Implementation and analysis of a facial recognition system combining a neural network and linear subspace model

E.J. Meyer, B.T. Maharaj, MIEEE

Abstract—**Many facial recognition and detection algorithms have been developed and tested according to accuracy, speed, size and environment. These tests are usually independent and lacking in terms of thoroughness. Each algorithm has its advantages and disadvantages. This paper describes the improved results obtained from the unique implementation of combining the neural network technique in facial detection and the linear subspace model in facial recognition. The advantage of this approach as shown by the results in this paper is to improve or combine the advantages of each algorithm and minimize the disadvantages.**

Index Terms—**Eigenfaces, eigenvalues, FPGA, linear subspace model, neural network**

I. Introduction

THE use of facial recognition in the form of biometric authentication can be used for a number of access control applications. Various algorithms have been developed to enable the implementation of such systems. This paper describes the implementation of two algorithms in a real time facial recognition system. The limiting parameters associated with the algorithms and environment are identified and measured. These measurements serve as an indication of the advantages and disadvantages of combining neural networking and linear subspace modeling to perform face detection and recognition.

Many different face detection algorithms have been developed and tested [1]. These algorithms have been developed from very simple, yet processor intensive functions to well developed artificial intelligence functions. A key parameter in face detection is the accuracy at which an algorithm can detect a face. Most algorithms are very limited in terms of their training set and do not allow for large variations in faces. Most algorithms are usually limited to detecting a face within a blank background. Neural networking is an artificial intelligence technique which allows for a larger amount of variation and allows for background to exist within a portrait image [1]. A face detection algorithm that has achieved high accuracies involving the use of neural networking can be found in [2].

A technique for face recognition that performs well and makes use of a linear subspace model can be found in [3]. This is a relatively simple model to implement and can be adjusted to improve on accuracy and speed. It makes use of eigenfaces to recognize an individual. This technique is very poor for face detection, but performs relatively well for face recognition.

There are many problems that have been identified with face recognition algorithms [4]. The dimensionality of a face recognition algorithm determines the degree of accuracy and number of training sets used. This describes the knowledge or information known a priori by the algorithm. The images that are used to test the algorithms on usually contain a large amount of noise such as background, lighting and variability of faces. Most algorithms that have been developed usually focus on improving one of these components and neglect many other factors. A study was conducted on a linear subspace model face recognition algorithm with respect to luminance [5]. This paper takes factors such as dimensionality, variability of faces and luminance into account and determines the effect of lighting on the accuracy of the combined algorithms.

This paper introduces the concept of a complete facial detection and recognition system making use of a novel approach of neural networking for detection and a linear subspace model for recognition. The advantage of making use of a neural network for face detections is because it allows one to train the network with face and non-face data. This improves the accuracy of the detection as it allows for greater face variation and a large variety of portrait background [1]. The advantages of using a linear subspace model for face recognition is because it allows for a practical solution to the recognition problem, it is fast, relatively simple and works well in somewhat constrained environments [3]. The use of the neural network reduces the limitation of the linear subspace model working within a constrained environment. The tradeoff in combining the two algorithms that were used in this paper is in processing power. The processing power improvements in today's competing markets, with the development of nanotechnology, will no longer be a limitation. This would result in the time taken for the complex

This work was supported by the NRF and University of Pretoria, South Africa.

E.J. Meyer and B.T Maharaj are with the Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Lynnwood Road, 0002, South Africa (tel: +27 82 568 4649; fax: +27 11 363 3344; e-mail: jonathan.meyer@kumbaresources.com).

mathematical computations required by the two algorithms used to become less. This means that the techniques used in this paper allow for a very powerful tool to perform facial detection and recognition.

In this paper section II describes the design choices made and implemented both in hardware and software for the facial detection and recognition system. Section III describes the new work that was accomplished in order to make the system practical to implement and to improve the algorithms that were used. The results in section IV illustrates the different ways in which the system and its subsystems were tested and verified. Section V mentions some of the further work that can be done as a result of this research and finally section VI concludes the paper.

