A Fast and Simple Face Detection Algorithm Using Neural Network and Its Implementation on FPGA

Mohammad Hossein DoostMohammadi[†] and Arman Garousi[†]

[†]Department of Electrical Engineering, Hamedan University of Technology, Hamedan, Iran

Summary

Face recognition is one of the interesting types of biometric which determines the presence or absence of human faces in the picture. In this paper, a face recognition system is presented that benefits from an optimized architecture based on the MLP neural network. The proposed method considerably improves the speed and the accuracy of detection compared to traditional architectures of neural network. To reduce the overall computation, neural network is organized so that to be able to rule out the majority of the non-image areas located in the image's background before applying the main algorithm. An important advantage of this new architecture is its homogeneous structure that makes it suitable for optimized implementation on a hardware platform. In this work, FPGA is used as the platform for implementation of the proposed algorithms. The implementation was done considering Taylor expansion of the activation functions. The performance of the proposed method and the implemented system was evaluated on the BioID dataset. Accomplishment of the proposed method is high precision while reducing training time and total calculations, together with appropriate robustness. Finally, a comparison with other face recognition methods has been done to show the performance of the presented system. The comparison result shows that the proposed system outperforms the other mentioned methods.

Keywords:

Biometric; Face detection; Neural network; FPGA platform

1. Introduction

Using biometrics is increasing nowadays. Biometrics topic contains the human biological features such as fingerprint, Iris, face etc. to be used in some applications. Identification and authentication are important fields in which biometrics are used. Among biometrics Face detection is one of the active filed of research in machine vision and identification. One reason could be that the face detection system has a wide range of applications including access controlling of security systems, video surveillance and video and image database management.

Face detection issue is determination of the fact that whether the human face is exist in the digital image or not. In the other words, the task of face detection systems is to find the location of all faces in the picture (if any). Face detection is the first stage of the algorithm presented for applications such as face tracking, face identification and facial recognition. The complexity of detection between faces and non-face classes increases due to any change to the images. These changes can be summarized under the following categories [1]:

(i) Gesture: the face image is variable due to the relative face gesture (facing, 45-degree, profile, etc.) So certain facial features such as eyes and nose may have remained relatively or completely hidden from view.

(ii) Facial expressions: the faces of the people are directly affected by facial expressions.

(iii) Lighting: two identical figures with the same conditions and viewing angles, look different under different lighting.

(iv) Beard and mustache: presence or absence of certain facial features such as beard, mustache and glasses is not specified. In addition, there are wide variations in these characteristics, including shape, color and size.

(v) Covering: some part of the face may be covered by other objects. Also, if the image contains several figures, some figures may cover the other faces.

According to this description, it is difficult to design a complete system to be able to solve the problem of face recognition, considering all the variables. Many researches have been done to improve face detection accuracy. Nanni and Lumini proposed a face detection method based on multiresolution local ternary patterns and local phase quantization descriptors which improved the total performance of the detection system [2]. Dornaika et al. used tracked facial actions to recognize facial expression with the help of face detection [3]. Mousavi et al. presented a new efficient method for face and eye detection [4]. Chakrabarty et al. used Volterra kernel method together with artificial bee colony to enhance the performance of the face recognition system [5].

Mamta and Hanmandlu presented a new entropy function and a new classifier to enhance the performance of the thermal face recognition [6]. Raghavendra et al.

Manuscript received April 5, 2025
Manuscript revised April 20, 2025
https://doi.org/10.22937/IJCSNS.2025.25.4.13

introduced a light field camera (LFC) for face presentation attack detection (PAD) together with improving the performance [7]. Miliki et al. used Mutual information technique together with Principle Component Analysis (PCA) to reduce complexity cost and to achieve better detection rate [8]. Jeong et al. proposed a semi-local structure patterns (SLSP) for robust face detection to overcome the problem of noise vulnerability of the conventional methods [9]. Zhang et al. proposed a new method for face detection using deep cascade multitask framework to improve the performance of detection system [10].

Neural networks have shown very good result in detection of a specific pattern in the image. But the problem is that neural networks are very complex in calculations due to the requirement of processing multiple small areas of the images. For example, we can point to the neural networks that are designed to detect skin color [11] or pattern recognition [12,13]. Identifying areas of skin color in color images can be used as a pre-processing stage which significantly improves the diagnosis speed of neural networks.

In some algorithms, the authors tried to increase the speed of neural network recognition process. Feraud et al. presented a face detection system based on Constrained Generative Model (CGM) of neural network model. They also proposed some methods ate each stage to increase the efficiency and speed of the system [14]. Ishak et al. used conventional neural network together with some filtering method to improve the speed of recognition [15]. Zuo et al. proposed a semi real-time face detector using a cascade of neural network (NN) ensembles for enhanced detection accuracy and efficiency [16]. In this algorithm, a correlation coefficient between the input image and neural network parameters is used in the frequency domain.

Nowadays, one of the efficient hardware platform for implementing image and signal processing algorithms is FPGA platform. FPGA-based implementations of various systems have been done in different literature. Romero-Troncoso et al. have implemented a monitoring systems for the squirrel-cage induction motor to accurately detect different faults in an incipient state [17]. Wojcikowski et al. the FPGA-based hardware implementation of an algorithm for an automatic traffic surveillance sensor network [18]. Fons et al. implemented a Fingerprints-based authentication system on FPGA chip [19].

