Capacity of an RCE-based Hamming Associative Memory for Human Face Recognition

Paul Watta

Department of Electrical & Computer Engineering
University of Michigan-Dearborn
Dearborn, MI 48128
watta@umich.edu

Xiaoyan Mu, Metin Artiklar, Mehmet Artiklar, and Mohamad H. Hassoun

Department of Electrical & Computer Engineering
Wayne State University
Detroit, MI 48202
Hassoun@brain.eng.wayne.edu

Abstract

In this paper we ask: How many people can be stored in a face recognition system which uses template matching as the basis for classification? To do this, we devise and study the capacity of an RCE-based associative memory model suitable for the human face recognition problem. The capacity of the memory is described by two measures: the ability of the system to correctly identify known individuals, and the ability of the system to reject individuals who are not in the database. Experimental results are given which show how the performance of the system varies as the size of the database increases—up to 1000 individuals.

1. Introduction

In the associative memory problem, we are given a memory set of M prototype patterns or input-output pairs of the form $\{(\mathbf{x}^i, \mathbf{y}^i): i = 1, 2, \ldots, M\}$, where $\mathbf{x}^i \in R^N$, $\mathbf{y}^i \in R^p$. The goal is to design a system which can store each of the $(\mathbf{x}^i, \mathbf{y}^i)$ associations in memory so that when pattern $\mathbf{x}^i$ is presented as input (the memory key), the system reliably retrieves the pattern $\mathbf{y}^i$. In addition, the system should retrieve $\mathbf{y}^i$ even when $\mathbf{x}^i$ is corrupted with various types of noise. But when the memory key is not sufficiently close to any of the stored prototypes, the system should reject the input as being noise (Hassoun, 1995).

The human face recognition problem can be seen as a heteroassociative memory whereby the memory set consists of a database of face images paired with the identity (name) of each individual. The input key is then a two-dimensional face image, and the retrieved pattern should be one of the following:

(1) the name of the person represented by the image

(2) reject state (either the input image is not a face, or else, it is a face of an individual who is not in the database).

The Hamming associative memory (or nearest neighbor classifier) is the simplest type of memory, whereby one stores in memory all of the fundamental input-output pairs $(\mathbf{x}^i, \mathbf{y}^i)$. Then for a given input key **x**, the system simply determines the closest matching stored input pattern $\mathbf{x}^{i^*}$ and then outputs the corresponding output pattern $\mathbf{y}^{i^*}$. The Hamming network can be modified to provide for a reject or no-decision state when the input pattern is not sufficiently close to any of the stored prototypes (Watta and Hassoun, 2001), as well as modified to allow for local and parallel distance computations (Ikeda, Watta, Artiklar, and Hassoun, 2001).

Of course to determine the distance between two patterns, a suitable metric is needed. We will denote the distance between patterns $\mathbf{x}_1$ and $\mathbf{x}_2$ by $d(\mathbf{x}_1, \mathbf{x}_2)$. In this paper, d will be the Euclidean metric, although any other suitable metric could be used, as well.

Many face recognition systems rely on the Hamming network to perform the classification.

For example, in a typical eigenface approach (Turk and Pentland, 1991; Belhumeur, Hespanha, and Kreigman, 1997), the input images are mapped onto a lower dimensional eigenspace, but once that mapping is done, a Hamming network is used to match the vector of eigen coefficients of the input to the closest stored database pattern.

The performance of an associative memory is measured by its *capacity*, which is how many patterns can be reliably stored in the system. There are several problems with the notion of capacity that is typically used in the associative neural memory literature (see Hassoun, 1993 for a review). First, capacity is typically expressed as a function of system dimension. For practical problems, though, the user does not have too much control over the system dimension. Furthermore, at least for image processing problems, there is a certain point beyond which increasing system dimension will have no effect on the capacity of the system. For example, increasing the dimension of a face image from 1000×1000 to 1200×1200 (a 69% increase in size) may have no noticeable effect on system capacity.

Second, capacity is usually reported for binary input patterns which are randomly generated and uncorrelated. For face recognition, though, the patterns are highly correlated; for example, all the patterns are expected to have two eyes and a nose and a mouth, and in relatively the same position (assuming the images are suitably normalized in terms of position, etc.).

In this paper, we will measure capacity by measuring the results on 2 different types of experiments:

1. **Correct classification experiments**. Measure the ability of the system to correctly recognize and identify individuals who are in the database.

2. **False positive experiments**. Measure the ability of the system to reject all images of individuals who are not part of the database.

Many face recognition algorithm designs are trained and tuned to perform well on one of these face recognition problems. However, it is clear that a practical system must have good performance on both. And of the two, the false positive problem presents a greater challenge. For example, a face database may contain hundreds or thousands of individuals. But the pool of potential false positive candidates numbers in the billions—all the rest of the people on the planet!

In a practical face recognition system, there are two major computational stages. First, a preprocessing stage is used to detect the face in a larger image, extract just the face portion of the image, and then normalize the scale, position, and rotation of the face image. Second, a recognition stage is used to classify the input as either a known individual or else reject the image.

In this work, we want to study the second stage of the process: the classification stage. Hence, we will use a database of face images that requires little preprocessing. That is, in each image, the face is centered in the image, the lighting is controlled, and there is little rotation, shift, and scaling effects. We will however consider the effect of facial expression, and so the database contains several samples (with different facial expression) of each subject.

The results presented here can be seen as a baseline measure of what can be achieved with state-of-the-art systems which have been used in the FERET tests (Phillips, Wechsler, Huang, and Rauss, 1998; Phillips, Moon, Rizvi, and Rauss, 2000). Note that these systems perform extensive amounts of preprocessing to normalize the image, and so one always wonders to what extent the final results are due to the efficiency/deficiency of the preprocessing.

The remainder of this paper is organized as follows. In the next section we will show how a Hamming network (Hamming, 1986) suitable for face recognition can be implemented using a (modified) RCE network. We describe the architecture of the network as well as a training method whereby the parameters of the system can be determined from the available training face images. In Section 3, an algorithm is presented which provides invariance to small amounts of image translation. In Section 4, we describe the database of face images that we collected for these experiments. In Section 5, we present experimental results for the RCE network and show how the performance of the system scales with database size. Finally, in Sections 6 and 7, we analyze the results and outline future work.