II. Design Choices and Implementation

The concept design used to develop and test a face recognition system by the Department of Electrical, Electronic and Computer Engineering of the University of Pretoria is summarized in Figure 1.

A digital camera was used to take photographs of people to be recognized by the system. The image processing is performed on this digital image data. The use of a digital camera reduces the interface requirements as opposed to an analog camera.

An FPGA was used to interface with the digital camera and perform image buffering and image compression. It makes use of digital logic to perform the processing. Since the camera is digital and the FPGA functions on digital logic the interfacing between the two did not create any complications and there is no loss of data or quantization noise created during sampling. The FPGA also acts as a network controller and will allow the image data to be transmitted over a TCP/IP network.

The face detection and face recognition algorithms are processed on a PC acting as a server connected to the FPGA client. Considerable processing power is required to implement the combined algorithms into a real time system.

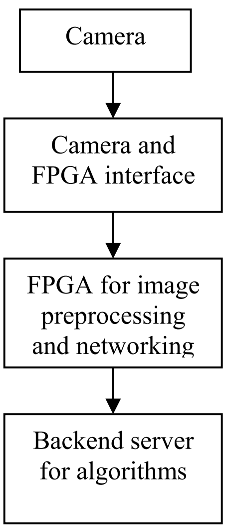

Fig. 1. Approach used to develop a face recognition system to be used for measurements of limiting parameters.

The backend server processes each image received from the FPGA by firstly detecting a face within the image. If no face is detected the image is discarded. If a face is detected the face is processed further such that it can be recognized with a database of face images. If a match occurs the matching face is displayed on the backend server.

A neural network similar to [2] was implemented to perform the face detection. This technique is very robust and can be trained a number of times to achieve a desired accuracy. A few modifications to the neural network architecture were made and are described in section IV of this paper. These modifications made the algorithm more practical to implement.

A linear subspace model similar to [3] was implemented to recognize the face that was detected by the neural network with a database of face images. This technique allows one to represent a known image as a set of characteristic features. An image matrix $\mathbf{I}$ of size M x N is converted into an image vector $\mathbf{\Gamma}$ of length K by concatenating each column to the previous column in the matrix. Once a number of facial images have been converted into image vectors, each image is approximated by a weighted sum of their eigenvectors. This is represented mathematically by

$$\hat{\mathbf{\Phi}} = \sum_{j=1}^{K} w_j \mathbf{u}_j \qquad (K \ll MN) \tag{1}$$

where $\hat{\mathbf{\Phi}}$ is an approximation of the mean-adjusted image $\mathbf{\Phi} = \mathbf{\Gamma} - \boldsymbol{\mu}_{face}$. w_j indicates the weights or features of a face and $\mathbf{u}_j$ indicates the eigenvectors or eigenfaces. K is selected such that a specific accuracy or approximation is achieved.

The eigenvectors and eigenvalues that have been calculated by the training set can then be used when recognizing an unknown image. By representing an unknown face image onto the eigenvectors obtained from the training set one can obtain a set of weights or features describing the image. For each known image l the minimum error

$$e_r = \min_l \left\| \mathbf{\Omega} - \mathbf{\Omega}^l \right\| \tag{2}$$

is determined. $\mathbf{\Omega}$ represents the unknown face image weights and $\mathbf{\Omega}^l$ represents the known training set face image weights. The image within the training set with the lowest minimum error, given within a certain threshold, can then be assumed to be the same as the unknown image.

III. Development Work

In order to improve the throughput of the system the neural network was applied to a compressed image. The image compression was taken from [6] where the image is segmented into blocks of 10x10 pixels. Each block is then averaged such that it is reduced to one pixel. The equation

$$f_c(x,y)=\left[\sum_{i=10x}^{10x+9}\sum_{j=10y}^{10y+9} f(i,j)\right]\Big/100 \qquad (3)$$

describes the lossy compression procedure where $(x,y)\in(0..63,0..47)$. This lossy compression is allowed for the compression of a 640x480 pixel image to 64x48 pixels (f is the uncompressed image and f_c is the compressed image).