Ashorian et al. implemented a license plate recognition algorithm on FPGA considering morphological approaches and improved them by using adaptive techniques to achieve more compatibility with practical applications [20]. Mandal and Mishra proposed a FPGA-based pipelined CORDIC architecture for digital demodulation in high performance, low power frequency modulated CW Radar [21]. Dwivedi et al. have used FPGA Virtex- 7 chip to implement a new designed FIR filter based on multi-objective artificial bee colony [22]. Yuan investigated the possibility and effectiveness of multi-mode vibration control of a plate through real-time FPGA implementation [23]. Zivarian et al. implemented a designed object distance measurement system on FPGA chip which led to an accurate real-time measuring system [24].

The main goal of this paper is to design and implement a low complex and cheap face detection system with high accuracy to locate the face when we have facing gesture. The presented method is based on cascade classifier. The advantage of cascade classifier is the use of a tree structure for the classification. In a big picture, many areas would be removed at the first step to remain the important ones. Therefore, the first step should be to popularize high enough to decrease the number of positive error for the next step of processing by MLP neural network. The purpose of cascade classifier is to reach the minimum value of negative error and therefore to increase the accuracy of diagnosis.

The rest of this paper is organized as follow: in Section 2, more details about cascade classifier and MLP neural network is presented. In Section 3, implementation process of the proposed algorithm on FPGA chip is discussed. The simulation results of implementing the proposed algorithm are also presented in this section. Finally in Section 4, a brief conclusion of this study is presented.

2. Methodology and Materials

The design of the proposed algorithm is based on cascade classification technique and MLP neural network with the aim of achieving better recognition accuracy. Other reasons of using cascade classifier could be its high speed, high precision and its simplicity of implementation. Using the proposed system, the face recognition process would be done in three stages:

(1) Preprocessing stage: in this stage, some basic image processing techniques such as lighting normalization, histogram normalization etc. are applied to the input image to increase the speed and accuracy of detection algorithm.
(2) Cascade classification stage: in this stage, the output image from former stage is considered as the input image, and non-face areas of this image are going to be removed using cascade classification technique.

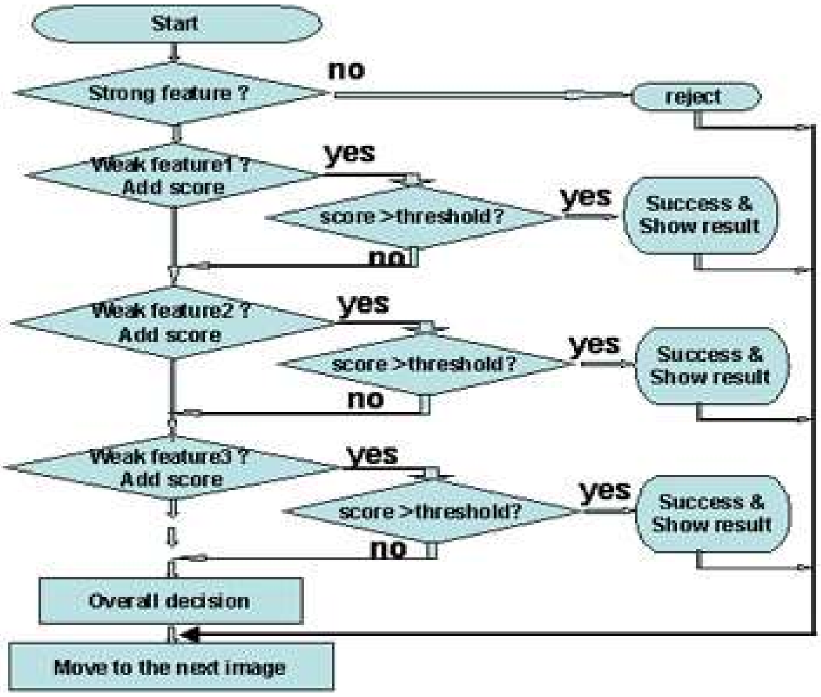


Fig.1. Flowchart of the face diagnosis process of the proposed algorithm

(3) MLP neural network stage: in this stage, the remained areas from former stage (candidate as possible face) are examined for the last time to detect the exact location of the face.

The face diagnosis process of the proposed algorithm is shown in Fig. 1. To avoid figure bustle, only three levels of cascade classification has been shown.

2.1. Combination of Cascade Classifier and Neural Network

As it was said, the proposed algorithm is based on cascade classifier and neural network. Before delivering the image to the neural network, some steps should be passed to achieve better performance. These steps are as follow:

(i) Image enhancement: The input images to the diagnosis system may have various format, size or resolution. Some initial techniques are needed to ensure the qualification of the input image. For example, histogram normalization can led to the better contrast of the input image and as consequences, balancing light variances in different images.

(ii) Preprocessing: In this step, the size of the input image is set to the standard one. Also, if the image is colorful, it is going to be mapped to a gray scale one using gray scale techniques.

(iii) Extracting face’s features: The resulting image from former step is turned to some areas with possibility of face existence in them. This job is done using cascade classifier technique. Then, obtained areas are fed to the desired neural network.

Some of these features used in the proposed algorithm are shown in Figs. 2-4.

