International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:11, 2007

Design of a Neural Networks Classifier for Face Detection

F. Smach, M. Atri, J. Mitéran and M. Abid

Abstract— Face detection and recognition has many applications in a variety of fields such as security system, videoconferencing and identification. Face classification is currently implemented in software. A hardware implementation allows real-time processing, but has higher cost and time to-market.

The objective of this work is to implement a classifier based on neural networks MLP (Multi-layer Perceptron) for face detection. The MLP is used to classify face and non-face patterns. The systm is described using C language on a P4 (2.4 Ghz) to extract weight values. Then a Hardware implementation is achieved using VHDL based Methodology. We target Xilinx FPGA as the implementation support.

***Keywords*—**Classification, Face Detection, FPGA Hardware description, MLP.

I. Introduction

HUMAN face detection and recognition is an active area of research spanning several disciplines such as image processing, pattern recognition and computer vision. Face detection and recognition are preliminary steps to a wide range of applications such as personal identity verification, video-surveillance, liptrocking, facial expression extraction, gender classification, advanced human and computer interaction.

Most methods are based on neural network approaches, feature extraction, Markov chain, skin color, and others are based on template matching [1].

Pattern localization and classification is the step which is used to classify face and non-face patterns. Many systems dealing with object classification are based on ANN (Artificial Neural Networks). In this paper we are intersted by the design of a ANN algorithm in order to achieve image classification.

This paper is organized as follows: In section II, we give an overview over classification for face detection. Description of our model is discussed in Section III. Section IV deals with the training method. Section V presents the hardware implementation of our architecture. Section 6 provides the results. Finally, we give some concluding remarks in Section VII.

Manuscript received Mars 4, 2005.

F. Smach is with the GMS Group of ENIS University, Sfax 3000, Tunisia, (corresponding author phone: +216 98 688 512; fax: +216 73 500 278 ; e-mail: smach_fethi@ yahoo.fr).

M. Atri, is with the EμE Laboratory of Science Monastir University, Monastir, 5000 Tunisia (e-mail: Mohamed.atri@fsm.rnu.tn).

J. Mitéran, is with the LE2I Laboratory of Université de Bourgogne Aile des Sciences de l'Ingénieur BP 47870 21078 Dijon Cedex, France, (e-mail: miteranj@u-bourgogne.fr).

M. Abid is with the GMS Group of ENIS University, Sfax 3000, Tunisia, (e-mail: Mohamed.abid@enis.rnu.tn).

II. Classification For Face Detection

While numerous methods have been proposed to detect face in a single image of intensity or color images. A related and important problem is how to evaluate the performance of the proposed detection methods [1]. Many recent face detection papers compare the performance of several methods, usually in terms of detection and false alarm rates. It is also worth noticing that many metrics have been adopted to evaluate algorithms, such as learning time, execution time, the number of samples required in training, and the ratio between detection rates and false alarms. In general, detectors can make two types of errors: *false negatives* in which faces are missed resulting in low detection rates and *false positives* in which an image is declared to be face.

$$False\ negative = \frac{Number.of.Missed Falses}{Total Number of.Actual Faces}$$

$$False\ positive = \frac{Number.of.Incorrectly.Detected Faces}{Total Number of.Actual Faces}$$

Face detection can be viewed as two-class recognition problem in which an image region is classified as being a "Face" or "nonFace". Consequently, face detection is one of the few attempts to recognize from images a class of objects for which there is a great deal of within-class variability. Face detection also provide interesting challenges to the underlying pattern classification and learning techniques. The class of face and no face image are decidedly characterized by multimodal distribution function and effective decision boundaries are likely to be nonlinear in the image space.

Pattern localization and classification are CPU time intensive being normally implemented in software, however with lower performance than custom implementations. Custom implementation in hardware allows real-time processing, having higher cost and time-to-market than software implementation. Some works [2,3,4] uses ANN for classification, and the system is implemented in software, resulting in a poor performance (10 sec for localization and classification). A similar work is presented in [5], aiming to object localization and classification, and it was also

implemented in software (10-15 frames/sec). An ANN MLP was implemented on DSPs, standard microprocessor and FPGA dedicated to image processing [6]. The proposed architecture is pipelined and results are given for a 256x256 image.