2. An RCE-type Hamming Network

In this section we describe an RCE-type neural network which operates as an associative memory (Hamming network) suitable for the face recognition problem.

Suppose that the database of face images *DB* contains *K* different face image samples of each of the *M* individuals (for example, different facial expressions or lighting conditions, etc.):

$$DB = \{\mathbf{x}_{mk} : m = 1, 2, \ldots, M;\ k = 1, 2, \ldots, K\}$$

Typically, the RCE network is a unit allocating system whereby the number of units is increased as training proceeds (Reilly, Cooper, and Erlbaum, 1982; Hasegawa, et al., 1996). We will simplify the training by just allocating one hidden unit for each prototype image in the database. Effectively, this network acts as a nearest neighbor classifier (with *K* prototypes per class) with rejection capabilities. The network structure is shown in Figure 1.

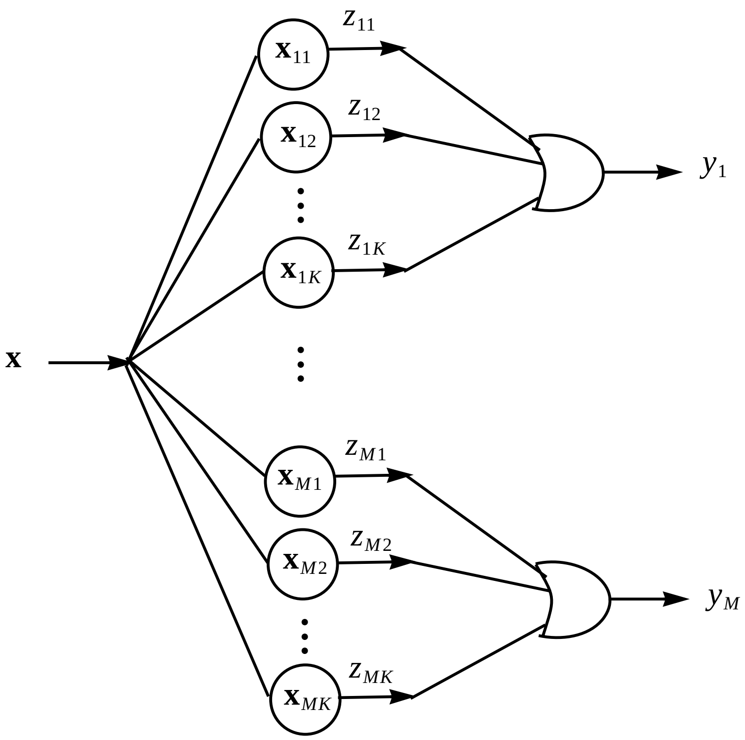

Figure 1. Architecture of the RCE network.

Associated with each hidden layer neuron in Figure 1 is a threshold or radius which is used to determine whether the input is sufficiently close to the prototype so that the unit should fire. The output z_{mk} of the *mk*th hidden unit (with radius r_{mk}) is given by:

$$z_{mk} = \begin{cases} 1 & \text{if } d(\mathbf{x}_{mk}, \mathbf{x}) \leq r_{mk} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Clearly, since the outputs y_m are obtained by ORing together the outputs of the hidden units $z_{m1}, z_{m2}, \ldots, z_{mK}$, then $y_m = 1$ if any of these hidden units fire; otherwise $y_m = 0$. For an input key of say the *m*th person, we want the output vector **y** to consist of all 0's except for a single 1 at the *m*th position.

The system will reject the input in the following cases: (1) More than 1 output unit fires, and (2) none of the output units fire.

For the *mk*th hidden unit, the radius parameter r_{mk} is computed as follows. First, we compute the (between-class) distances between $\mathbf{x}_{mk}$ and all the other images in the database (excluding images of person *m*):

$$d_{ij} = \{d(\mathbf{x}_{mk}, \mathbf{x}_{ij}) : i = 1, \ldots, M; i \neq m; j = 1, \ldots, K\} \qquad (2)$$

And then choose the minimum such distance

$$d_{i^* j^*} = \min\{d_{ij}\} \qquad (3)$$

The number $d_{i^* j^*}$ represents the largest radius that unit *mk* can have, because at that radius, another (incorrect) unit (namely, the $i^* j^*$ unit) will also fire. Hence we set the radius r_{mk} as a fraction of that radius:

$$r_{mk} = \alpha d_{i^* j^*} \qquad (4)$$

where $0 < \alpha < 1$. Experimental results (given below) shows that α should be chosen at around 0.8 or 0.85.

3. Shift Invariance

Noise is always present in any practical image capturing system. There are several ways to make the RCE-based Hamming network less prone to errors in small shifts (translations) and rotations in the image. For example, in the elastic band matching, the distance between the input and each of the database images is computed on a rectangular grid of points and neighbors of the grid points (Duc and Fischer, 1999). The collection of grid points which yields the smallest distance is chosen (deformed template).

Figure 2 shows the distance between the two identical face images as the first image is shifted in various directions (up and down; left and right) and compared to the first. Notice that even a 1-pixel shift in the image can yield a large change in the distance.

In previous work, we developed a simple and computationally efficient method which provides invariance to small amounts of image shift (Artiklar, Hassoun, and Watta, 1999). The method involves determining an optimal sequence of shifts to reduce the Euclidean distance between two (different) images. The optimal path is determined using a greedy algorithm of locally moving in the best direction at each step.

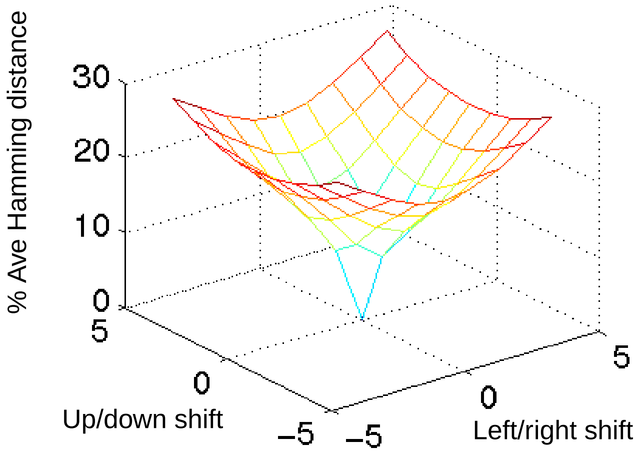

Figure 2. The distance between the two identical images as the first image is shifted in various directions (measured in pixels) and compared to the second.

