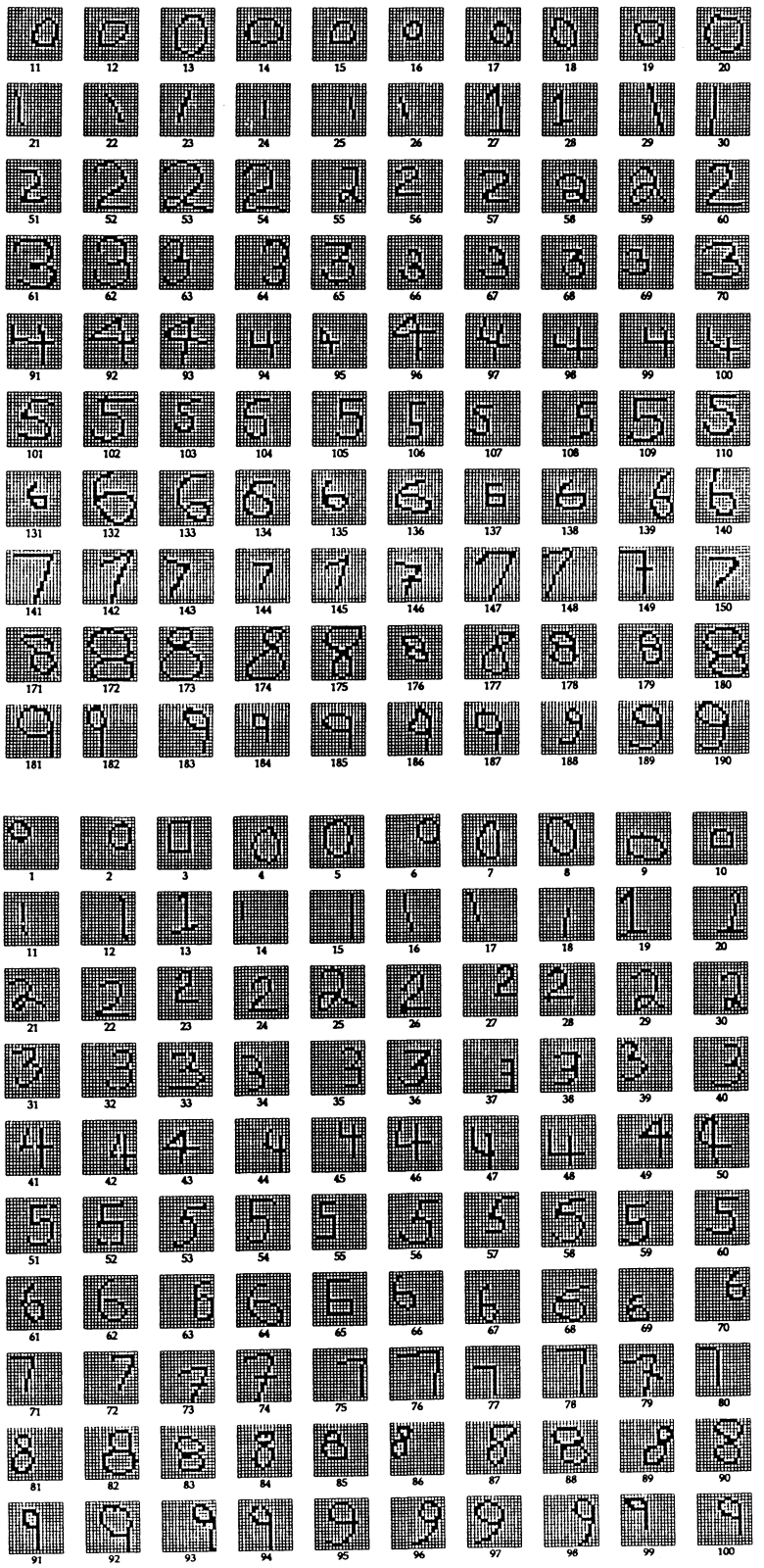


Figure 3 : Example of handwritten numerals from (top) the training set and (bottom) the testing set used in the optical RBF experiment.