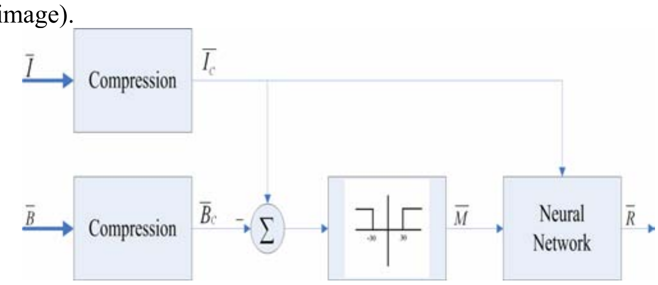

Fig. 2. Procedure used to improve system throughput.

A mask was created such that the objects in front of the camera could be extracted and processed upon. This allows for improved throughput as a large amount of irrelevant information is discarded. Figure 2 illustrates the image masking procedure used. Both the background $\overline{B}$ and image frame $\overline{I}$ are compressed by making use of (3). The compressed image frame $\overline{I}_c$ is then subtracted from the compressed background $\overline{B}_c$ and processed by a threshold function. By firstly compressing each image and then subtracting them improves the system performance because of fewer subtraction calculations that take place. The threshold function can be adjusted to alter the difference deviation and thereby increasing or decreasing the size of the image mask. This creates a binary image $\overline{M}$ representing the objects in front of the camera. The neural network is then applied to the compressed image and to the regions within the image mask.

The neural network designed makes use of an input window of 15x24 pixels instead of a 20x20 pixel window described in [2]. This is due to the fact that a person's face is generally more rectangular than square as described in [1] where the face has a height to width ratio (H:W) of $\frac{1+\sqrt{5}}{2}$. The architecture that was designed can be seen in Figure 3. The network makes use of the sigmoid function

$$\sigma(x)=\frac{1}{1+e^{x}} \qquad (4)$$

where x represents each nodes weighted sum and outputs a value in the range from 0 to 1. Each weight in the middle layer is updated by

$$\Delta w_{jk}(t)=\alpha y_i\delta_k+\beta\Delta w_{jk}(t-1) \qquad (5)$$

where the learning rate α was set to 0.8 and the momentum β was set to 0.95. The image window $\overline{x}$ was normalized by its mean before being processed. The training of this network was obtained from [7] where momentum has been used to speed up the learning process. The input layer was updated in a similar fashion to (5). The network was trained for approximately 3000 epochs by making use of 150 faces and 150 non-face images. Tests were made to determine the accuracy of the trained neural network.

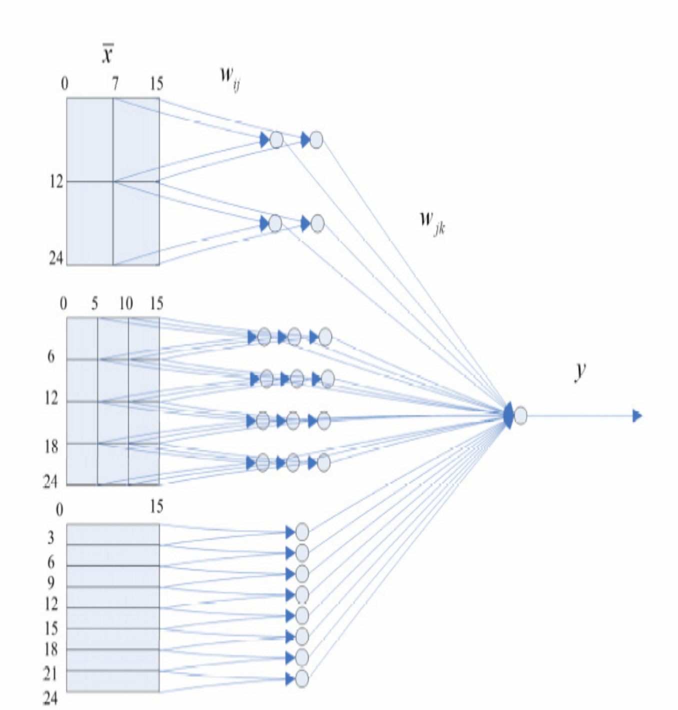

Fig. 3. Neural network architecture used to detect faces.