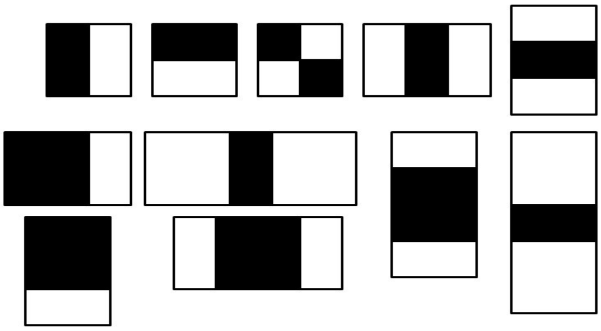


Fig.2. Features used in the proposed algorithm. Second and third rows show the new complementary features.

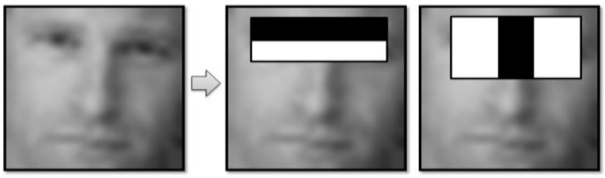


Fig.3. The first two features chosen by cascade classifier technique

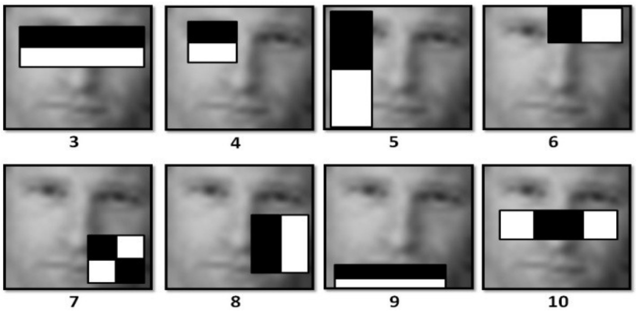


Fig.4. The third to tenth features chosen by cascade classifier technique

Finally, extracted areas from last step are delivered to the neural network. Multilayer feed forward neural network contains neurons with sigmoid activation function. This neural network can be used in two modes. The first one is called "training mode". The purpose of the training on artificial neural networks is to adjust the weight of synapses to achieve the desired processing on a training set of face and non-face samples. More precisely, at the first mode, neural network is fed by a number of training couples that makes the network parameters to be set using supervised learning algorithm. Second mode is called "classification mode". In this mode, areas extracted from the cascade classifier are delivered to the input layer of the neural network and get released during the network to calculate the activation value of each output neurons.

2.2. Cascade Classification Technique

After processing the input image, there are many sub-windows. About 99% of these sub-windows are non-face areas. Due to this fact, the total processing time is essentially the consumption time of rejecting the non-face sub-windows. Therefore, an appropriate system should be able to reject the non-face sub-window as quickly as possible and deals with face sub-windows _ the main goal of processing _as soon as possible. In the case of cascade structure, if any sub-window is recognized as non-face area in each level of cascade, it will be rejected immediately.

The set of images used for training and validation of system is one of the most important factors in designing a powerful face detection system. As the distribution of faces in the collection is larger, system flexibility increases. In this paper, dataset from BioID is used [25]. About 1520 images were selected from this database. Then, they were resized to the dimensions of 45 x 60. The mirror image was produced to increase flexibility and as a result, the number of images increased to 3040. Also, about 440 image that contains no faces were used to produce non-face images. For extracting non-faces parts, images collected were used with different window sizes starting from 36x24 to the size of original picture. 70% of the face image and 70% of non-face images were used to train the presented system (seven percent for each level of the cascade structure). The remaining 30% was used for system validation.

In a cascaded system, the performance of each process has a direct impact on overall system performance. The positive detection rate of the presented system (in percent) and the percentage of false positive detection for a cascade classifier can be calculated using (1) and (2), respectively:

$$D = \prod_{i=1}^{K} di \tag{1}$$

$$F = \prod_{i=1}^{K} fi \tag{2}$$

where d_i is diagnosis percentage of i_{th} stage of the cascade system, and f_i is the percentage of false positive detection of the same stage. The number of stages used here for the cascade system is considered as K.

2.3. MLP Neural Network

This network received windows (36x24) of the original image as input and returns output values between +1 and -1. As the output is closer to +1, the window is considered more likely as a face, and the closer to -1, the more likely non-face. To ensure that all faces in the image are specified, the windows from all segments of the overall picture are going to be checked. Also, to recognize the faces bigger than the desired window, the original image resized to a smaller one, and all of the windows will be rechecked. This can be repeated several times. The number of repetition is determined according to the type of application and the type of images that are given to the system. The general trend here is as follow: first, some windows of the image are separated and pre-processing is done on them. Then, the enhanced image is given to the neural network and the network response will be analyzed. Then, an arbitrary threshold is applied on the network response to determine whether this window represents a face or not.

Since this algorithm performs processing only on the intensity of image pixels, so there is no need to colorful images. But in the case of colorful images, the areas with skin color could be isolated from other areas by applying an appropriate filter and then, searching for the face can be done in this isolated area. At pre-processing stage, when the unification of image pixels are done, the shadows on the face will be estimated and subtracted from the image. To estimate the shade, a linear spectrum is applied to the intensity of image pixels. The effect of the linear filter is shown in Fig. 5. By doing so, the impact of the shade has been reduced as much as possible, and the light becomes uniform all over the face.