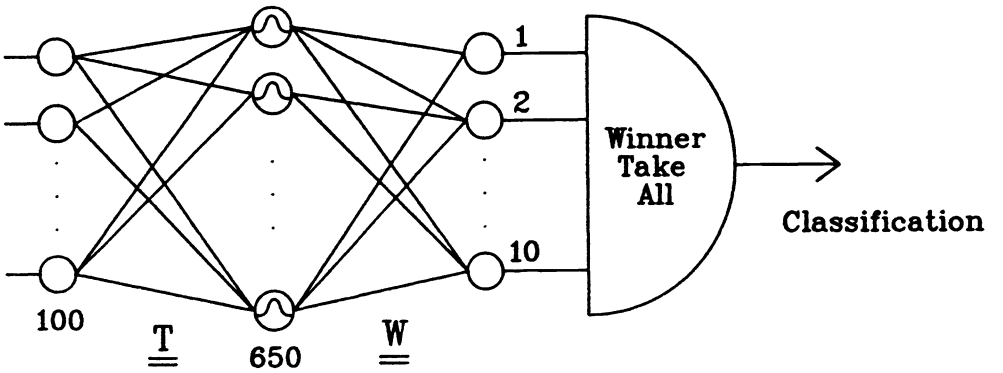


Figure 4 : RBF network for handwritten digit recognition.

$\tilde{\sigma}$	Training Set (Correct out of 650)	Testing Set (Corrrect out of 300)
0.5	650	226
0.7	650	257
1.0	650	266
1.2	650	267

Table 1 : Classification results obtained using a 1-nearest neighbor rule for training the RBF widths.

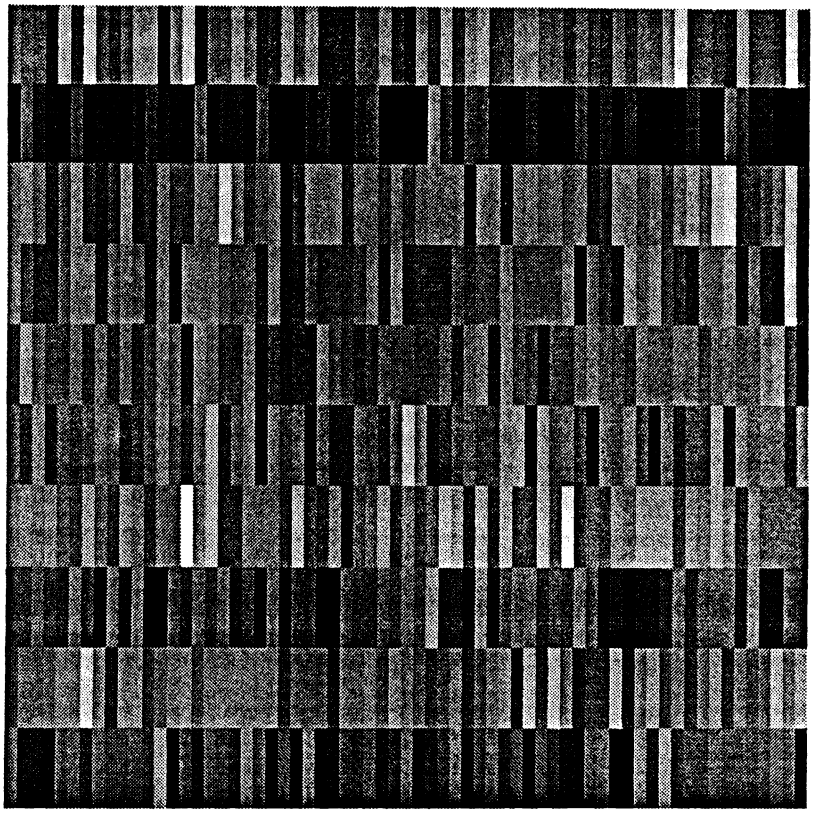


Figure 5 : RBF widths computed using the 1-nearest neighbor rule with $\tilde{\sigma} = 1.2$.

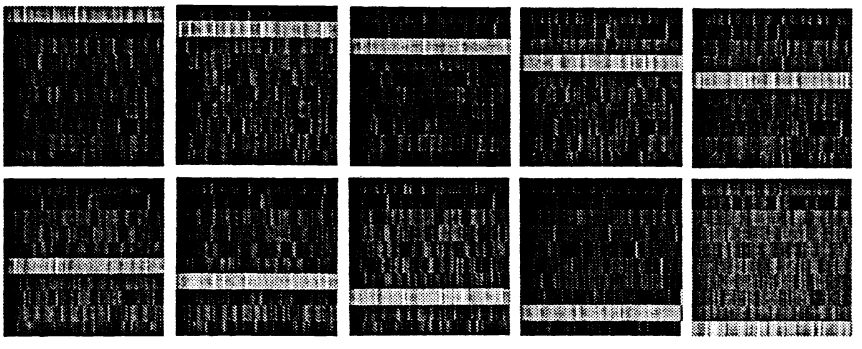


Figure 6 : Second layer weights computed using the perceptron algorithm after initialization with the binary address algorithm. The weights for neurons 1-10 appear consecutively from left to right and top to bottom.