We are intersted by the implementation of a ANN algorithm in order to provide image classification. The MLP *(Multi-layer Perceptron)* algorithm is used to classify face and non-face patterns before the recognition step.

III. Multi-Layers Perceptron

The MLP neural network [1] has feedforword architecture within input layer, a hidden layer, and an output layer. The input layer of this network has N units for an N dimensional input vector. The input units are fully connected to the ***I*** hidden layer units, which are in turn, connected to the ***J*** output layers units, where ***J*** is the number of output classes.

A Multi-Layers Perceptron (MLP) is a particular of artificial neural network [7]. We will assume that we have access to a training dataset of *l* pairs (x_i, yi) where x_i is a vector containing the pattern, while y_i is the class of the corresponding pattern. In our case a 2-class task, y_i can be coded 1 and -1.

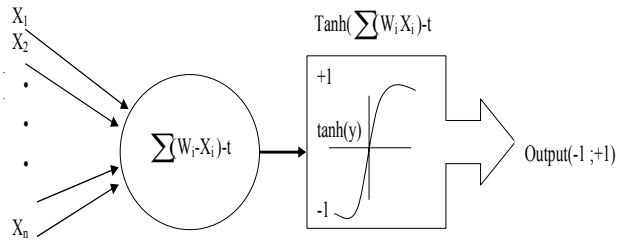

Fig.1 The neuron of supervised trainning [8].

We considered a MLP (Multi-Layers Perceptron) with a 3 layers, the input layer is a vector constituted by n^2 units of neurons (*n* x *n* pixel input images). The hidden layer has n neurons, and the output layer is a single neuron which is active to 1 if the face is presented and to otherwise.

The activity of a particular neuron ***j*** in the hidden layer is writing by: $S_j = \sum_{i \in input} w_{ji} x_i$, $x_i = f(s_j)$ (1), f a sigmoid function.

Where W_{1i} is the set of weights of neuron ***i,*** $b_1(i)$ is the threshold and x_i is an input of the neuron.

Similarly the output layer activity is: $S_j = \sum_{i \in input} w_{ji} x_i$

In our system, the dimension of the retina is 15x15 pixels represent human faces and non face, the input vector is constituted by 225 neurons, the hidden layer has 15 neurons.

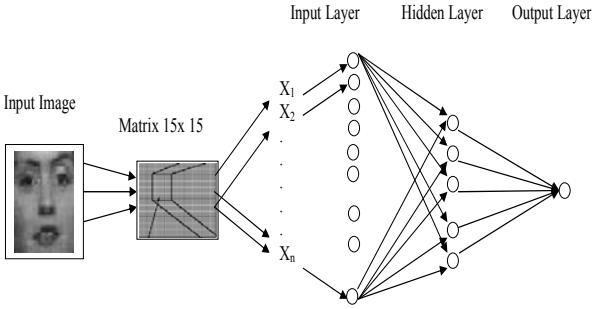

Fig.2 Architecture of proposed system

The examples were taken from the FERET database. The MLP was trained on 500 face and 200 non-face examples.

IV. Training Methodology

The MLP with the training algorithm of back propagation is universal mappers, which can in theory, approximate any continuous decision region arbitrarily well. Yet the convergence of back propagation algorithms is still an open problem. It is well known that the time cost of back propagation training often exhibits a remarkable variability. It has been demonstrated that, in most cases, rapid restart method can prominently suppress the heavy-tailed nature of training instances and improve efficiency of computation.

Multi-Layer Perceptron (MLP) with a back propagation learning algorithms was chosen for the proposed system because of its simplicity and its capability in supervised pattern matching. It has been successfully applied to many pattern classification problems [9]. Our problem has been considered to be suitable with the supervised rule since the pairs of input-output are available.

For training the network, we used the classical backpropagation algorithm. An example is picked from the training set, the output is computed. The error is calculated as the difference between the actual and the desired output. It is minimized by back-propagating it and by adjusting the weights.