Software was developed to run on a backend server to compute the networks output on the compressed and masked images received from the FPGA. The face images used for the linear subspace model were obtained from participants from the University of Pretoria. Each face image was 150x240 pixels in size. These images were normalized by their mean before being processed.

The implementation of the recognition subsystem was developed using the algorithm as in [3]. The software that was developed calculated the eigenvectors and associated weights for each participant and stored the values within a database. This data was then used to test whether or not if the participants were recognized correctly, the effect of lighting on the system and the effect of system accuracy when adjusting the linear subspace dimensionality.

IV. Test Results

The neural network was tested by performing the feed forward function on all 150 face images and 150 non-face images used for training.

Figure 4 illustrates the values returned by the neural network when performed on all 150 faces. When the neural network outputs a value close to 1 it indicates that it has detected a face. Values close to 0 on this graph indicate that the network made an error and determined that a face was a non-face. This test resulted in an average value of 0.93 being returned by the network when applied to the 150 face images.

Figure 5 illustrates the values retuned by the neural network when performed on all 150 non-faces. Values close to 1 indicate an error as this means the network has detected a non-face as a face. The average value returned by the network in this test was 0.09. These graphs serve as an indication of how well the network has been trained and to what accuracy on average will the network perform.

In order to make a non-biased test of the neural networks accuracy the network was tested on 20 non-training set faces. The reason for using a far smaller number of images than the training set is because the algorithm is limited to its a priori knowledge. Since the a priori knowledge was limited to 150 face images and 150 non-face images it was decided to take approximately 10% of that to test the algorithm on non-biased data. Figure 6 illustrates the results that were achieved. The network achieved an average output value of 0.91.

According to [1] the network used in [2] was tested with 130 images containing 507 labeled faces and achieved an accuracy of 86.2%. Even though this test involved the searching of a face within background, it still serves as an indication of the performance of the algorithm in terms of percentage of correct detections. A correct detection meant that eyes and mouth were visible within the detected face window.

The average accuracy results that were obtained in this paper indicate that the neural network used resulted in approximately 90% correct detections taking place. These results are comparable to [2] because even though the network was trained on a smaller number of face and non-face images, it was tested on a smaller non-training face image set. This means that the 90% accuracy obtained for the neural network used in this paper is acceptable for the demonstration purposes of the entire system.

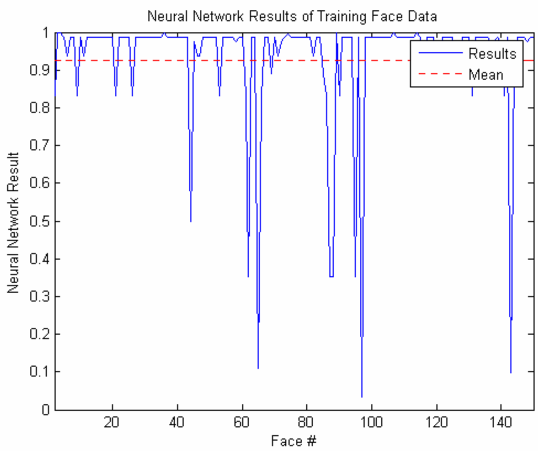

Fig. 4. Neural network results when processed on the 150 face training set.

