Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp208-213)

New High Speed Normalized Neural Networks For Pattern Detection

HAZEM M. EL-BAKRY, and QIANGFU ZHAO

University of Aizu, Aizu Wakamatsu, Japan 965-8580
Phone: +81-242-37-2760, Fax: +81-242-37-2743,

Abstract:- Finding an object or a face in the input image is a search problem in the spatial domain. Neural networks have shown good results for detecting a certain face/object in a given image. In this paper, faster neural networks for face/object detection are presented. Such networks are designed based on cross correlation in the frequency domain between the input image and the input weights of neural networks. This approach is developed to reduce the computation steps required by these faster neural networks for the searching process. The principle of divide and conquer strategy is applied through image decomposition. Each image is divided into small in size sub-images and then each one is tested separately by using a single faster neural network. Furthermore, fastest face/object detection is achieved by using parallel processing techniques to test the resulting sub-images at the same time using the same number of faster neural networks. In contrast to using only faster neural networks, the speed up ratio is increased with the size of the input image when using faster neural networks and image decomposition. Moreover, the problem of local subimage normalization in the frequency domain is solved. The overall speed up ratio of the detection process is increased as the normalization of weights is done off line.

Key-Word:- Fast Face/object Detection, Neural Networks, Cross Correlation, Image Normalization, Parallel Processing.

1 Introduction

Object detection is a fundamental step before object recognition. Its reliability and performance have a major influence in a whole pattern recognition system. Nowadays, neural networks have shown very good results for detecting a certain face/object in a given image [2,4,6,8,9,10,12]. Among other techniques [3,5,7], neural networks are efficient pattern detectors [2,4,6,9]. However, the problem with neural networks is that the computational complexity is very high because the networks have to process many small local windows in the images [5,7]. The main objective of this paper is to reduce the detection time using neural networks. Our idea is to fast the operation of neural networks by performing the testing process in the frequency domain instead of spatial domain. A cross correlation between the input image and the weights of neural networks is applied in the frequency domain. This model is called faster neural networks. Compared to conventional neural networks, faster neural networks show a significant reduction in the number of computation steps required to detect a certain pattern in a given image under test. Furthermore, another idea to increase the speed of faster neural networks through image decomposition is presented. There is no problem to normalize the training examples used for learning neural networks. Also, there is no problem to normalize each sub-image if the test is done in the spatial domain. However, practically the most computational signal procedures are realized in the frequency domain. Thus, the question arises, how to normalize each sub-image in the frequency domain?. The problem of sub-image (local) normalization in the Fourier space was presented in [4]. Here, a simple solution for this problem is presented. Normalizing the input image in the frequency domain is achieved by normalizing the input weights in the spatial domain. The number of computation steps required for weight normalization is proved to be less than that needed for image normalization. Mathematical calculations prove that the new idea of weight normalization, instead of image normalization, provides good results and increases the speed up ratio. Moreover, for neural networks, normalization of weights can be easily done off line before starting the search process. In section 2, faster neural networks for face/object detection are described. The details of conventional neural networks, faster neural networks, and the speed up ratio for face/object detection are given. A faster searching algorithm for face/object detection that reduces the number of the required computation steps through image decomposition is presented in section 3. Accelerating the new approach using parallel processing techniques is also introduced. Subimage normalization in the frequency domain through normalization of weights is introduced in section 4.

2 Fast Face/object Detection Using MLP and FFT

In this section, a fast algorithm for face/object detection based on two dimensional cross correlations that take place between the tested image and the sliding window (20x20 pixels) is described. Such window is represented by the neural network weights situated between the input unit and the hidden layer. The convolution theorem in mathematical analysis says that a convolution of *f* with *h* is identical to the result of the following steps: let *F* and *H* be the results of the Fourier transformation of f and h in the frequency domain. Multiply *F* and *H* in the frequency domain point by point and then transform this product into spatial domain via the inverse Fourier transform [1]. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain a speed up in an order of magnitude can be achieved during the detection process [6,8,9,10,11,12,13,14,15,16].