Although back-propagation can be applied to network with any number of layers, it has been shown that one layer of hidden units suffices to approximate any function [9]. Therefore, in most application, a MLP NNs with a single layer of hidden units is used with a sigmoid activation function for the units $f(a) = \frac{1}{1+e^{-a}}$ (2), this function has the interesting property of having an easy to compute derivative $f'(a) = f(a)[1 - f(a)]$ (3)

The MLP training is amount to: Repeatedly presented with sample inputs and desired targets, then the output and targets are compared and the error measured. At last, adjusts weights until correct output for every input.

V. Hardware Implementation

Our aim is to implement an efficient model of unconstrained face tracking and real time face detection in arbitrary images. Artificial Neural Networks (ANNs) have been proved to be an effective way to solve this problem [10], but due to long time process in software implementation. However, with today's design technology, we are given the chance to perform face detection at higher level, which involves the real time domain. We are able to shift the detection stage in hardware implementation, to achieve several advantages [8]. The papers [11, 12] discuss other methods implemented in a hardware implementation.

The MLP implemented is a three-layer perceptron (Fig.2), one hidden layer and one output layer are used in this network system. The block diagram presented in Fig.3 corresponds to the hardware implementation of our system. It is constituted of four units: input register bank, control unit, neuron and output register.

Computation of any activation neuron coefficient may be executed by employing a multiply and accumulate method where partial product are computed separately and subsequently added. Each neuron takes a vector input of N patterns. Each vector component is multiplied by a fixed weigt value, which is determined at trainning time by software implementation. Weigts are updated during the training process, but remain constant during the detection process. The result of the MAC operation is passed an input to a function activation and returns the value of the hyperbolic tangent (tanh) between 1 and -1.

This function implementation in hardware is very difficult in its known expression. In order to simplify function expression, it was linearized on several intervalls $[C_i, C_{i+1}]$ and its value is evaluated using two constants (a_i and b_i) corresponding to this intervall.

$F(x)= a_i x+ b_i$ for $x \in [C_i, C_{i+1}]$

$F(x)= 1$ for $x > 3$

The W_{ij} values, calculated in the training step, were stored in a ROM in each neuron. Every W_{ij} coded with 8 bits represents a fixed-point weight. The operation of multiplication provide every time a result with 16 bits fraction.

A multiplexing bloc (Mux+counter) was used to provide one neuron output at each clock cycle to the next stage. The output layer neuron achieve the same tasks as the other neurons before giving the output result.

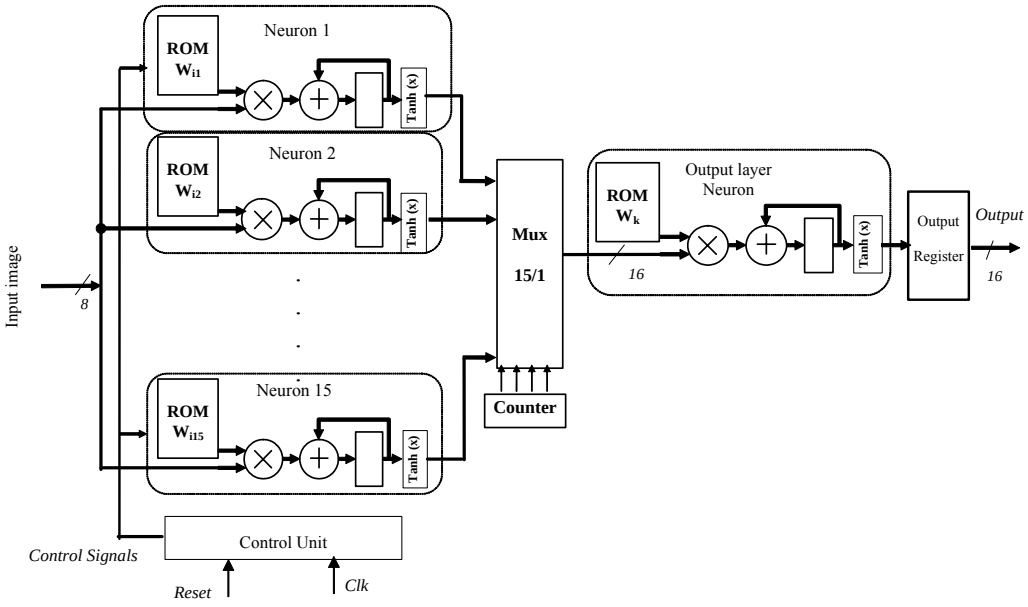

Fig.3 Block diagram

VI. Results

This architecture was implemented using Matlab in a graphical environment allowing face detection in a database. It has been evaluated using the test data of 500 images containing faces, on this test set we obtained a good detection, if the input image presented face the answer of the output neuron is rate of 0.9. These results encouraged us to implement this architecture targeting a hardware device using a HDL based methodology.