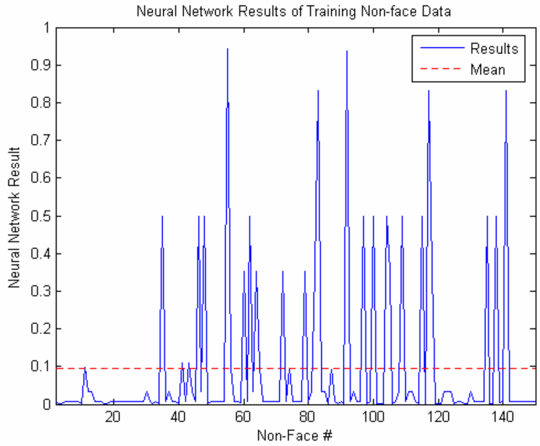

Fig. 5. Neural network results when processed on the 150 non-face training set.

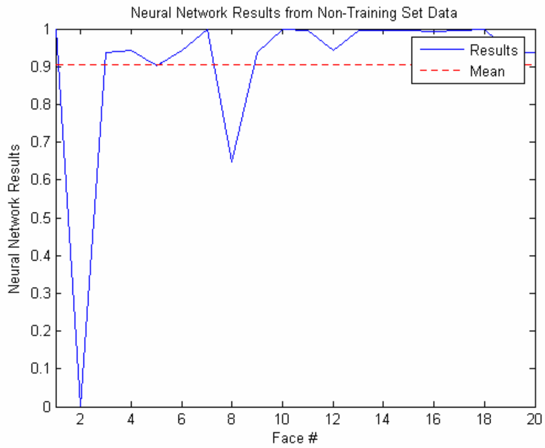

Fig. 6. Neural network results when processed on 20 non-training set faces.

When the entire system was completed a test was performed on the linear subspace model dimensionality. The luminosity at which the test was conducted was kept constant. Table 1 describes the effect of varying the percentage of eigenvalues or eigenvectors (dimensionality) from 10 to 100. The duration that it took for the system to correctly recognize an individual was noted. The neural network value that was returned was also noted. The accuracy of the system was determined primarily from the frequency of correct recognitions that occurred. The threshold value or minimum error between the face being recognized and the face within the database was recorded. These values provide valuable information as to what settings the linear subspace model should be set to and what threshold should be used.

It can be seen that at very low dimensionality (10%-50%) the system performs very poorly as the linear subspace model has very little information about the training set. The duration taken for the system to correctly determine the individual takes longer at low dimensionality as the system spends most of its time returning incorrect results.

At higher dimensionalities (60%-80%) the system performs relatively well since the linear subspace model contains enough information about the training set. The duration taken for the system to return a correct result is also improved.

At very high dimensionalities (90%-100%) the system accuracy performs very well, although the duration taken for the system to correctly identify and individual takes longer. This is because the algorithm has become too rigid and does not allow for any deviation or error between the training set and the individual to be recognized. Running the algorithm at 80% dimensionality helped to improve the system throughput while keeping it fairly accurate.

TABLE I
SYSTEM ACCURACY AND THROUGHPUT WITH VARIATION TO LINEAR SUBSPACE DIMENSIONALITY

%Eigenvalues	10	20	30	40	50
Duration	19s	2.86s	2min	5.84s	1min
Correctly Detected	Sometimes	Sometimes	Never	Sometimes	Sometimes
Linear Subspace Model Threshold	4.71E-09	3.99E-09	6.80E-03	2.00E-02	1.20E-03
Neural Network Result	0.99999	0.9999	0.9989	0.99	0.99999
Overall Accuracy	Very poor	Poor	Very poor	Poor	Very poor
%Eigenvalues	60	70	80	90	100
Duration	4.51s	3.19s	4.47s	39.35s	10.06s
Correctly Detected	Most of the time	Almost all of the time	Almost all of the time	Almost all of the time	Almost all of the time
Linear Subspace Model Threshold	3.80E-02	2.00E-02	4.70E-02	6.50E-02	3.30E-01
Neural Network Result	0.99989	0.99999	0.99999	0.9999	0.9999
Overall Accuracy	Average	Very good	Very good	Excellent	Excellent

After determining the system settings, the effect that lighting had on performance is shown in figure 7. At each independent luminosity level a picture of the person to be tested was inserted into the linear subspace database and the duration for a correct match and the accuracy was then determined.