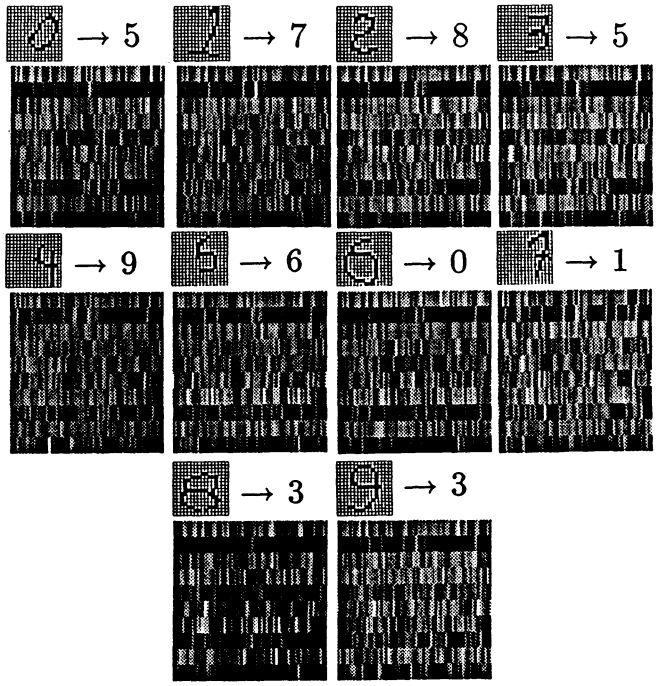


Figure 7 : Examples of misclassifications using the RBF classifier.

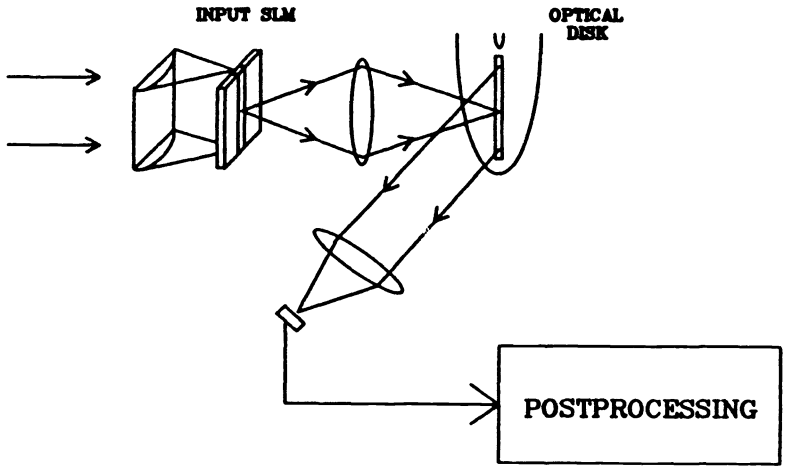


Figure 8 : Optical system used to compute the distance between an input and an array of stored templates.

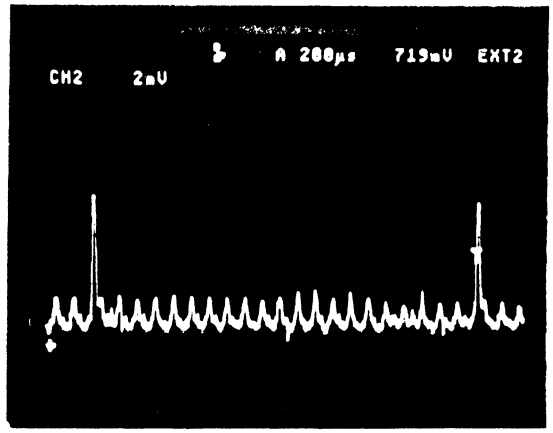


Figure 9 : Example of raw detector output indicating the optically computed inner products.

Class	Experiment	Simulation
0	25	29
1	28	29
2	28	28
3	25	27
4	28	23
5	20	25
6	25	24
7	23	28
8	24	25
9	22	29
Total	248	267
	83%	89%

Table 2 : Performance comparison between optical RBF classifier and simulation.

