FAST PATTERN DETECTION USING PARALLEL NEURAL PROCESSORS AND IMAGE DECOMPOSITION

HAZEM M. EL-BAKRY

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

ABSTRACT- In this paper, an approach to reduce the computation steps required by fast neural networks for the searching process is presented. 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 using a fast neural network. The operation of fast neural networks based on applying cross correlation in the frequency domain between the input image and the weights of the hidden neurons. Compared to conventional and fast neural networks, experimental results show that a speed up ratio is achieved when applying this technique to locate human faces automatically in cluttered scenes. Furthermore, faster face detection is obtained by using parallel processing techniques to test the resulting sub-images at the same time using the same number of fast neural networks. In contrast to using only fast neural networks, the speed up ratio is increased with the size of the input image when using fast neural networks and image decomposition.

KEYWORDS: Fast Neural Networks, 2D-FFT, Cross Correlation, Image decomposition, Parallel Processing.

I. INTRODUCTION

The human face is a complex pattern. Finding human faces automatically in a scene is a difficult yet significant problem. It is the first step in fully automating human face recognition system. Face detection is the fundamental step before the face recognition or identification procedure. Its reliability and time response have a major influence on the performance and usability of the whole face recognition system. For web indexation applications, the processing time must be kept as low as possible as the number of images on the web increases continuously [8]. Among other techniques [6], neural networks are efficient face detectors [2,5].

The main objective of this paper is to reduce the detection time using neural networks. Compared to conventional neural networks, fast neural networks based on cross correlation between the input image and the weights of neural networks in frequency domain have shown a significant reduction in the number of computation steps required to detect an object (face/iris) in the image under test [1,2]. In section II, fast neural networks for face detection are described. A faster searching algorithm for face detection that reduces the number of the required computation steps through image decomposition is presented in section III. Accelerating the new approach using parallel processing techniques is introduced in section IV.

II. FAST NEURAL NETWORKS FOR HUMAN FACE DETECTION

In this section, 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 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 sub image and the hidden layer. This vector has a size of mxn and can be represented as mxn matrix. The output of hidden neurons h(i) can be calculated as follows:

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

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 sub-image I. It can be obtained for the whole image Z as follows:

Proceedings 19th European Conference on Modelling and Simulation
Yuri Merkuryev, Richard Zobel, Eugène Kerckhoffs © ECMS, 2005
ISBN 1-84233-112-4 (Set) / ISBN 1-84233-113-2 (CD)

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

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

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

Therefore, Eq.2 may be written as follows:

$$h_i = g\left(Z \otimes X_i + b_i\right) \quad (4)$$

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

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

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

Hence, by evaluating this cross correlation between the input image and the weights of the hidden layer, a speed up ratio can be obtained compared to conventional neural networks. Also, the final output of the neural network can be evaluated as follows:

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

O(u,v) is the output of the neural network when the sliding window located at the position (u,v) in the input image Z, and w_o is the vector of weights between the hidden and the output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows [1]:

1- For a tested image of NxN pixels, the 2D-FFT requires a number equal to $N^2\log_2N^2$ 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 2DFFT to compute is $(2q+1)N^2\log_2N^2$.

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 fast neural networks is complex and must be converted into a real version. It is known that the two dimensions Fast Fourier Transform requires $(N^2/2)\log_2N^2$ complex multiplications and $N^2\log_2N^2$ 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. So, the total number of computation steps required to obtain the 2D-FFT of an NxN image is [1]:

$$\rho = 6((N^2/2)\log_2N^2) + 2(N^2\log_2N^2) \quad (7)$$

which may be simplified to:

$$\rho = 5(N^2\log_2N^2) \quad (8)$$

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. So, 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 FFT2 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 FFT2. To create a complex number requires two real floating point operations. So, the total number of computation steps required for fast neural networks becomes [1]:

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

which can be reformulated as:

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

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 object/face detection process. The theoretical speed up factor η can be evaluated as follows [3]:

$$\eta = \frac{q(2n^2 - 1)(N^2 - n^2 + 1)}{(2q+1)(5N^2 \log_2 N^2) + q(8N^2 - n^2) + N} \quad (11)$$

III. A NEW FASTER ALGORITHM FOR HUMAN FACE DETECTION BASED ON IMAGE DECOMPOSITION

In this section, a new faster algorithm for face detection is presented. The number of computation steps required for fast neural networks with different image sizes is listed in Table 1. From this table, we may notice that as the image size is increased, the number of computation steps required by fast 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 detection can be achieved. The number of computation steps required by fast neural networks to test an image after decomposition can be calculated as follows [7]:

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:

$$\alpha=(N/L)^2 \quad (12)$$

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:

$$T = \alpha \beta \quad (13)$$

To detect a face of size 20x20 pixels in an image of any size by using fast neural networks after image decomposition into sub-images, the optimal size of these sub-images must be computed. From Table 1, we may conclude that, the most suitable size for the sub-image which requires the smallest number of computation steps is 25x25 pixels. A comparison between the speed up ratio for fast neural networks and fast neural networks after image decomposition with different sizes of the tested images is listed in Table 2 (n=20, q=30). The speed up ratio is increased with the size of the input image when using fast neural networks and image decomposition. This is in contrast to using only fast neural networks.