In the detection phase, a subimage X of size mxn (sliding window) is extracted from the tested image, which has a size PxT, and fed to the neural network. Let W_i be the vector of weights between the input subimage and the hidden layer. This vector has a size of mxz and can be represented as mxn matrix. The output of hidden neurons *h(i)* can be calculated as follows:

$$h_i = g\left(\sum_{j=1}^{m}\sum_{k=1}^{z} W_i(j,k)X(j,k) + b_i\right) \qquad (1)$$

where g is the activation function and $b(i)$ is the bias of each hidden neuron *(i).* Eq.1 represents the output of each hidden neuron for a particular subimage I. It can be computed for the whole image Ψ as follows:

$$h_i(u,v) = g\left(\sum_{j=-m/2}^{m/2}\sum_{k=-z/2}^{z/2} W_i(j,k)\Psi(u+j, v+k) + b_i\right) \qquad (2)$$

Eq. (2) represents a cross correlation operation. Given any two functions f and g, their cross correlation can be obtained by [1]:

$$f(x,y) \otimes g(x,y) = \left(\sum_{m=-\infty}^{\infty}\sum_{z=-\infty}^{\infty} f(x+m, y+n)g(m,z)\right) \qquad (3)$$

Therefore, Eq. (2) can be written as follows [10-14]:

$$h_i = g\left(\Psi \otimes W_i + b_i\right) \qquad (4)$$

where h_i is the output of the hidden neuron *(i)* and $h_i(u,v)$ is the activity of the hidden unit *(i)* when the sliding window is located at position *(u,v)* in the input image Ψ and $(u,v) \in [P-m+1, T-n+1]$.

Now, the above cross correlation can be expressed in terms of the Fourier Transform:

$$\Psi \otimes W_i = F^{-1}\left(F(\Psi) \bullet F^*\left(W_i\right)\right) \qquad (5)$$

(*) means the conjugate of the *FFT* for the weight matrix. Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u,v) = g\left(\sum_{i=1}^{q} W_o(i)\, h_i(u,v) + b_o\right) \qquad (6)$$

where q is the number of neurons in the hidden layer. $O(u,v)$ is the output of the neural network when the sliding window located at the position *(u,v)* in the input image Ψ. W_o is the weight matrix between hidden and output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1. For a tested image of *NxN* pixels, the *2D-FFT* requires a number equal to $N^2 log_2 N^2$ of complex computation steps. Also, the same number of complex computation steps is required for computing the *2D-FFT* of the weight matrix for each neuron in the hidden layer.

2. At each neuron in the hidden layer, the inverse *2D-FFT* is computed. So, q backward and $(1+q)$ forward transforms have to be computed. Therefore, for an image under test, the total number of the *2D-FFT* to compute is $(2q+1)N^2 log_2 N^2$.

3. The input image and the weights should be multiplied in the frequency domain. Therefore, a number of complex computation steps equal to qN^2 should be added.

4. The number of computation steps required by the faster neural networks is complex and must be converted into a real version. It is known that the two dimensions Fast Fourier Transform requires $(N^2/2)log_2 N^2$ complex multiplications and $N^2 log_2 N^2$ complex additions [20,21]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. So, the total number of computation steps required to obtain the *2D-FFT* of an *NxN* image is:

$$\rho = 6((N^2/2)log_2 N^2) + 2(N^2 log_2 N^2) \qquad (7)$$

which may be simplified to:

$$\rho = 5N^2 log_2 N^2 \qquad (8)$$

Performing complex dot product in the frequency domain also requires $6qN^2$ real operations.

5. In order to perform cross correlation in the frequency domain, the weight matrix must have the same size as the input image. Assume that the input object/face has a size of (nxn) dimensions. So, the search process will be done over subimages of (nxn) dimensions and the weight matrix will have the same size. Therefore, a number of zeros = (N^2-n^2) must be added to the weight matrix. This requires a total real number of computation steps = $q(N^2-n^2)$ for all neurons. Moreover, after computing the *2D-FFT* for the weight matrix, the conjugate of this matrix must be obtained. So, a real number of computation steps $=qN^2$ should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers $(e^{-jk(2\Pi n/N)})$, where $0<K<L$. These $(N/2)$ complex numbers are multiplied by the elements of the input image or by previous complex numbers during the computation of the *2D-FFT*. To create a complex number requires two real floating point operations. So, the total number of computation steps required for the faster neural networks becomes:

$$\sigma=(2q+1)(5N^2log_2N^2)+6qN^2+q(N^2-n^2)+qN^2+N \qquad (9)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N^2log_2N^2)+q(8N^2-n^2)+N \qquad (10)$$