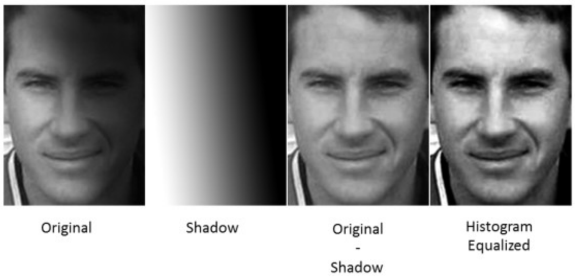


Fig.5. The result of applying a linear filter to the image and as a result, reducing the impact of the shade

The original frame of this window is in the form of a rectangular, however, this frame is different in various databases. For example in some of databases, the frame's boundary is from the eyebrow to the lips (e.g. DataSet MIT Center for Biological and Computation) and in some other databases, from the forehead to the chin (e.g. DataSet Yale). But in the second type of images, the corners of this frame usually contain background and some parts of the clothes. Thus, the frame contains irrelevant information which may result in a significant error in the process of face detection. Therefore, an oval mask is applied on the windows to remove background.

To do this, the minimum pixel value is being searched among the whole pixels of the original image and the mask. The structure of the oval mask is described as follow: there is an oval area around a center where the intensity of the pixels are equal to 255, and the intensity of surrounding areas are equal to 0. As a result, the intensity of all pixels is a value between 0 and 255, in which, the zero value indicates the black color. Thus, after searching for minimum value, the areas around the frame are became black and will have no effect on the network's decision-making. The effect of the oval mask is shown in Fig. 6.

After pre-processing and enhancement stage, the output image is given to the neural network. The number of neurons in the input layer is the same as the number of window's pixels. Each neuron receives its corresponding pixel's intensity and gets to the next layer up. Finally, the network has an output that specifies whether the input image is a face or not.

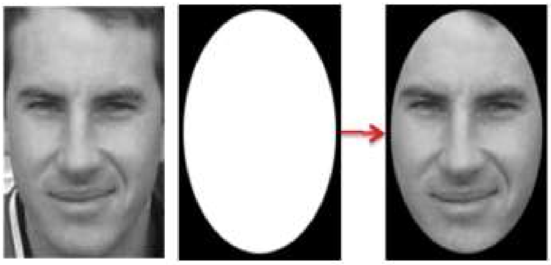


Fig.6. The result of applying an oval mask to the face image

3. Implementation on FPGA

3.1. Why FPGA Platform

There are different platforms which have been used for implementation of neural network. Some of the most important platforms together with their advantages and disadvantages in terms of efficiency and flexibility are shown in Fig. 7.

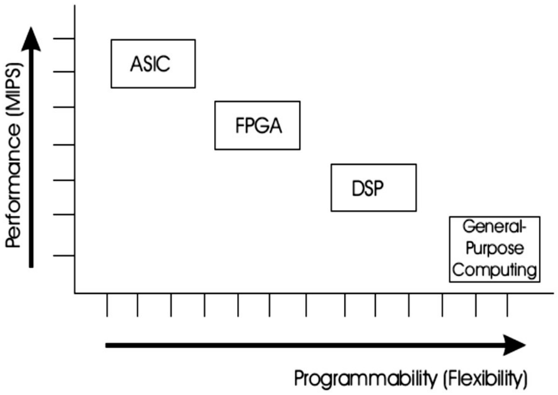


Fig.7. Different platforms for implementation of neural networks

As is clear from Fig. 7, FPGAs have relatively high efficiency with acceptable flexibility which makes it an appropriate choice for implementation of neural networks. Although the ASICs provide better performance compared to the FPGAs, but they are not practically used in such cases due to high non-mass production cost and being non-reconfigurable (without extra cost). In addition to the FPGA's appropriate speed for real-time applications, they can be reconfigured so many times as required with no extra cost.

3.2. Implementing on FPGA Chip

For ease of implementation on FPGA, network training can be done using an appropriate software such as MATLAB. Then, after determining the network parameters, they can be implemented on FPGA chip. This will reduce the complexity of required hardware and, consequently, increase the speed of face detection system. The first step is implementation of neuron's mathematical model which is shown in Fig. 8.