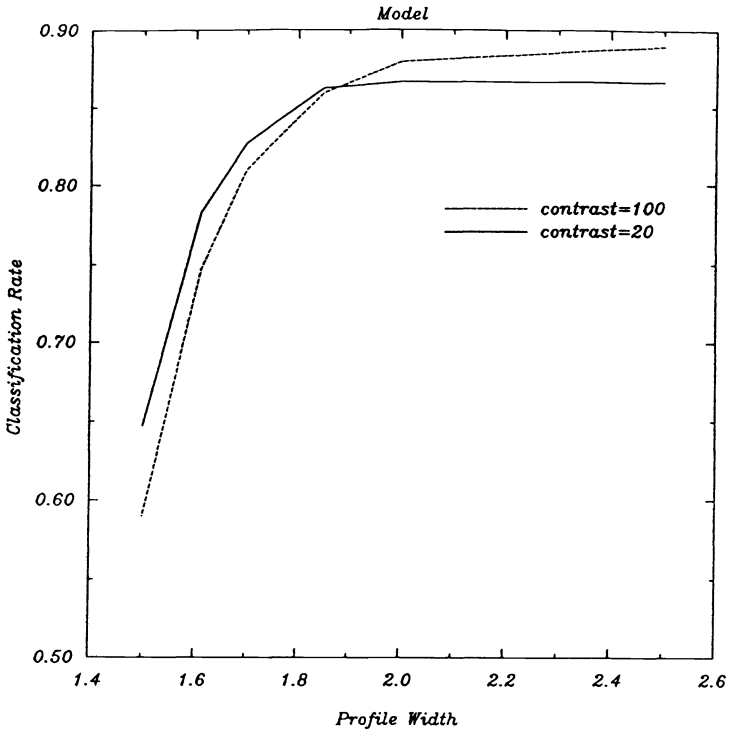
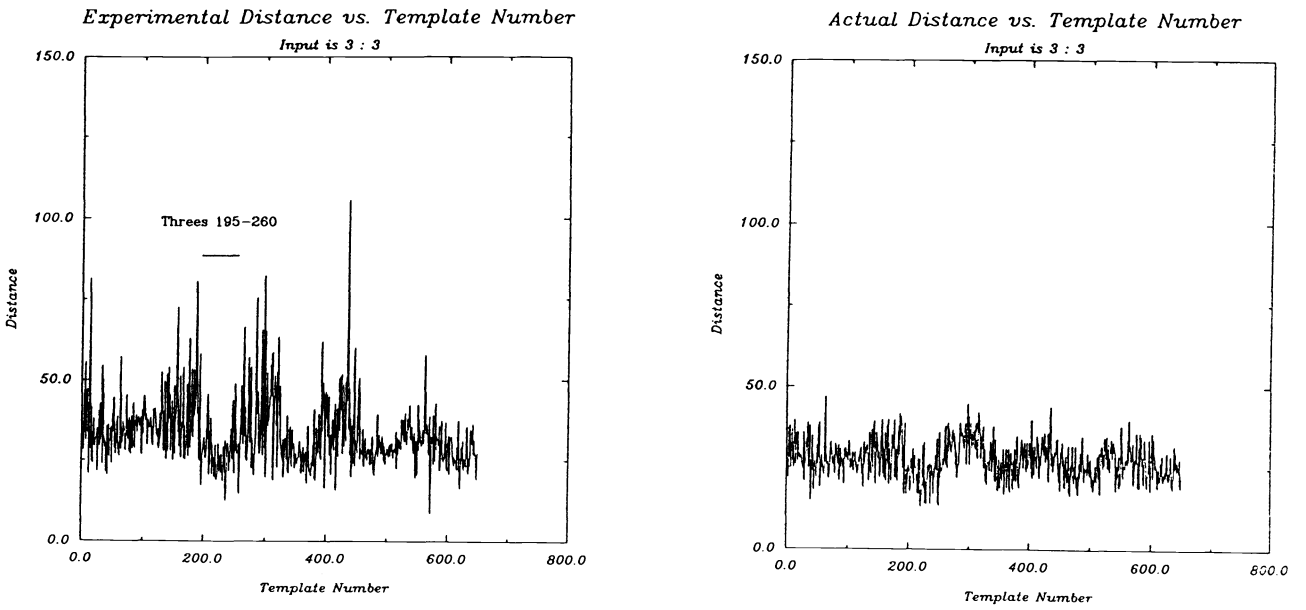


Figure 10 : Predicted recognition rate vs. illumination profile width.



(a) (b)

Figure 11 : (a) Experimental and (b) actual distance vs. template number for a single input image.