A schematic diagram of such a path is shown in Figure 2. In this table, each number represents the (normalized) distance between two images if one of the images is shifted by a certain number of pixels. The center of the table (with value 17.7%) is the distance between the two images with no shift. As the input image is shifted 1 pixel down, the distance drops to 13.9%. From there, shifting the input image one pixel to the right results in a distance of 12.4%. This process continues until the distance can no longer be decreased by shifting in any of the 8 surrounding directions (or until a maximum number of steps has been taken). Here, the shifting process decreased the distance between the two images from 17.7% to 10.1%.

Unfortunately, this process can also improve the distance for images of different people. Of course, for recognition purposes, it is desired that the distance between images of different people be as large as possible. The conjecture here (supported by extensive simulations) is that even though the shifting process decreases the distance between different people, it tends to do so by a lesser amount than the distance improvement for images of the same person. Hence, there is an overall increase in the separation ability of the classifier.

29.4	28.5	27.5	27.3	27.3	28.5	29.4	30.5	31.5
27.9	27.0	25.8	25.4	25.6	26.4	27.8	29.1	30.6
27.0	25.4	24.2	23.5	23.7	25.0	25.9	27.6	29.9
25.8	23.7	22.0	21.4	21.4	22.8	24.5	26.8	29.2
25.0	22.3	19.6	18.0	17.7	18.4	19.3	23.2	27.6
24.0	21.2	17.2	14.2	13.9	12.4	19.9	21.7	25.1
23.7	19.7	15.7	11.3	12.6	11.6	17.3	20.2	23.2
24.2	20.8	16.9	12.9	12.8	10.9	10.1	13.3	19.9
25.4	22.8	19.8	17.5	17.0	19.2	22.0	18.5	27.9

Figure 3. This diagram shows the Euclidean distance (in percent) between a database image and shifted versions of an input image.

Note that it is pointless to try to shift images that are very dissimilar from each other. Hence, to reduce the time it takes to perform the classification, we only allow the top 40 matches to participate in the shifting process. Also note that the shift processing algorithm occurs right before the hidden layer stage (see Figure 1). Hence, for a given input image **x**, all distances $\hat{d}(\mathbf{x}_{mk}, \mathbf{x})$ are computed. The 40 such smallest distances are selected and the input image is shift-processed against each of these 40 closest units to obtain a (possibly) smaller set of distances: $d(\mathbf{x}_{mk}, \mathbf{x})$. From there, Equation 1 is used, and the output is computed.

4. The Database of Face Images

A variety of face databases have already been described in the literature, and a summary of some of these databases can be found in (Chellappa, Wilson, and Sirohey, 1995; Robertson and Craw, 1994; Phillips, Moon, Rizvi, and Rauss, 2000).

A detailed discussion of the construction of the face database that is used in these experiments can be found in (Watta, Artiklar, Masadeh, and Hassoun, 2000). Briefly, the database consists of 1000 different men and women of various ages (between 15 and 65 years old). The images were collected at an urban University and hence the database contains samples from a variety of ethnic backgrounds. In order to minimize the amount of preprocessing that is necessary in order to eliminate complicating effects, such as tilt, rotation, shifting, scaling, and changes in illumination, the images were collected in a laboratory setting. A head brace was used to position the subject's head in front of the camera, but note that no attempt was made to normalize the position of the eyes, etc.

For each person, two groups of 4 training images were snapped, giving a total of 8 training images per person. The first group of 4 images all show a blank facial expression. The second group of 4 images show different facial expressions: blank, smile, angry, and surprised.

Two test images were also collected for each person: a test image where the subject shows a blank facial expression, and a test image where the subject gives an arbitrary or unusual expression to try to fool the recognition system.

Both the database and test images were snapped at a dimension of 82×115 and stored as 8-bit gray scale (256 levels). The 82×115 images were cropped to a size of 72×72 (using a fixed cropping position) in order to eliminate the hair and forehead of the person from the image.

Figure 4 shows a set of 82×115 sample images for one of the subjects in the database, and Figure 5 shows several samples of the cropped 72×72 images.

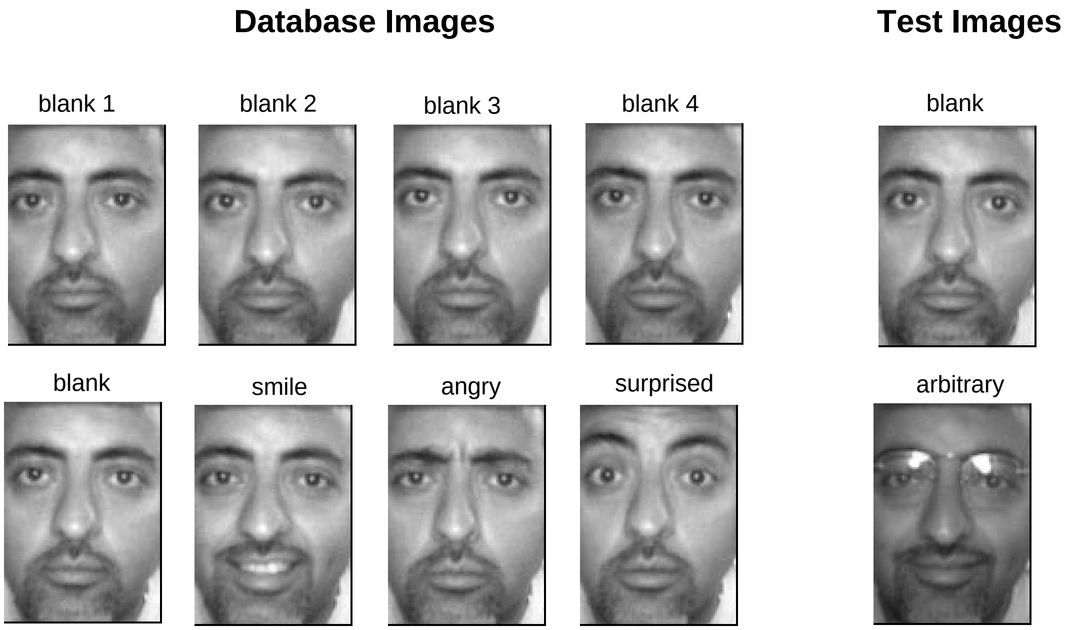

Figure 4. Sample 82×115 images of one of the subjects in the database.