6. Using a sliding window of size nxn for the same image of NxN pixels, $q(2n^2-1)(N-n+1)^2$ computation steps are required when using traditional neural networks for face/object detection process. The theoretical speed up factor η can be evaluated as follows:

$$\eta=\frac{q(2n^2-1)(N-n+1)^2}{(2q+1)(5N^2log_2N^2)+q(8N^2-n^2)+N} \qquad (11)$$

An interesting property with faster neural networks is that the number of computation steps does not depend on eith the size of the input subimage or the size of the weighth matrix (n). The effect of (n) on the the number of computation steps is very small and can be ignored. This is incontrast to conventional networks networks in which the number of computation steps is increased with the size of both the input subimage and the weight matrix (n).

In practical implementation, the multiplication process consumes more time than the addition one. The effect of the number of multiplications required for conventional neural networks in the speed up ratio (Eq. 11) is more than the number of of multiplication steps required by the faster neural networks. In order to clear this, the following equation (η_m) describes relation between the number of multiplication steps required by conventional and faster neural networks:

$$\eta_m=\frac{qn^2(N-n+1)^2}{(2q+1)(3N^2log_2N^2)+6qN^2} \qquad (12)$$

For general fast cross correlation the speed up ratio (η_g) is in the following form:

$$\eta_g=\frac{q(2n^2-1)N^2}{(2q+1)(5(N+\tau)^2log_2(N+\tau)^2)+q(8(N+\tau)^2-n^2)+(N+\tau)} \qquad (13)$$

where τ is a small number depends on the size of the weight matrix. General cross correlation means that the process starts from the first element in the input matrix.

3 A New Faster Algorithm for Face/Object Detection Based on Image Decomposition

In this section, a new faster algorithm for face/object detection is presented. Ss the image size is increased, the number of computation steps required by faster neural networks is much increased. For example, the number of computation steps required for an image of size (50x50 pixels) is much less than that needed for an image of size (100x100 pixels). Also, the number of computation steps required for an image of size (500x500 pixels) is much less than that needed for an image of size (1000x1000 pixels). As a result, for example, if an image of size (100x100 pixels) is decomposed into 4 sub-images of size (50x50 pixels) and each sub-image is tested separately, then a speed up factor for face/object detection can be achieved. The number of computation steps required by faster neural networks to test an image after decomposition can be calculated as follows:

1. Assume that the size of the image under test is (NxN pixels).
2. Such image is decomposed into α (LxL pixels) sub-images. So, α can be computed as:

$$\alpha=(N/L)^2 \qquad (23)$$

3. Assume that, the number of computation steps required for testing one (LxL pixels) sub-image is β. So, the total number of computation steps (T) required for testing these sub-images resulting after the decomposition process is:

$$T=\alpha\beta \qquad (24)$$

The speed up ratio in this case (η_d) can be computed as follows:

$$\eta_d=\frac{q(2n^2-1)(N-n+1)^2}{(q(\alpha+1)+\alpha)(5N_s^2log_2N_s^2)+\alpha q(8N_s^2-n^2)+N_s^2+\Delta} \qquad (25)$$

where,
Ns: is the size of each small sub-image.
Δ: is a small number of computation steps required to obtain the results at the boundaries between subimages and depends on the size of the subimage.