In general, the process of designing a system will proceed from a behavioral to a physical representation, gaining implementation details along the way. High level synthesis converts a behavioral specification of a digital system into an equivalent RTL design that meets a set of stated performance constraint [13]. The designer describes his system with a high level specification at one of abstraction levels. This description with a HDL (Hardware Description Language) is synthesized using existent synthesis tool allowing passage to the next abstraction level until reaching either integration in ASIC or implementation in FPGA.

The system was implemented in VHDL, and synthesized

using Leonardo synthesis tool. Target technology was FPGA Xilinx operating at 52 Mhz. The used device was a virtex v1000bg560. It contains 12248 slices and was occupied at 99.67%.

VII. Conclusion

Our experiments have shown that using MLP neural networks for face detection is a very promising approach. The model's robustness has been obtained with a back propagation learning algorithms and the *tanh* activation function. In our approach no pre-processing is needed since the normalization is incorporated directly in the weights of the input network.

Face classification are normally implemented in hardware allowing real-time processing. Classification is a step which must be complemented with feature extraction in order to demonstrate detection accuracy and performances.

References

[1] Ming-Husan Yang, David J.Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey ", IEEE transaction on pattern analysis and machine intelligence, vol.24 no.1, January 2002.

[2] H. A. Rowley, S. Baluja, T. Kanade, "Neural Network-Based Face Detection", IEEE Trans. On Pattern Analysis and Machine Intelligence, vol.20, No. 1, Page(s). 39-51, 1998.

[3] Zhang ZhenQiu, Zhu Long, S.Z. Li, Zhang Hong Jiang, "Real-time multi-view face detection" Proceeding of the Fifth IEEE International Conference on automatic Face and Gesture Recognition, Page(s): 142-147, 20-21 May 2002.

[4] R.Feraund, O.J. Bernier, J. Viallet, M Collobert,"A fast and accurate face detector based on neural network", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 23 Issue: 1,Pages(s):42-53, Jan.2001.

[5] Gavrila, D.M; Philomin, V."Real-Time Object detection for Smart Vehicules". International Conference on Computer Vision (ICCV99). Vol. 1. Corfu, Greece, 20-25 September, 1999.

[6] Rolf F. Molz, Paulo M. Engel, Fernando G. Moraes, Lionel Torres, Michel robert,"System Prototyping dedicated to Neural Network Real-Time Image Processing", ACM/SIGDA ninth international Symposium On Field Programmable Gate Arrays(FPGA 2001).

[7] Haisheng Wu, John Zelek," A Multi-classifier Based Real-time Face Detection System", Journal of IEEE Transaction on Robotics and Automation, 2003.

[8] Theocharis Theocharides, Gregory Link, Vijaykrishnan Narayanan, Mary Jane Irwin, "Embedded Hardware Face Detection", 17th Int'l Conf.on VLSI Design, Mumbai, India. January 5-9, 2004.

[9] Fan Yang and Michel Paindavoine,"Prefiltering for pattern Recognition Using Wavelet Transform and Neural Networks", Adavances in imaging and Electron physics, Vol. 127, 2003.

[10] Xiaoguang Li and shawki Areibi,"A Hardware/Software co-design approach for Face Recognition", The 16th International Conference on Microelectronics, Tunisia 2004.

[11] Fan Yang and Michel Paindavoine,"Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification", IEEE Transactions on Neural Networks, vol.14, No.5, September 2003.

[12] R. McCready, "Real-Time Face Detection on a Configurable Hardware System", International Symposium on Field Programmable Gate Arrays, 2000, Montery, California, United States.

[13] D. Gajski, N. Dutt, A. Wu, "High-Level Synthesis: Introduction to Chip and System Design", *Kluwer Academic Publishers*, Boston, 1992.