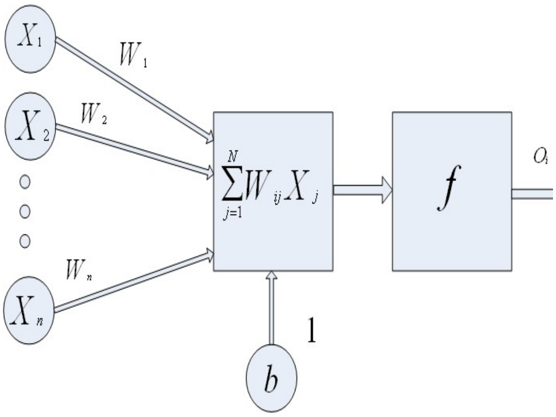


Fig. 8. The mathematical model of neuron

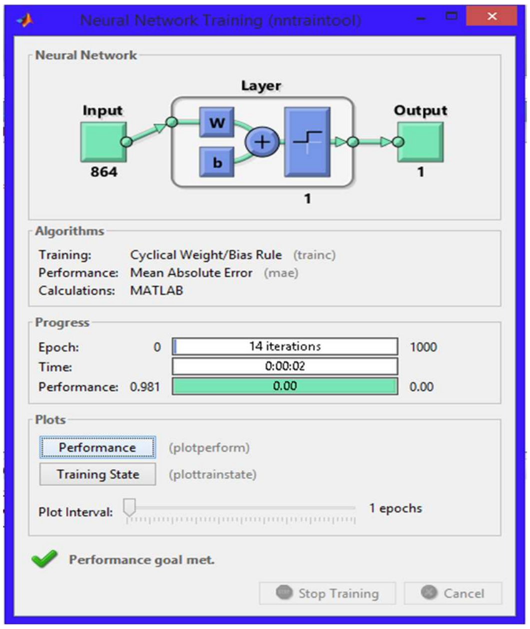
MLP neural network consists of several layers in which, each layer includes several simple neurons as shown in Fig. 8. Thus, according to the mathematical model of neuron, for implementation of the neuron on the FPGA, it is necessary to have a memory to store the weights vector and biases, and even a hardware to implement the activation function. The implementation can be efficiently done by designing a RAM memory in the FPGA and approximating the activation function using Taylor expansion around zero.

To prevent complexity of the floating-point calculations, a unit precision can be determined for all calculation, so that the floating-point numbers are multiplied by a fixed number to be out of floating-point mode. Then, when all the calculations are done, the numbers will be divided to the same fixed number. In the other hand, the fixed point calculation can be used. The operation of training a cell of neural network in the MATLAB software, and the block diagram of the proposed system are shown in Fig. 9 and Fig. 10, respectively.

4. Results and Discussion

Experimental set up used in this work is composed of an Intel Corei7 CPU with a 2.66 GHz core and 4 GB DDR3. The implementation has been done using EDK tools from Xilinx Company [26]. The selected chip is the low cost xc3s400 from Spartan family. This chip operates at 27 MHz frequency. A summary of implementation and the sources occupied is shown in Fig. 11.

When implementation was done, it is needed to evaluate the performance of the implemented system. There are two traditional standard measures to assess the performance of a biometric system which are called "false acceptance rate (FAR) and false rejection rate (FRR)".


Neural Network Training (nntraintool)

Neural Network

Algorithms

Training: Cyclical Weight/Bias Rule (trainc)
Performance: Mean Absolute Error (mae)
Calculations: MATLAB

Progress

Epoch: 0 14 iterations 1000
Time: 0:00:02
Performance: 0.981 0.00 0.00

Plots

(plotperform)
(plottrainstate)

Plot Interval: 1 epochs

✓ Performance goal met.

Fig. 9. Operation of training a cell of neural network in MATLAB software

- False acceptance rate (FAR) is the probability that the system incorrectly matches the input pattern to a nonmatching template in the database. It measures the percent of invalid inputs which are incorrectly accepted.
- False rejection rate (FRR) is the probability that the system fails to detect a match between the input patterns and a matching template in the database. It measures the percent of valid inputs which are incorrectly rejected.

As it was said in the former section, BioID dataset has been used for evaluation of the presented system. This dataset consists of 1521 gray level images with a resolution of 384x286 pixel. Each one shows the frontal view of a face of one out of 23 different test persons. For comparison reasons the set also contains manually set eye positions [25].

The result of applying the proposed algorithm on one of BioID's image in MATLAB software is shown in Fig. 12. It's clear that the face has been extracted appropriately. The result of applying the implemented algorithm on the same image in Modelsim software is shown in Fig. 13. According to the cascade classification, the face was located on the 17th segment of the original image. As shown in Fig. 13, the face has been properly diagnosed by the implemented system.

The implemented system was evaluated on different statistical populations. For this evaluation, only the images are chosen which have not been used in a network training process. The results of evaluating the implemented system on various statistical populations are presented in Table 1.