To detect a face/object of size 20x20 pixels in an image of any size by using faster neural networks after image decomposition into sub-images, the optimal size of these sub-images must be computed. The speed up ratio is reduced when the size of the sub-image (L) is increased. The speed up ratio is increased with the size of the input image when using faster neural networks and image decomposition. This is in contrast to using only faster neural networks. The number of computation steps required by faster neural networks is increased rapidly with the size of the input image. Therefore the speed up ratio is decreased with the size of the input image. While in case of using faster neural networks and image decomposition, the number of computation steps required by faster neural networks is increased smoothly. Thus, the linearity of the computation steps required by faster neural networks in this case is better. As a result, the speed up ratio is increased. Increasing the speed up ratio with the size of the input image is considered an important achievement. Furthermore, for very large size matrices, while the speed up ratio for faster neural networks is decreased, the speed up ratio still increase in case of using faster neural networks and matrix decomposition. Moreover, the speed up ratio in case of faster neural networks and image decomposition is increased with the size of the weight matrix which has the same size *(n)* as the input window. For example, the speed up ratio is for window size of 30x30 is larger than that of size 20x20. To detect small in size matrices such as 5x5 or 10x10 using only faster neural networks, the speed ratio becomes less than one. On the other hand, it is clear that using fast neural and image decomposition, the speed up ratio becomes higher than one and increased with the dimensions of the input image. The dimensions of the new subimage after image decomposition *(L)* must not be less than the dimensions of the face/object which is required to be detected and has the same size as the weight matrix. Therefore, the following equation controls the relation between the subimage and the size of weight matrix (face/object to be detected) in order not to loss any information in the input image.

$$L \geq n \tag{26}$$

For example, in case of detecting 5x5 pattern, the image must be decomposed into no more subimages of size 5x5.

To further reduce the running time as well as increase the speed up ratio of the detection process, a parallel processing technique is used. Each sub-image is tested using a faster neural network simulated on a single processor or a separated node in a clustered system. The number of operations (ω) performed by each processor / node (sub-images tested by one processor/node) =

$$\omega = \frac{The\ total\ number\ of\ sub-images}{Number\ of\ Processors\ /\ nodes} \tag{27}$$

$$\omega = \frac{\alpha}{Pr} \tag{28}$$

where, *Pr* is the number of processors or nodes.

The total number of computation steps (γ) required to test an image by using this approach can be calculated as:

$$\gamma = \omega\beta \tag{29}$$

By using this algorithm, the speed up ratio in this case (η_{dp}) can be computed as follows:

$$\eta_{dp} = \frac{q(2n^2-1)(N-n+1)^2}{ceil(((q(\alpha+1)+\alpha)(5N_s^2 log_2 N_s^2)+\alpha q(8N_s^2 - n^2)+N_s)/pr)} \tag{30}$$

where, *ceil(x)* is a *MATLAB* function rounds the elements of x to the nearest integers towards infinity.

A further reduction in the computation steps can be obtained by dividing each sub-image into groups. For each group, the neural operation (multiplication by weights and summation) is performed for each group by using a single processor. This operation is done for all of these groups as well as other groups in all of the sub-images at the same time. The best case is achieved when each group consists of only one element. In this case, one operation is needed for multiplication of the one element by its weight and also a small number of operations (ε) is required to obtain the over all summation for each sub-image. If the sub-image has n^2 elements, then the required number of processors will be n^2. As a result, the number of computation steps will be $\alpha q(1+\varepsilon)$, where ε is a small number depending on the value of n. For example, when $n=20$, then $\varepsilon=6$ and if $n=25$, then $\varepsilon=7$. The speed up ratio can be calculated as:

$$\eta=(2n^2-1)(N-n+1)^2/\alpha(1+\varepsilon) \tag{31}$$

Moreover, if the number of processors = αn^2, then the number of computation steps will be $q(1+\varepsilon)$, and the speed up ratio becomes:

$$\eta=(2n^2-1)(N-n+1)^2/(1+\varepsilon) \tag{32}$$

Furthermore, if the number of processors = $q\alpha n^2$, then the number of computation steps will be $(1+\varepsilon)$, and the speed up ratio can be calculated as:

$$\eta=q(2n^2-1)(N-n+1)^2/(1+\varepsilon) \tag{33}$$

In this case, as the length of each group is very small, then there is no need to apply cross correlation between the input image and the weights of the neural network in frequency domain.

4 Subimage Centering and Normalization in the Frequency Domain

In [4], the authors stated that image normalization to avoid weak or strong illumination could not be done in the frequency space. This is because the image normalization is local and not easily computed in the Fourier space of the whole image. Here, a simple method for image normalization is presented. Normalizing the image can be obtained by centering and normalizing the weights as follows [4].

