

A FAST NEURAL ALGORITHM FOR PATTEN DETECTION USING CROSS CORRELATION IN THE FREQUENCY DOMAIN

Hazem M. El-Bakry

*University of Aizu, Aizu Wakamatsu, Japan 965-8580
hazemelbakry1@yahoo.com*

ABSTRACT—Recently, fast neural networks for object/face detection were presented in [1-3]. The speed up factor of these networks relies on performing cross correlation in the frequency domain between the input image and the weights of the hidden layer. But, these equations given in [1-3] for conventional and fast neural networks are not valid for many reasons presented here. In this paper, correct equations for cross correlation in the spatial and frequency domains are presented. Furthermore, correct formulas for the number of computation steps required by conventional and fast neural networks given in [1-3] are introduced. A new formula for the speed up ratio is established. Also, corrections for the equations of fast multi scale object/face detection are given. Moreover, commutative cross correlation is achieved. Simulation results show that sub-image detection based on cross correlation in the frequency domain is faster than classical neural networks.

KEYWORDS—Conventional Neural Networks, Fast Neural Networks, Cross Correlation in the Frequency Domain.

I. INTRODUCTION

Pattern detection is a fundamental step before pattern 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 pattern in a given image [7,20]. But 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 [4,19]. The authors in [1-3] have proposed a multilayer perceptron (MLP) algorithm for fast object/face detection. The same authors claimed incorrect equation for cross correlation between the input image and the weights of the neural networks. They introduced formulas for the number of computation steps needed by conventional and fast neural networks. Then, they established an equation for the speed up ratio. Unfortunately, these formulas contain many errors which lead to invalid speed up ratio. Other authors developed their work based on these incorrect equations [5-18],[20-30]. So, the fact that these equations are not valid must be cleared to all

researchers. It is not only very important but also urgent to notify other researchers not to waste their time and effort doing research based on wrong equations. The main objective of this paper is to correct the formulas of cross correlation as well as the equations which describe the computation steps required by conventional and fast neural networks presented in [1-3]. Some of these wrong equations were corrected in our previous publications [5-18], [20-30]. Here, all of these errors are corrected. In section II, fast neural networks for object/face detection are described. Comments on conventional neural networks, fast neural networks, and the speed up ratio of object/face detection are presented in section III.

II. FAST OBJECT/FACE DETECTION USING MLP AND FFT

In [1-3], a fast algorithm for object/face detection based on two dimensional cross correlations that take place between the tested image and the sliding window (20x20 pixels) was 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. 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 [1-3].

In the detection phase, a sub image I of size $m \times n$ (sliding window) is extracted from the tested image, which has a size $P \times T$, and fed to the neural network. Let W_i be the vector of weights between the input sub image and the hidden layer. This vector has a size of $m \times n$ and can be represented as $m \times n$ matrix. The output of hidden neurons $h(i)$ can be calculated as follows:

$$h_i = g \left(\sum_{j=1}^m \sum_{k=1}^n W_i(j,k) I(j,k) + b_i \right) \quad (1)$$

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 sub-image I . It can be computed for the whole image Ψ as follows:

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

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

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

Therefore, Eq.2 may be written as follows [1,2]:

$$h_i = g(W_i \otimes \Psi + b_i) \quad (4)$$

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:

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

(*) 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:

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

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 .

The authors in [1-3] analyzed their proposed fast neural network as follows: For a tested image of $N \times N$ pixels, the 2D-FFT requires $O(N^2(\log_2 N)^2)$ computation steps. For the weight matrix W_i , the 2D-FFT can be computed off line since these are constant parameters of the network independent of the tested image. The 2D-FFT of the tested image must be computed. As a result, q backward and one forward transforms have to be computed. Therefore, for a tested image, the total number of the 2D-FFT to compute is $(q+1)N^2(\log_2 N)^2$. In addition, the input image and the weights should be multiplied in the frequency domain. Therefore, computation steps of (qN^2) should be added. This yields

a total of $O((q+1)N^2(\log_2 N)^2 + qN^2)$ computation steps for the fast neural network.