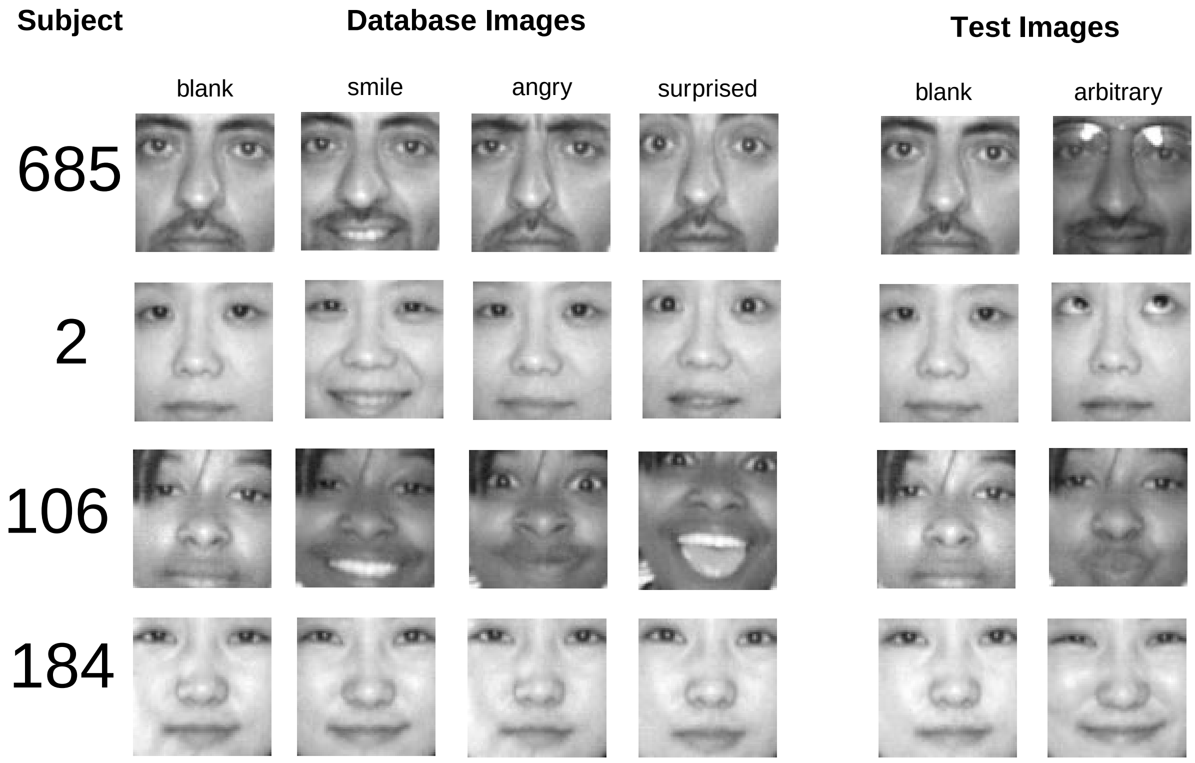

Figure 5. Samples of the 72x72 database and test images for different people (the 4 additional blank images are not shown). In total, there are 1000 individuals in the database.

5. Experimental Results

The database of face images here consists of the 4 facial expression images (blank, smile, angry, and surprised) of each of the M known individuals. So there's a total of $4M$ images in the database, and hence there's $4M$ hidden units in the RCE network.

5.1 Correct Classification Experiments

In the training phase, the database images are used to determine the radius r_{mk}, $m = 1, 2, \ldots, M$; $k = 1, 2, 3, 4$, for each of the hidden units. After training, the system was tested for both correct classification performance and false positive performance. To see how the performance scales with increasing number of subjects in the database, we looked at the performance of the system as the number of people in the database varied from 100 to 1000 individuals.

Num People M	**Blank**		**Arbitrary**	
	Reject	Error	Reject	Error
100	3.4	0	28.4	0.4
200	2.7	0	31.8	0
300	2.7	0	33.4	0
400	3.4	0	36.5	0
500	3.5	0	36.2	0
600	4.0	0.1	37.9	0
700	3.5	0.1	39.3	0
800	4.2	0.1	39.3	0
900	4.2	0.1	40.7	0
1000	4.5	0.1	41.4	0

Table 1. Correct classification results for the blank and arbitrary expression test sets for various number of people in the database.

Table 1 shows the results of the correct classification experiments for the RCE-based Hamming network for both the blank and arbitrary expression test images. Note that each number in Table 1 (except for the $M = 1000$ case) is an average over 5 different trials. In each trial, a different database was (randomly) chosen from the available 1000 individuals.

The results show that the algorithm scales well and for the blank expression test images gives an error rate of 0.1% (or less) with a reject rate of less than 5%. As expected, for the arbitrary expression test images, the system rejects a much larger number. However, there are no classification errors on the arbitrary expression test set.

5.2 False Positive Experiments

Next we ran false positive experiments and tested the system with 200 randomly chosen individuals who are not present in the database to see how well the system rejects these images. In this case, we used all available training and test images (10 images total) for each test person. Note that we could not perform this experiment on the $M = 1000$ database due to lack of additional images, and could only test with 100 people for the $M = 900$ database.

Num People *M*	**False Positive** %
100	2.9
200	1.6
300	1.1
400	1.0
500	1.2
600	0.7
700	0.6
800	0.4
900	0.4

Table 2. False positive results for various number of people in the database.