IV. SIMULATION OF THE FAST FACE DETECTION PROCESS (AFTER IMAGE DECOMPOSITION) USING PARALLEL PROCESSING TECHNIQUES

In the previous section, a new algorithm for face detection based on decomposing the image under test to many sub-images has been presented. Then, for each sub-image, a fast neural network has been used to detect the presence/absence of human faces. Here, 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 fast 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) =

$$\omega = \frac{\text{The total number of sub} - \text{images}}{\text{Number of Processors / nodes}} \quad (14)$$

$$\omega = \frac{\alpha}{\Pr} \quad (15)$$

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:

$$\gamma=\omega\beta \quad (16)$$

As shown in Table 3, using a symmetric multiprocessing system with 16 parallel processors or 16 nodes in either a massively parallel processing system or a clustered system, the speed up ratio (with respect to conventional neural networks) for human face detection is increased. 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 ε=6 and if n=25, then ε=7. The speed up ratio can be calculated as:

$$\eta=O((2n^2-1)(N-n+1)^2/\alpha(1+\varepsilon)) \quad (17)$$

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:

$$\eta = O((2n^2-1)(N-n+1)^2/(1+\varepsilon)) \quad (18)$$

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:

$$\eta = O(q(2n^2-1)(N-n+1)^2/(1+\varepsilon)) \quad (19)$$

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.

V. CONCLUSIONS

A faster neural network approach has been introduced to identify frontal views of human faces. Such approach has decomposed the image under test into many small in size sub-images. A simple algorithm for fast face 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. Furthermore, simulation results have shown that, using a parallel processing technique, large values of speed up ratio could be achieved. Moreover, by using fast neural networks and image decomposition, the speed up ratio has been increased with the size of the input image. The proposed approach can be applied to detect the presence/absence of any other object in an image.

REFERENCES

[1] 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.

[2] H. M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks " The International Journal on Machine Graphics, Vol. 10, No. 1, 2001, pp. 47-73.

[3] 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.

[4] R. Klette, and Zamperon, "Handbook of image processing operators, " John Wiley & Sons, Ltd, 1996.

[5] 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.

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

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

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

Table 1: The number of computation steps required by fast neural networks (FNN) for images of sizes (25x25 - 1050x1050 pixels).

Image size	No. Of computation steps in case of using FNN	Image size	No. of computation steps in case of using FNN
25x25	1.9085e+006	550x550	1.7524e+009
50x50	9.1949e+006	600x600	2.1130e+009
100x100	4.2916e+007	650x650	2.5096e+009
150x150	1.0460e+008	700x700	2.9426e+009
200x200	1.9610e+008	750x750	3.4121e+009
250x250	3.1868e+008	800x800	3.9186e+009
300x300	4.7335e+008	850x850	4.4622e+009
350x350	6.6091e+008	900x900	5.0434e+009
400x400	8.8203e+008	950x950	5.6623e+009
450x450	1.1373e+009	1000x1000	6.3191e+009
500x500	1.4273e+009	1050x1050	7.0142e+009

Table 2: The speed up ratio in case of using FNN and FNN after image decomposition into sub-images (25x25 pixels) for images of different sizes.

Image size	Speed up ratio in case of using (FNN)	Speed up ratio in case of using FNN after image decomposition
50x50	2.505 2	4.5871
100x100	3.6646	8.9997
150x150	3.9325	10.7600
200x200	4.0045	11.6707
250x250	4.0136	12.2228
300x300	3.9985	12.5923
350x350	3.9736	12.8565
400x400	3.9449	13.0547
450x450	3.9151	13.2088
500x500	3.8855	13.3320

Table 3: The speed up ratio in case of using FNN after image decomposition into sub-images (25x25 pixels) for images of different sizes using 16 parallel processors or 16 nodes.

Image size	Speed up ratio
50x50	73.3935
100x100	143.9953
150x150	172.1592
200x200	186.7312
250x250	195.5652
300x300	201.4760
350x350	205.7032
400x400	208.8745
450x450	211.3405
500x500	213.3126