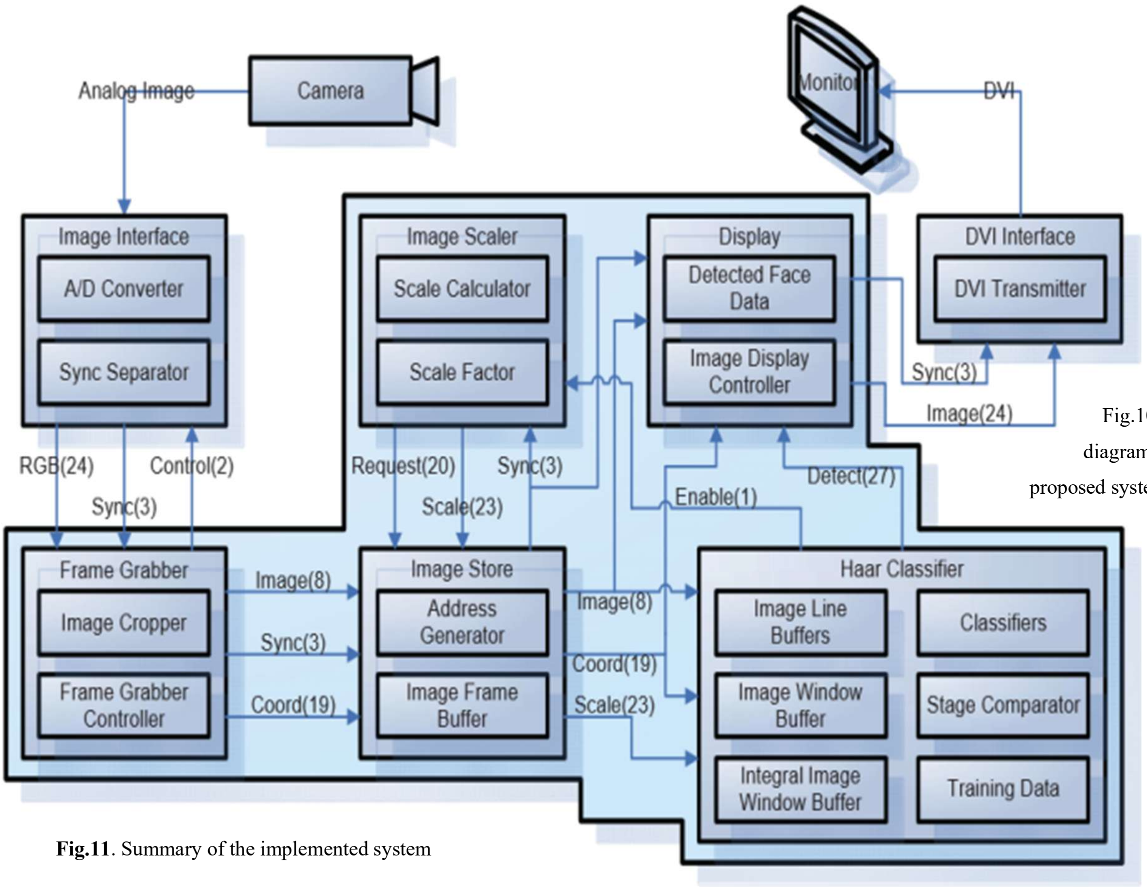


Fig.10. Block diagram of the proposed system

Logi
Numt
Numt
Numt
Numt
Numt
Numt

Fig.11. Summary of the implemented system

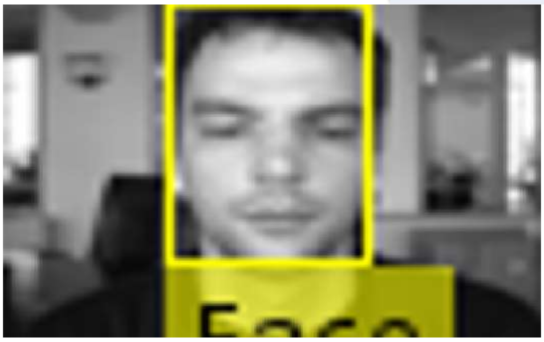
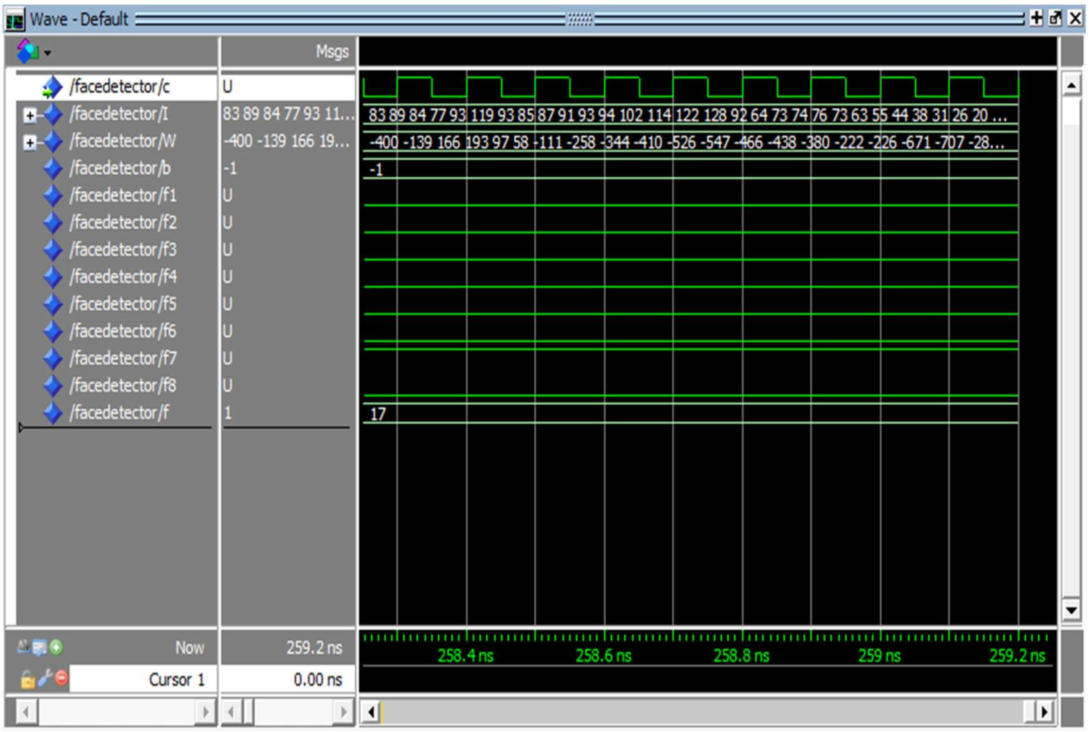