The results of the false positive experiments are shown in Table 2. Curiously, the number of false positive matches decreases as the number of people in the database increase, and is only 0.4% for $M = 800$ and $M = 800$. The reason for this is that as more images are added to the database, the hidden layer thresholds necessarily become tighter as Equations 2 and 3 are computed over more images. Another reason is that as more and more images are added to the database, it is more likely that more than 1 output unit will fire. But, by our simple decision rule, if more than 1 output fires, we reject the input.

5.3 The Correct Classification/False Positive Trade-off

A plot of both the classification (rejection) and false positive results is shown in Figure 6. From this figure, we see that the general trend in rejection performance increases linearly (after $M = 200$. We did not plot the number of misclassified images, since it was so low: there was at most 1 misclassified image for each test. However, it is expected that as M increases, the number of misclassified images will also begin to rise. A larger database is needed in order to perform those experiments.

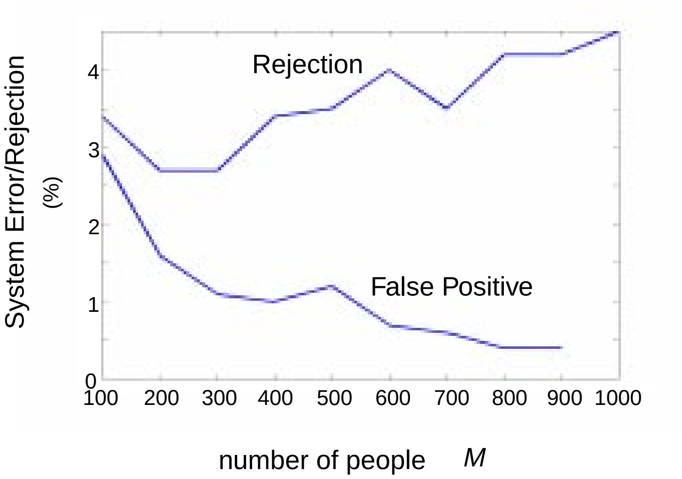

Figure 6. Results of the rejection performance and false positive performance of the system as a function of database size.

5.4 The Effect of Radius Size α

For all the results given above, a threshold scaling parameter of $\alpha = 0.8$ was used. Figure 7 show how the rejection and false positive results vary as α is varied. Of course to get good false positive performance, we require a very small value for α. But setting α too low causes a lot of rejections (in the correct classification experiments) because the condition for a hidden unit to fire (Equation 1) is too stringent. On the other hand, setting α too high also causes a lot of rejections because it is more likely that more than 1 output unit will fire.

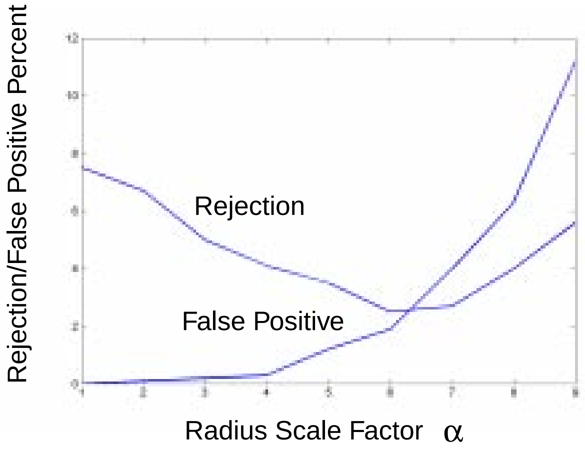

Figure 7. Rejection and false positive performance as a function of α.

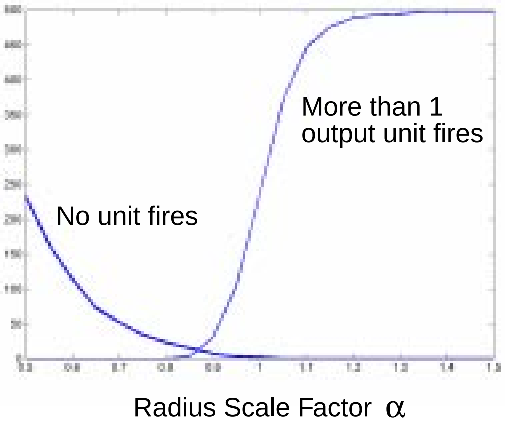

Figure 8. Number of times no unit fires (thick) and more than 1 unit fires (thin) as a function of α. The sum of these is the number of images which are rejected by the RCE network.

To analyze the rejection performance of the system, we measured the number of rejections due to the fact that *no hidden units were activated*, as well as the number of rejections due to the fact that *more than* 1 *output unit was activated*. Figure 8 shows the result of this experiment on a database of individuals. The sum of both of these quantities is the number of images rejected by the system.

5.5 Performance on Different Types of Databases

For all the above simulations, the memory set consisted of the 4 expression images: blank, hint of a smile. Clearly, the best result is to store 8 images per person. However, the price to be paid for using 8 samples per person rather than 4 samples is in terms of storage (double the size) and retrieval time (double the time).

6. Analysis of Results

In order to compare the results obtained here to the results given in the FERET competition (Phillips, Moon, Rizvi, and Rauss, 2000), we ran a nearest neighbor classifier on the 1000-person database and using the blank and arbitrary expression test images. In this case, there is no reject state, and we simply take the best matching image (nearest neighbor) as the output of the system. Since the system is tested with only test images of people in the database, there's only two possible outcomes: the test images is correctly classified, or it is misclassified. The result we obtained with this nearest neighbor classifier is: 98.7% correctly classified on the blank expression test images, and 85.8% correctly classified on the arbitrary expression test images. The best result reported in the FERET competition and on the FERET face database (Phillips, Moon, Rizvi, and Rauss, 2000) was sl2H(than)-i,done on the FERET images was very effective. (The goal of the pre the images into a standard position and scale, characteristic of our database of images).

Of course, as mentioned above, just looking at the ability of the classifier to identify images of people who are in the database does not provide a good measure of system performance. The system must have a mechanism to reject unknown individuals.