Let $\bar{X}_{rc}$ be the zero-mean centered subimage located at (r,c) in the input image ψ:

$$\bar{X}_{rc} = X_{rc} - \bar{x}_{rc} \tag{34}$$

where, $\bar{x}_{rc}$ is the mean value of the sub image located at position (r,c). We are interested in computing the dot multiplication between the subimage $\bar{X}_{rc}$ and the weights W_i the of hidden layer as follows:

$$\overline{X}_{rc} \bullet W_i = X_{rc} \bullet W_i - \bar{x}_{rc} \bullet W_i \qquad (35)$$

where,

$$\bar{x}_{rc} = \frac{\sum_{k,j=1}^{n} X_{rc}(k,j)}{n^2} \qquad (36)$$

The dot multiplication denoted by (•) is not a matrix multiplication but is done element-wise (multiply each element in the first matrix by its corresponding element at the same position in the second matrix and sum up the results to obtain a one final value). Combining Eq. (24) and Eq. (25), we get the following expression:

$$\overline{X}_{rc} \bullet W_i = X_{rc} \bullet W_i - \frac{\sum_{k,j=1}^{n} X_{rc}(k,j)}{n^2} \bullet W_i \qquad (37)$$

For any two matrices with the same size, multiplying the first matrix dot by the mean of the second and summing the results the same as multiplying the second matrix dot by the mean of the first one and summing the results of multiplication. Therefore, Eq. (26) can be written as:

$$\overline{X}_{rc} \bullet W_i = X_{rc} \bullet W_i - X_{rc} \bullet \frac{\sum_{k,j=1}^{n} W_i(k,j)}{n^2} \qquad (38)$$

The zero mean weights are given by:

$$\overline{W}_i = W_i - \frac{\sum_{k,j=1}^{n} W_i(k,j)}{n^2} \qquad (39)$$

Also, Eq. (27) can be written as:

$$\overline{X}_{rc} \bullet W_i = X_{rc} \bullet \left(W_i - \frac{\sum_{k,j=1}^{n} W_i(k,j)}{n^2} \right) \qquad (40)$$

So, we may conclude that:

$$\overline{X}_{rc} \bullet W_i = X_{rc} \bullet \overline{W}_i \qquad (41)$$

which means that multiplying a normalized image with a non-normalized weight matrix dot multiplication is equal to the dot multiplication of the non – normalized image with the non-normalized weight matrix.

5 Conclusion

Normalized neural networks for faster face/object detection in a given image have been presented. It has been proved mathematically and practically that the speed of the detection process becomes faster than conventional neural networks. This has been accomplished by applying cross correlation in the frequency domain between the input image and the normalized input weights of the neural networks. A new general formulas for fast cross correlation as well as the speed up ratio have been given. A faster neural network approach foe face/object detection has been introduced. Such approach has decomposed the input image under test into many small in size sub-images. Furthermore, a simple algorithm for faster face/object detection based on cross correlations in the frequency domain between the sub-images and the weights of the neural net has been presented in order to speed up the execution time. Simulation results have shown that, using a parallel processing technique, large values of speed up ratio could be achieved. Moreover, by using faster neural networks and image decomposition, the speed up ratio has been increased with the size of the input image. Also, the problem of local subimage normalization in the frequency space has been solved. It has been generally proved that the speed up ratio in the case of image normalization through normalization of weights is faster than subimage normalization in the spatial domain. This speed up ratio is faster than the one obtained without normalization. The proposed approach can be applied to detect the presence/absence of any other object in an image.

References

[1] R. Klette, and Zamperon, *Handbook of image processing operators*, John Wiley & Sonsltd, 1996.

[2] H. A. Rowley, S. Baluja, and T. Kanade, “ Neural Network -Based Face Detection,“ *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 1, pp. 23-38, 1998.

[3] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition, " *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 45-51, SantaBarbara, CA, 1998.

[4] R. Feraud, O. Bernier, J. E. Viallet, and M. Collobert, "A Fast and Accurate Face Detector for Indexation of Face Images," *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, 28-30 March, 2000.

[5] Y. Zhu, S. Schwartz, and M. Orchard, "Fast Face Detection Using Subspace Discriminate Wavelet Features," *Proc. of IEEE Computer Society International Conference on Computer Vision and Pattern Recognition (CVPR'00)*, South Carolina, June 13 - 15, 2000, vol.1, pp. 1636-1643.