Figure 7 illustrates the results that were obtained by ranging the luminosity from 50 lux to 300 lux in 50 lux intervals. It can be seen that at low and very high illumination the system returns a high number of incorrect matches. The time taken for the system to return the correct match is lengthened because of the frequent incorrect results. When the luminosity was set to 150 lux the system performed best by returning the correct match on the first attempt and in less than 3.8s.

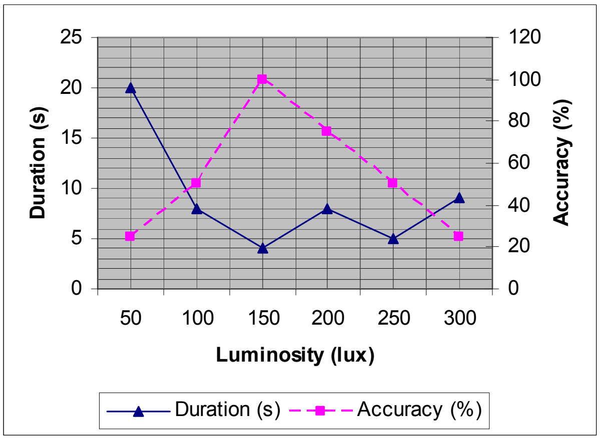

Fig. 7. Face recognition system accuracy and throughput when tested at different luminosity levels.

V. RECOMMENDATIONS

Further tests can be done to compare the performance of the neural network architecture used in this paper to the neural network described in [2] using the same image data set. It would also be useful to compare different types of detection and recognition systems with the same data set and the described system implementation.

VI. CONCLUSION

The authors have successfully implemented and characterized a combination of neural networking for facial detection and a linear subspace model for facial recognition system. The neural networking technique allowed for improved face detection on an image with increased variability in faces and complex portrait backgrounds. The linear subspace model technique was fast, relatively simple and worked well within restricted conditions. The combination of these techniques resulted in a very thorough and competent recognition system.

For this real time system employing image masking, assisted in improving the system performance and decreased the time taken to search for a face. Measuring the percentage of correct detections, dimensionality, accuracy and lighting enabled one to optimize and calibrate the system.

REFERENCES

[1] I. E. Hjelmas and B.K. Low, "Face Detection: A Survey," *Computer Vision and Image Understanding,* vol. 83, pp. 236 – 274, 2001.

[2] H.A. Rowley, S. Baluja and T. Kanada, "Neural Network-Based Face Detection," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 203 – 207, IEEE Computer Society, 1996.

[3] M.A. Turk and A.P. Pentland, "Face Recognition using Eigenfaces," *in Proc. IEEE Society Conference on Computer Vision and Pattern Recognition*, pp. 586-591, IEEE Computer Society Press, 1991.

[4] J. Ortega-Garcia, J. Bigun, D. Reynolds and J. Gonzalez-Rodriguez, "Authentication Gets Personal with Biometrics," *IEEE Signal Processing Magazine*, vol. 21, pp. 50-62, 2004.

[5] E.P. Kukula and S.J. Elliot, "Evaluation of a Facial Recognition Algorithm Across Three Illumination Conditions," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, pp. 19-23, 2004.

[6] V. Bonato, A.K. Sanches, M.M. Fernandes, J.M.P. Cardoso, E.D.V. Simoes and E. Marques, "A Real Time Gesture Recognition System for Mobile Robots", *in Proc. International Conference of Informatics in Control, Automation and Robotics*, pp. 25-28, 2004.

[7] B. Coppin, *Artificial Intelligence Illuminated* Sudbury, Jones and Barlett Publishers, 2004.