Using sliding window of size $n \times n$, for the same image of $N \times N$ pixels, qN^2n^2 computation steps are required when using traditional neural networks for the face detection process. The theoretical speed up factor η can be evaluated as follows [1]:

$$\eta = \frac{qn^2}{(q+1)\log^2 N} \quad (7)$$

III. COMMENTS ON FAST NEURAL NET PRESENTED FOR OBJECT/ FACE DETECTION

The speed up factor introduced in [1] and given by Eq.7 is not correct for the following reasons:

1- The number of computation steps required for the 2D-FFT is $O(N^2 \log_2 N^2)$ and not $O(N^2 \log^2 N)$ as presented in [1,2]. Also, this is not a typing error as the curve in Fig.2 in [1] realizes Eq.7, and the curves in Fig.15 in [2] realizes Eq.31 and Eq.32 in [2].

2- Also, the speed up ratio presented in [1] not only contains an error but also is not precise. This is because for fast neural networks, the term $(6qN^2)$ corresponds to complex dot product in the frequency domain must be added. Such term has a great effect on the speed up ratio. Adding only qN^2 as stated in [2] is not correct since a one complex multiplication requires six real computation steps.

3- For conventional neural networks, the number of operations is $(q(2n^2-1)(N-n+1)^2)$ and not (qN^2n^2) . The term n^2 is required for multiplication of n^2 elements (in the input window) by n^2 weights which results in another new n^2 elements. Adding these n^2 elements, requires another (n^2-1) steps. So, the total computation steps needed for each window is $(2n^2-1)$. The search operation for a face in the input image uses a window with $n \times n$ weights. This operation is done at each pixel in the input image. Therefore, such process is repeated $(N-n+1)^2$ times and not N^2 as stated in [1,3].

4- Before applying cross correlation, the 2D-FFT of the weight matrix must be computed. Because of the dot product, which is done in the frequency domain, the size of weight matrix should be increased to be the same as the size of the input image. Computing the 2D-FFT of the weight matrix off line as stated in [1-3] is not practical. In this case, all of the input images must have the same size. As a result, the input image will have only a one fixed size. This means that, the testing time for an image of size 50×50 pixels will be the same as that image of size 1000×1000 pixels and of course, this is unreliable. So, another number of complex computation steps to perform 2D-FFT for $(N \times N)$ matrix should be added to the complex number of computation

steps (σ) required by the fast neural networks as follows:

$$\sigma = ((2q+1)(N^2 \log_2 N^2) + 6qN^2) \quad (8)$$

This will increase the computation steps required for the fast neural networks especially when q is more than one neuron.

5- It is not valid to compare number of complex computation steps by another of real computation steps directly. The number of computation steps given by pervious authors [1-3] for conventional neural networks is for real operations while that is required by the fast neural networks is for complex operations. To obtain the speed up ratio, the authors in [1-3] have divided the two formulas directly without converting the number of computation steps required by the fast neural networks 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. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 2D-FFT of an $N \times N$ image is:

$$\rho = 6((N^2/2)\log_2 N^2) + 2(N^2 \log_2 N^2) \quad (9)$$

which may be simplified to:

$$\rho = 5(N^2 \log_2 N^2) \quad (10)$$

6- For the weight matrix to have the same size as the input image, 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 2D-FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required by the fast neural networks is:

$$\sigma = ((2q+1)(5N^2 \log_2 N^2) + 6qN^2 + q(N^2 - n^2) + qN^2 + N) \quad (11)$$

which can be reformulated as:

$$\sigma = ((2q+1)(5N^2 \log_2 N^2) + q(8N^2 - n^2) + N) \quad (12)$$

Therefore, the correct speed up ratio is as follows:

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

The correct theoretical speed up ratio (Eq.13) with different sizes of the input image and different in size

weight matrices is listed in Table 1. Practical speed up ratio for manipulating images of different sizes and different in size weight matrices is listed in Table 2 using 700 MHz processor and Matlab ver 5.3.

For general fast cross correlation the speed up ratio becomes in the following form:

$$\eta = \frac{q(2n^2 - 1)(N^2 - n^2 + 1)}{(2q + 1)(5(N + \tau)^2 \log_2 (N + \tau)^2) + q(8(N + \tau)^2 - n^2) + (N + \tau)} \quad (14)$$

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. The theoretical speed up ratio for general fast cross correlation (Eq.14) is shown in Table 3. Compared with MATLAB cross correlation function (`xcorr2`), experimental results show that the our proposed algorithm is faster than this function as shown in Table 4.