[6] H. M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," *Machine Graphics & Vision Journal (MG&V)*, vol. 10, no. 1, 2001, pp. 47-73.

[7] S. Srisuk and W. Kurutach, "A New Robust Face Detection in Color Images," Proc. of IEEE Computer Society International Conference on Automatic Face and Gesture Recognition, Washington D.C., USA, May 20-21, 2002, pp. 306-311.

[8] H. M. El-Bakry, "Face detection using fast neural networks and image decomposition," *Neurocomputing Journal*, vol. 48, 2002, pp. 1039-1046.

[9] H. M. El-Bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Networks and Image Decomposition," *Machine Graphics & Vision Journal (MG&V)*, vol. 11, no. 4, 2002, pp. 498-512.

[10] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Object/Face Detection Using Neural Networks and Fast Fourier Transform,”

International Journal of Signal Processing, vol.1, no.3, pp. 182-187, 2004.

[11] Hazem M. El-Bakry, and Qiangfu Zhao, “A Modified Cross Correlation in the Frequency Domain for Fast Pattern Detection Using Neural Networks,” *International Journal of Signal Processing*, vol.1, no.3, pp. 188-194, 2004.

[12] Hazem M. El-Bakry, and Qiangfu Zhao, “Face Detection Using Fast Neural Processors and Image Decomposition,” *International Journal of Computational Intelligence*, vol.1, no.4, pp. 313-316, 2004.

[13] Hazem M. El-Bakry, and Qiangfu Zhao, “Fast Complex Valued Time Delay Neural Networks,” *International Journal of Computational Intelligence*, vol.2, no.1, pp. 16-26, 2005.

[14] Hazem M. El-Bakry, and Qiangfu Zhao, “A Fast Neural Algorithm for Serial Code Detection in a Stream of Sequential Data,” *International Journal of Information Technology*, vol.2, no.1, pp. 71-90, 2005.

[15] H. M. El-Bakry, and Q. Zhao, "A New Symmetric Form for Fast Sub-Matrix (Object/Face) Detection Using Neural Networks and FFT," *International Journal of Signal Processing*, to be published.

[16] H. M. El-Bakry, and Q. Zhao, "Fast Pattern Detection Using Normalized Neural Networks and Cross Correlation in the Frequency Domain," *EURASIP Journal on Applied Signal Processing,* to be published.

[17] S. Ben-Yacoub, B. Fasel, and J. Luettin, “Fast Face Detection using MLP and FFT,” *Proc. of the Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA'99)*, 1999.

[18] B. Fasel, “Fast Multi-Scale Face Detection,” *Research report IDIAP-Com 98-04, at IDIAP research institute*, 1998, http://www.idiap.ch/

[19] S. Ben-Yacoub, “Fast Object Detection using MLP and FFT,” *Research report IDIAP-RR 11, at IDIAP research institute*, 1997, http://www.idiap.ch/

[20] James W. Cooley and John W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput*. 19, 297–301 (1965).

[21] J.P. Lewis, “Fast Normalized Cross Correlation”, Available from:<http://www.idiom.com/~zilla/ Papers/nvisionInterface/nip.html >

[22] H. M. El-Bakry, "Face detection using fast neural networks and image decomposition," *Neurocomputing Journal*, vol. 48, 2002, pp. 1039-1046.

[23] H. M. El-Bakry, "Comments on Using MLP and FFT for Fast Object/Face Detection," *Proc. of IEEE IJCNN’03*, Portland, Oregon, pp. 1284-1288, July, 20-24, 2003.

[24] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Object/Face Detection," *Proc. of the 30th Anniversary SOFSEM Conference on Current Trends in Theory and Practice of Computer Science*, 24-30 January, 2004, Hotel VZ MERIN, Czech Republic.

[25] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Sub-Matrix (Object/Face) Detection," *Proc. of IEEE International Symposium on Circuits and Systems*, Vancouver, Canada, 23-26 May, 2004.

[26] H. M. El-Bakry, "Fast Sub-Image Detection Using Neural Networks and Cross Correlation in Frequency Domain," *Proc. of IS 2004: 14th Annual Canadian Conference on Intelligent Systems*, Ottawa, Ontario, 6-8 June, 2004.