6.1 Rejections

When collecting the images for this database, even though we tried to limit the amount of variation in scale, rotation, intensity, we did not completely eliminate such effects. Figure 9 shows samples of some of the test images that were rejected by the RCE network. In these cases, it is easy to see why the blank test image was rejected. For subject 80, the blank database image is slrotated whereas the blank test image is not. For subject 279, there is a difference in intensity between the blank test image and all of the database images. For subject 332, the blank test image shows the person looking up and to the left whereas the database images show the subject looking straight. In subject 454 there are intensity variations as well as expression variations.

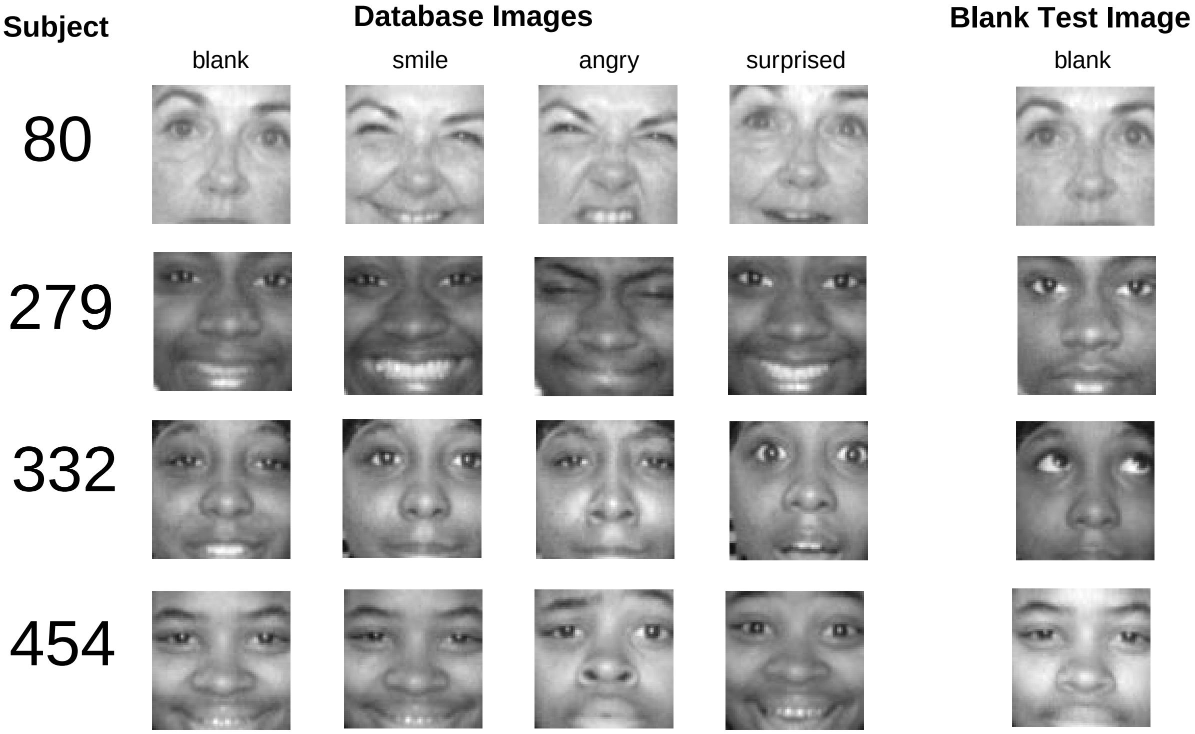

Figure 9. Samples of the database and blank test images which show various amounts of noise: changes in rotation, illumination, and facial expression. These test images were rejected by the RCE network.

6.2 Misclassifications

As mentioned above, there was only one blank test image that was misclassified. Figure 10 shows the misclassified test image (subject 260) as well as the corresponding database images, and the database images of the individual to whom the RCE network wrongly matched (subject 274). In this case, subject 260 is shown smiling in the blank database image, but not in the blank test image. Also, there are intensity variations among all the images for subject 260.

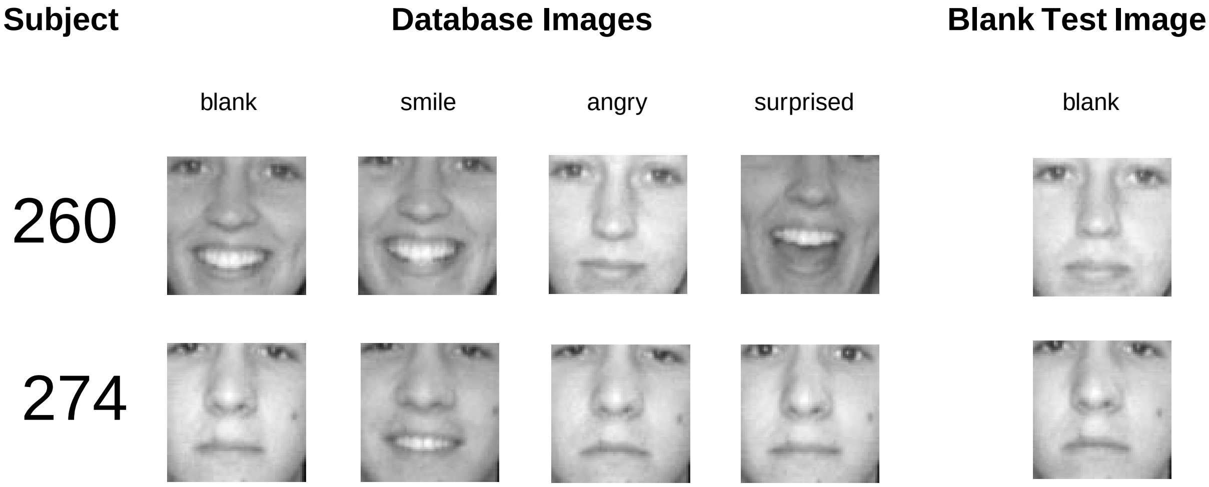

Figure 10. The blank test image of subject 260 was the only image misclassified for all the databases. The blank image was matched to the blank database image of subject 274.

6.3 False Positive Matches

Figure 11 shows some of the cases of erroneous matches in the false positive experiments. In the first case, subject 629's blank database image was matched to subject 510's angry image, and his blank 1 image was matched to subject 510's blank image. In both cases, although the subject's don't look exactly alike, the images have very similar orientation. In the next case, 3 images of subject 774 were matched to subject 400. Here, a human observer can obviously distinguish between the two individuals. Clearly, the radius parameters for subject 774 are too large.