Fig.12. Result of applying the proposed face detection algorithm on one of BioID's image in MATLAB software



```
/facedetector/c     U
/facedetector/I     83 89 84 77 93 11...   83 89 84 77 93 119 93 85 87 91 93 94 102 114 122 128 92 64 73 74 76 73 63 55 44 38 31 26 20 ...
/facedetector/W     -400 -139 166 19...    -400 -139 166 193 97 58 -111 -258 -344 -410 -526 -547 -466 -438 -380 -222 -226 -671 -707 -28...
/facedetector/b     -1                     -1
/facedetector/f1    U
/facedetector/f2    U
/facedetector/f3    U
/facedetector/f4    U
/facedetector/f5    U
/facedetector/f6    U
/facedetector/f7    U
/facedetector/f8    U
/facedetector/f     1                      17

Now        259.2 ns
Cursor 1   0.00 ns
```

Fig.13. Result of applying the implemented face detection system on one of BioID's image in Modelsim software

A plot of recognition accuracy changes in terms of population changing is shown in Fig. 14. In our largest statistical society (500 images), the proposed algorithm resulted in detection accuracy of 93%. On the other hand, with regard to the accuracy trend with the increasing number of images in the statistical populations, it can be said that the proposed method has been appropriately stable in spite of increment in the number of statistical populations and shows less variation. So, the results proved that the proposed algorithm provides not only appropriate detection accuracy but proper stability in results as the number of images increased.

Table 1. Results of evaluating the implemented system on various statistical populations

Number of images used	Accuracy rate of detection (%)	False alarms Average
20	96.00	0.8
50	95.40	2.3
100	94.90	5.1
150	94.40	8.4
200	94.09	12.2
300	93.60	18.9
400	93.30	26.8
500	93.07	34.6

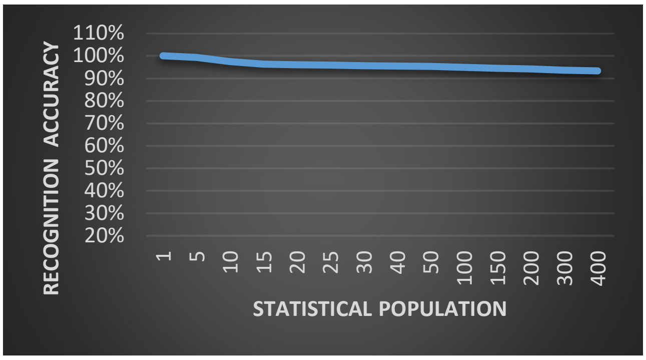


Fig. 14. Recognition accuracy changes vs. statistical population changes

Finally, a comparison with other face recognition methods has been done to show the performance of the implemented system. Many method have been presented for face recognition. One of the simplest method may be the PCA which was first proposed in face detection in 1991 [27]. Another method used widely is "independent component analysis (ICA)" [28]. Also, another method called "local feature analysis (LFA)" is used for face detection [29]. Yang et al. reported a performance comparison between methods based on ICA and PCA [1]. Also, these methods versus a new one has been compared by Leung and his colleagues [30]. The results are illustrated in Table 2. BioID dataset and FERET dataset have been used for this comparison [25,31]. The comparison result shows that the proposed system outperforms the rest of mentioned methods.

5. Conclusions

Up to now, neural network-based methods have been one of the most effective methods for online facial recognition. But these methods have some weakness such as long time required for training or designing an optimum cascade structure. So, in this paper a method was presented to overcome such issues. The presented system contains several new sections; including a series of new rectangular features, improved cascade structure and parallel processing of the input image. Also, it can be generalized easily to other objects like cars, machines, animals etc.. The proposed algorithm was examined first in the MATLAB software. BioID database was used to evaluate the performance of the proposed method. Also, the robustness of the presented system was evaluated in a statistical community changed from 20 to 500 images. While the number of images increases up to 500, the precision of recognition system reduces down to 93% which confirms the proper robustness in results.

Then, the face detection system was implemented on FPGA. In this paper, Spartan3\xc9500 chip from Xilinx Inc. was used as a platform. The performance of the implemented system was evaluated using BioID database. The results proved that the implemented system provides not only appropriate detection accuracy but proper robustness in results as the number of images increased. Finally, a comparison with other face recognition methods has been done to show the performance of the presented system. The comparison result shows that the proposed system outperforms the other mentioned methods.

Table 3. Face recognition rates of the standard PCAs, ICAs, LFA and DLFA techniques in comparison with the proposed system

Reference	Methods	Average recognition rate (%) on the BioID dataset	Average recognition rate (%) on the FERET dataset
[1]	ICA Arch. I	52.15	51.76
	ICA Arch. II	63.77	63.29
	PCA I	52.43	51.83
	PCA II	66.02	63.98
	Standard PCA	52.94	51.64
[29]	LFA	61.8	61
	DLFA	91.22	90.02
[32]	Log-Gabor	86.67	83.14
[33]	Gabor	92.91	91.31
[30]	Feature-based	90.74	89.16
[8]	Modified PCA	92.3	91.57
Proposed system	MLP NN	93.07	91.72

Reference

[1] M.H. Yang, D. Kriegman, N. Ahuja, Detecting face in images: a survey, {IEEE} Trans. Pattern Anal. Mach. Intell. 24 (2002) 34–58.