[27] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Code Detection in a Stream of Sequential Data," *Proc. of CIC 2004 International Conference on Communications in Computing*, Las Vegas, Nevada, USA, 21-24 June, 2004.

[28] H. M. El-Bakry, "Fast Neural Networks for Object/Face Detection," *Proc. of 5th International Symposium on Soft Computing for Industry with Applications of Financial Engineering*, June 28 - July 4, 2004, Sevilla, Andalucia, Spain.

[29] H. M. El-Bakry, and H. Stoyan, "A Fast Searching Algorithm for Sub-Image (Object/Face) Detection Using Neural Networks," *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, 18-21 July, 2004, Orlando, USA.

[30] H. M. El-Bakry, and H. Stoyan, "Fast Neural Networks for Code Detection in Sequential Data Using Neural Networks for Communication Applications," *Proc. of the First International Conference on Cybernetics and Information Technologies*, Systems and Applications: CITSA 2004, 21-25 July, 2004. Orlando, Florida, USA, Vol. IV, pp. 150-153.

[31] H. M. El-bakry, M. A. Abo-elsoud, and M. S. Kamel, "Fast Modular Neural Networks for Human Face Detection," *Proc. of IEEE-INNS-ENNS International Joint Conference on Neural Networks*, Como, Italy, Vol. III, pp. 320-324, 24-27 July, 2000.

[32] H. M. El-bakry, "Fast Iris Detection using Cooperative Modular Neural Nets," *Proc. of the 6th International Conference on Soft Computing*, 1-4 Oct., 2000, Japan.

[33] H. M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," *Machine Graphics & Vision Journal (MG&V)*, vol. 10, no. 1, 2001, pp. 47-73.

[34] H. M. El-bakry, "Fast Iris Detection Using Cooperative Modular Neural Networks," *Proc. of the 5th International Conference on Artificial Neural Nets and Genetic Algorithms*, pp. 201-204, 22-25 April, 2001, Sydney, Czech Republic.

[35] H. M. El-bakry, "Fast Iris Detection Using Neural Nets," *Proc. of the 14th Canadian Conference on Electrical and Computer Engineering*, pp.1409-1415, 13-16 May, 2001, Canada.

[36] H. M. El-bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Nets," *Proc. of INNS-IEEE International Joint Conference on Neural Networks*, pp. 577-582, 14-19 July, 2001, Washington, DC, USA.

[37] H. M. El-bakry, "Human Iris Detection for Information Security Using Fast Neural Nets," *Proc. of the 5th World Multi-Conference on Systemics, Cybernetics and Informatics*, 22-25 July, 2001, Orlando, Florida, USA.

[38] H. M. El-bakry, "Human Iris Detection for Personal Identification Using Fast Modular Neural Nets," *Proc. of the 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences,* pp. 112-118, 25-28 July, 2001, Monte Carlo Resort, Las Vegas, Nevada, USA.

[39] H. M. El-bakry, "Human Face Detection Using Fast Neural Networks and Image Decomposition," *Proc. the fifth International Conference on Knowledge-Based Intelligent Information & Engineering Systems*, 6-8 September 2001, Osaka-kyoiku University, Kashiwara City, Japan, pp. 1330-1334.

[40] H. M. El-Bakry, "Fast Iris Detection for Personal Verification Using Modular Neural Networks," *Proc. of the International Conference on Computational Intelligence*, 1-3 Oct., 2001, Dortmund, Germany, pp.-269-283.

[41] H. M. El-bakry, "Fast Cooperative Modular Neural Nets for Human Face Detection," *Proc. of IEEE International Conference on Image Processing,* 7-10 Oct., 2001, Thessaloniki, Greece.

[42] H. M. El-Bakry, "Fast Face Detection Using Neural Networks and Image Decomposition," *Proc. of the 6th International Computer Science Conference*, Active Media Technology, Dec. 18-20, 2001, Hong Kong – China, pp. 205-215, 2001.

[43] H. M. El-Bakry, "Face Detection Using Fast Neural Networks and Image Decomposition," *Proc. of INNS-IEEE International Joint Conference on Neural Networks*, 14-19 May, 2002, Honolulu, Hawaii, USA.