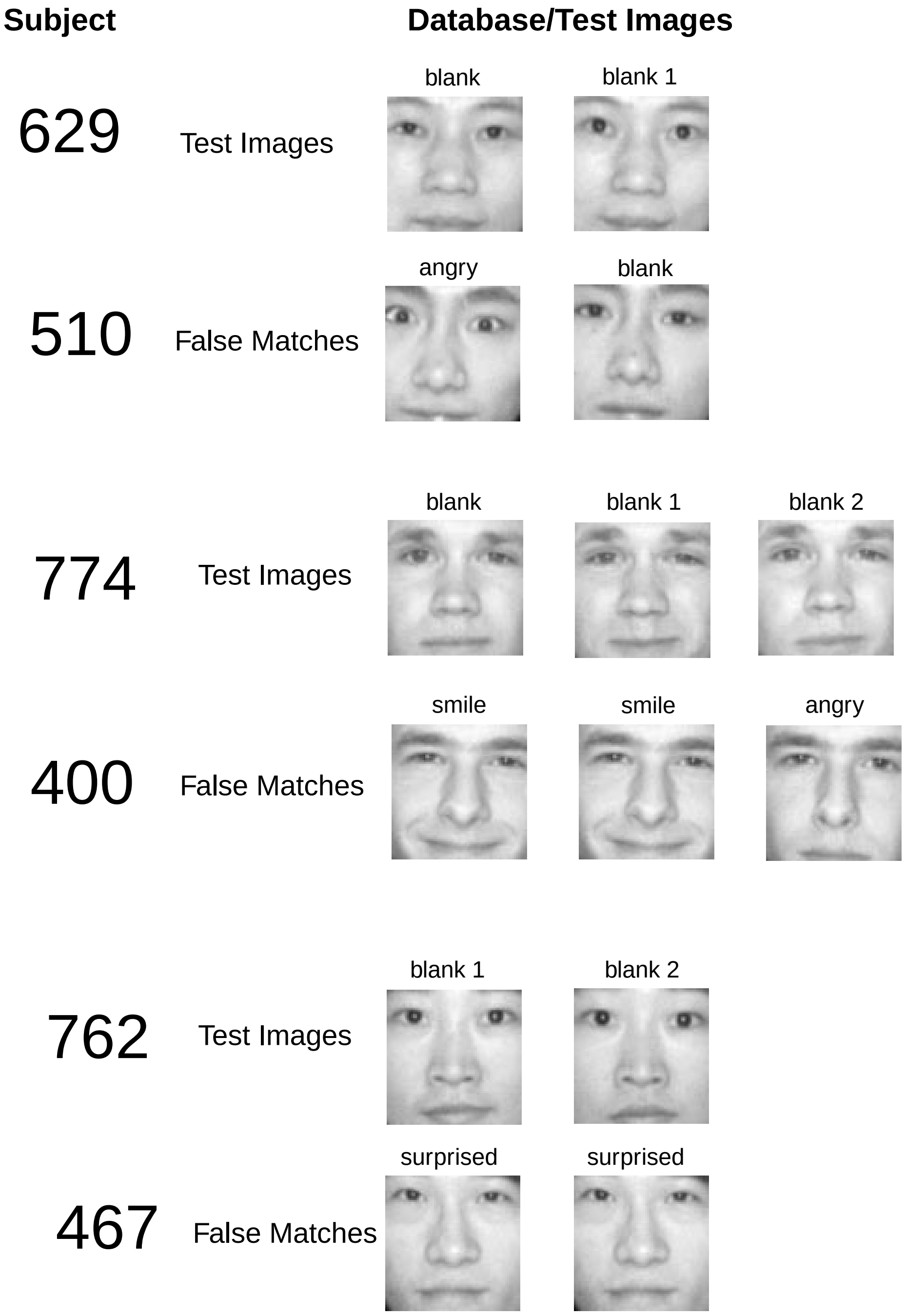

Figure 11. Examples of false positive matches.

6.4 Training Time

Finally, Figure 12 shows how the training time scales as a function of the number of people in the database. Since the training here simply involves computing inter-class distances among the database images, the training scales linearly with the number of people in the database.

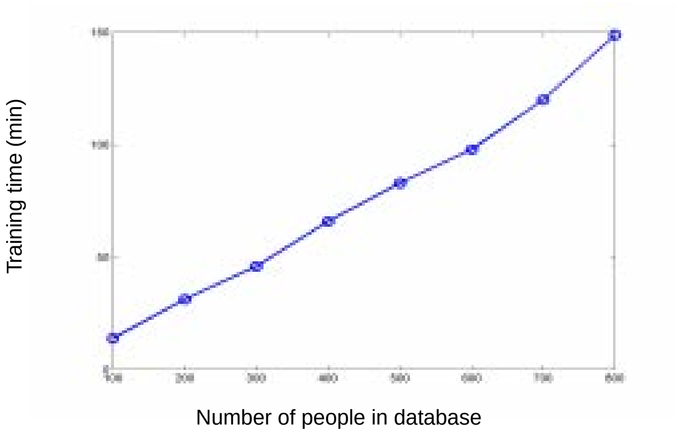

Figure 12. Training time (in minutes) vs. the number of people in the database.

An important design consideration of associative memories is how easy it is to add and delete associations from the memory set. For the given RCE network, it is easy to delete individuals from the database—simply delete the hidden units allocated to that person (or else shrink the radius to 0).

In terms of adding individuals to the database, the fact that the number of false positive matches decreases as M increases (see Figure 6), allows us to use a modular approach for constructing and training large databases. For example, suppose we start with a $M = 400$ person database. Now suppose we want to add an additional 400 people to the database. Rather than retraining on the whole 800 person system, we can just formulate and train a separate 400-person network to handle the new patterns. Then the outputs of the original network and the new network can be combined with a simple decision: If both networks reject the input image or both networks classify the input image, then the image is rejected. However, if one of the networks classifies the pattern and the other network rejects it, then we accept that classification. Figure 13 shows a schematic diagram of this modular design.