7- Furthermore, there are critical errors in Eq.3 and Eq.4 (which is Eq.4 in [1] and also Eq.13 in [2]). Eq.3 is not correct because the definition of cross correlation is:

$$f(x,y) \otimes g(x,y) = \left(\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(x+m, y+n) g(m,n) \right) \quad (15)$$

and then Eq.4 must be written as follows:

$$h_i = g(\Psi \otimes W_i + b_i) \quad (16)$$

Therefore, the cross correlation in the frequency domain given by Eq.5 does not represent Eq.4. This is because the fact that the operation of cross correlation is not commutative ($W \otimes \Psi \neq \Psi \otimes W$). As a result, Eq.4 does not give the same correct results as conventional neural networks. This error leads the researchers in [22-30] who consider the references [1-3] to think about how to modify the operation of cross correlation so that Eq.4 can give the same correct results as conventional neural networks. Therefore, errors in these equations must be cleared to all the researchers. In [21-30], the authors proved that a symmetry condition must be found in input matrices (images and the weights of neural networks) so that fast neural networks can give the same results as conventional neural networks. In case of symmetry $W \otimes \Psi = \Psi \otimes W$, the cross correlation becomes commutative and this is a valuable achievement. In this case, the cross correlation is performed without any constrains on the arrangement of matrices. As presented in [22-30], this symmetry condition is useful for reducing the number of patterns that neural networks will learn. This is because the image is converted into symmetric shape by rotating it down and then the up image and its rotated down version are tested together as one (symmetric) image. If a pattern is detected in the rotated down image, then, this means that this pattern is found at the relative position in the up image. So, if

conventional neural networks are trained for up and rotated down examples of the pattern, fast neural networks will be trained only to up examples. As the number of trained examples is reduced, the number of neurons in the hidden layer will be reduced and the neural network will be faster in the test phase compared with conventional neural networks.

8- Moreover, the authors in [1-3] stated that the activity of each neuron in the hidden layer (Eq.4) can be expressed in terms of convolution between a bank of filter (weights) and the input image. This is not correct because the activity of the hidden neuron is a cross correlation between the input image and the weight matrix. It is known that the result of cross correlation between any two functions is different from their convolution. As we proved in [21-30] the two results will be the same, only when the two matrices are symmetric or at least the weight matrix is symmetric.

9- Images are tested for the presence of a face (object) at different scales by building a pyramid of the input image which generates a set of images at different resolutions. The face detector is then applied at each resolution and this process takes much more time as the number of processing steps will be increased. In [1-3], the authors stated that the Fourier transforms of the new scales do not need to be computed. This is due to a property of the Fourier transform. If $z(x,y)$ is the original and $a(x,y)$ is the sub-sampled by a factor of 2 in each direction image then:

$$a(x, y) = z(2x, 2y) \quad (17)$$

$$Z(u, v) = FT(z(x, y)) \quad (18)$$

$$FT(a(x, y)) = A(u, v) = \frac{1}{4} Z\left(\frac{u}{2}, \frac{v}{2}\right) \quad (19)$$

This implies that we do not need to recompute the Fourier transform of the sub-sampled images, as it can be directly obtained from the original Fourier transform. But experimental results have shown that Eq.17 is valid only for images in the following form:

$$\Psi = \begin{bmatrix} A & A & B & B & C & C & \dots & \dots & \dots \\ A & A & B & B & C & C & \dots & \dots & \dots \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ \cdot & & & & & & & & \\ S & S & X & X & Y & Y & \dots & \dots & \dots \\ S & S & X & X & Y & Y & \dots & \dots & \dots \end{bmatrix} \quad (20)$$

In [1], the author claimed that the processing needs $O((q+2)N^2 \log^2 N)$ additional number of computation steps. Thus the speed up ratio will be [1]:

$$\eta = \frac{qn^2}{(q+2)\log^2 N} \quad (21)$$

Of course this is not correct, because the inverse of the Fourier transform is required to be computed at each neuron in the hidden layer (for the resulted matrix from the dot product between the Fourier matrix in two dimensions of the input image and the Fourier matrix in two dimensions of the weights, the inverse of the Fourier transform must be computed). So, the term $(q+2)$ in Eq.21 should be $(2q+1)$ because the inverse 2D-FFT in two dimensions must be done at each neuron in the hidden layer. In this case, the number of computation steps required to perform 2D-FFT for the fast neural networks will be:

$$\varphi = (2q+1)(5N^2 \log_2 N^2) + (2q)5(N/2)^2 \log_2 (N/2)^2 \quad (22)$$

In addition, a number of computation steps equal to $6q(N/2)^2 + q((N/2)^2 - n^2) + q(N/2)^2$ must be added to the number of computation steps required by the fast neural networks.