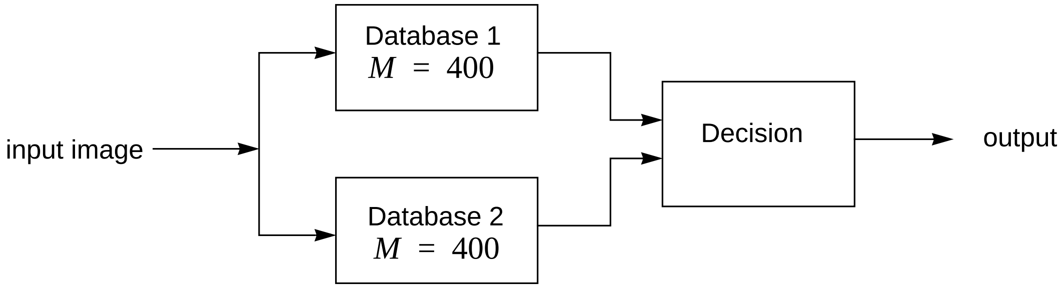

Figure 13. Block diagram of the modular construction.

Note that we expect to use such a modular approach for large size databases ($M = 1000$ and above) where the number of false positive matches are sufficiently low. We did, however, test the idea on the $M = 400$ database. In this case, two 400-people RCE networks were independently trained and then the outputs combined with the simple decision rule described above. The results are shown in Table 4. The correct classification results show remarkable similar performance. The false positive performance is a bit worse for the two 400 person network, but not by much.

Memory set	**Blank Test** %		**Arbitrary Test** %		**False Positive** %
	Reject	**Error**	**Reject**	**Error**	
800 person database	4.2	0.1	39.3	0	0.40
Two 400-person databases	4.6	0.4	37.6	0.1	0.65

Table 4. Performance of the modular RCE network (two 400-person databases) compared to the full network (a single 800 person database).

7. Summary

This paper investigated the capacity of an RCE-based associative memory for the human face recognition problem. Capacity here is defined in terms of the trade-off between correct classification performance and false positive performance. In the correct classification experiments, we want to minimize the number of misclassifications as well as the number of rejections. Simultaneously, though, we want to minimize the number of false positive matches for people who are not in the database. The results given in Figure 7 show that it is not possible to simultaneously optimize both capacity measures. In terms of design, one would specify the maximum amount of misclassification or false positive matches that the system should produce, and then one would determine an appropriate database size.

When using the RCE network on databases of up to 1000 people, the results indicate that the system can achieve a correct classification rate of over 99%, with a rejection rate of about 5%, and a false positive rate of less than 1%. Of course when storing random and uncorrelated images, the capacity of the Hamming associative memory is much higher (Hassoun and Watta, 1996). With faces, though, the significant degree of correlation has the effect of dropping the capacity to only a few thousand.

In future work, we will compare the results of this classifier with other pattern recognition systems (Duda, Hart, and Stork, 2000), as well as other neural net (Hassoun, 1995), and associative memory models (Hassoun, 1993). In addition, we will collect a larger database of face images in order to determine how the system performs as *M* is increased beyond 1000.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under contract ECS-9618597.

References

1. Artiklar, M., Hassoun, M., and Watta, P. (1999). “Application of a Postprocessing Algorithm for Improved Human Face Recognition,” *Proceedings of the IEEE International Conference on Neural Networks*, IJCNN-1999, July 10-16, 1999, Washington, DC., Paper #JCNN 2166.

2. Artiklar, M., Masadeh, A., Hassoun, M. H., and Watta, P. (2000). “The Effect of Expression in a Database of Face Images,” *Proceedings of the 43rd Midwest Symposium on Circuits and*

Systems, August 8-11, 2000, Lansing, MI.

3. Belhumeur, P., Hespanha, J., and Kreigman, D. (1997). "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 711-720.

4. Duc, B. and Fischer, S. (1999). "Face Authentication with Gabor Information on Deformable Graphs," *IEEE Transactions on Image Processing*, **8**(4).

5. Duda, R., Hart, P., and Stork, D. (2000). *Pattern Classification*, Second Ed., John Wiley & Sons, New York.

6. Hamming, R. (1986). *Coding and Information Theory.* Prentice-Hall, Englewood Cliffs, NJ.

7. Hasegawa, A., Shibata, K., Itoh, K., Ichioka, Y., and Inamura, K. (1996). "An Adaptive Neural Network: Application to Character Recognition on X-ray Films," *Neural Networks*, **9**(1), 121-172.

8. Hassoun, M. H., ed. (1993). *Associative Neural Memories: Theory and Implementation*. Oxford University Press, New York.

9. Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, Mass.

10. Ikeda, N., Watta, P. B., Artiklar, M., and Hassoun, M. H. (2001). "Generalizations of the Hamming Net for High Performance Associative Memory," accepted for publication in *Neural Networks*

11. Phillips, P., Moon, H., Rizvi, S., and Rauss, P. (2000). "The FERET Evaluation Methodology for Face-Recognition Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(10), 1090-1104.

12. Phillips, P., Wechsler, H., Huang, J., and Rauss, P. (1998). "The FERET Database and Evaluation Procedure for Face-Recognition Algorithms," *Image and Vision Computing J.*, **16**(5), 295-306.

13. Reilly, D. L., Cooper, L. N., and Erlbaum, C. (1982). “A Neural Model for Category Learning,” *Biological Cybernetics*, **45**, 35-41.

14. Turk, M., and Pentland, A. (1991). “Eigenfaces for Recognition,” *J. Cognitive Neuroscience*, **3**(1), 71-86.

15. Watta, P., Artiklar, M., Masadeh, A., and M. H. Hassoun, (2000). “Construction and Analysis of a Database of Face Images which Requires Minimal Preprocessing,” *Proceedings of the IASTED Conference on Modeling and Simulation*, MS-2000, May 15-17, 2000, Pittsburg Pennsylvania, 465-469.

16. Watta, P. and Hassoun, M. H. (2001). “Generalizations of the Hamming Associative Memory,” *Neural Processing Letters*, 13(2), 183-194.