IV. CONCLUSION

It has been shown that the equations given in [1-3] for conventional and fast neural networks contain errors. The reasons for these errors have been proved. Correct equations for cross correlation in the spatial and frequency domains have been presented. Furthermore, correct equations for the number of computation steps required by conventional, and fast neural networks have been introduced. A new correct formula for the speed up ratio has been established. Also, correct equations for fast multi scale object/face detection have been given. Moreover, commutative cross correlation has been achieved by converting the non-symmetric input matrices into symmetric forms. Theoretical and practical results after these corrections have shown that generally fast neural networks requires fewer computation steps than conventional one. Finally, the proposed model has no shortcomings. This is because the results of conventional and fast neural networks are the same. Therefore, the performance (detection rate) is the same.

REFERENCES

- [1] S. Ben-Yacoub, "Fast Object Detection using MLP and FFT," IDIAP-RR 11, IDIAP, 1997.
- [2] B. Fasel, "Fast Multi-Scale Face Detection, " IDIAP-Com 98-04, 1998.
- [3] S. Ben-Yacoub, B. Fasel, and J. Luetin , "Fast Face Detection using MLP and FFT, " in Proc. Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA'99), 1999.
- [4] 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.

- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] Hazem 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.
- [17] H. M. El-Bakry, "Face detection using fast neural networks and image decomposition," Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.
- [18] Hazem 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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, Andalusia, Spain.
- [27] 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.
- [28] 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.
- [29] H. M. El-Bakry, and Q. Zhao, "A New Symmetric Form for Fast Sub-Matrix (Object/Face) Detection Using Neural Networks and FFT," accepted and under publication in the International Journal of Signal Processing.
- [30] H. M. El-Bakry, and Q. Zhao, "Fast Pattern Detection Using Normalized Neural Networks and Cross Correlation in the Frequency Domain," accepted and under publication in the EURASIP Journal on Applied Signal Processing.

Table 1: The theoretical speed up ratio for images with different sizes.

Image size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	3.67	5.04	6.34
200x200	4.01	5.92	8.05
300x300	4.00	6.03	8.37
400x400	3.95	6.01	8.42
500x500	3.89	5.95	8.39
600x600	3.83	5.88	8.33
700x700	3.78	5.82	8.26
800x800	3.73	5.76	8.19
900x900	3.69	5.70	8.12
1000x1000	3.65	5.65	8.05

Table 2: Practical Speed up ratio for images with different sizes Using MATLAB ver 5.3.

Image size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	7.88	10.75	14.69
200x200	6.21	9.19	13.17
300x300	5.54	8.43	12.21
400x400	4.78	7.45	11.41
500x500	4.68	7.13	10.79
600x600	4.46	6.97	10.28
700x700	4.34	6.83	9.81
800x800	4.27	6.68	9.60
900x900	4.31	6.79	9.72
1000x1000	4.19	6.59	9.46

Table 3: The theoretical speed up ratio for the general fast cross correlation algorithm.

Image size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	5.39	8.36	11.95
200x200	4.81	7.49	10.75
300x300	4.51	7.03	10.16
400x400	4.32	6.73	9.68
500x500	4.18	6.52	9.37
600x600	4.07	6.35	9.13
700x700	3.99	6.21	8.94
800x800	3.91	6.10	8.77
900x900	3.84	6.00	8.63
1000x1000	3.78	5.91	8.51

Table 4: Simulation results of the speed up ratio for the general fast cross correlation compared with the MATLAB cross correlation function (xcorr2).

Image size	Speed up ratio (n=20)	Speed up ratio (n=25)	Speed up ratio (n=30)
100x100	10.14	13.05	16.49
200x200	9.17	11.92	14.33
300x300	8.25	10.83	13.41
400x400	7.91	9.62	12.65
500x500	6.77	9.24	11.77
600x600	6.46	8.89	11.19
700x700	5.99	8.47	10.96
800x800	5.48	8.74	10.32
900x900	5.31	8.43	10.66
1000x1000	5.91	8.66	10.51