[2] L. Nanni, A. Lumini, Combining face and eye detectors in a high- Performance face-detection system, IEEE Multimed. 19 (2012) 20–27. doi:10.1109/MMUL.2011.57.

[3] F. Dornaika, A. Moujahid, B. Raducanu, Facial expression recognition using tracked facial actions: Classifier performance analysis, Eng. Appl. Artif. Intell. 26 (2013) 467–477. doi:10.1016/j.engappai.2012.09.002.

[4] S. Mohsen Mousavi, H. Kalbkhani, M.G. Shayesteh, Efficient algorithms for detection of face, eye and eye state,

IET Comput. Vis. 7 (2013) 184–200. doi:10.1049/iet-cvi.2011.0091.

[5] A. Chakrabarty, H. Jain, A. Chatterjee, Volterra Kernel Based Face Recognition Using Artificial Bee Colonyoptimization, Eng. Appl. Artif. Intell. 26 (2013) 1107–1114. doi:10.1016/j.engappai.2012.09.015.

[6] Mamta, M. Hanmandlu, A new entropy function and a classifier for thermal face recognition, Eng. Appl. Artif. Intell. 36 (2014) 269–286. doi:10.1016/j.engappai.2014.06.028.

[7] R. Raghavendra, K.B. Raja, C. Busch, Presentation attack detection for face recognition using light field camera, Image Process. IEEE Trans. 24 (2015) 1060–1075. doi:10.1109/TIP.2015.2395951.

[8] H. Mliki, E. Fendri, M. Hammami, Face Recognition Through Different Facial Expressions, J. Signal Process. Syst. 81 (2015) 433–446. doi:10.1007/s11265-014-0967-z.

[9] K. Jeong, J. Choi, G.J. Jang, Semi-Local Structure Patterns for Robust Face Detection, IEEE Signal Process. Lett. 22 (2015) 1400–1403. doi:10.1109/LSP.2014.2372762.

[10] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks, IEEE Signal Process. Lett. PP (2016). doi:10.1109/LSP.2016.2603342.

[11] A. Mohamed, Y. Weng, J. Jiang, S. Ipson, Face detection based neural networks using robust skin color segmentation, 2008 5th Int. Multi-Conference Syst. Signals Devices. (2008) 1–5. doi:10.1109/SSD.2008.4632827.

[12] H.A. Rowley, T. Poggio, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 23–38. doi:10.1109/34.655647.

[13] R. Hsu, S. Member, M. Abdel-mottaleb, A.K. Jain, Face Detection in Color Images, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 696–706. doi:10.1109/34.1000242.

[14] R. Féraud, O.J. Bernier, J.-E. Viallet, M. Collobert, A fast and accurate face detector based on neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 42–53.

[15] B.Y. Ishak, K.A., Samad, S.A., Hussain, A., Majlis, A Fast and Robust Face Detection Using Neural Networks, in: Proc. Int. Symp. Inf. Commun. Technol., 2003: pp. 1767–1774.

[16] F. Zuo, P.H.N. De With, Fast Face Detection Using a Cascade of Architecture Component : A Neural Network Ensemble, Network. (2005) 26–34.

[17] R.J. Romero-Troncoso, R. Saucedo-Gallaga, E. Cabal-Yepez, A. Garcia-Perez, R.A. Osornio-Rios, R. Alvarez-Salas, H. Miranda-Vidales, N. Huber, FPGA-based online detection of multiple combined faults in induction motors through information entropy and fuzzy inference, IEEE Trans. Ind. Electron. 58 (2011) 5263–5270. doi:10.1109/TIE.2011.2123858.

[18] M. Wojcikowski, R. Zaglewski, B. Pankiewicz, FPGA-Based Real-Time Implementation of Detection Algorithm for Automatic Traffic Surveillance Sensor Network, J. Signal Process. Syst. Signal Image Video Technol. 68 (2012) 1–18. doi:10.1007/s11265-010-0569-3.

[19] M. Fons, F. Fons, E. Cantó, M. López, FPGA-based personal authentication using fingerprints, J. Signal Process. Syst. 66 (2012) 153–189. doi:10.1007/s11265-011-0629-3.

[20] M. Ashourian, N. Daneshmandpour, O. Sharifi Tehrani, P. Moallem, Real time implementation of a license plate location recognition system based on adaptive morphology, Int. J. Eng. Trans. B Appl. 26 (2013) 1347–1356. doi:10.5829/idosi.ije.2013.26.11b.10.

[21] A. Mandal, R. Mishra, Design and implementation of digital demodulator for frequency modulated CW radar, Int. J. Eng. Trans. A Basics. 27 (2014) 1581–1590. doi:10.5829/idosi.ije.2014.27.10a.12.

[22] A.K. Dwivedi, S. Ghosh, N.D. Londhe, Low power FIR filter design using modified multi-objective artificial bee colony algorithm, Eng. Appl. Artif. Intell. 55 (2016) 58–69. doi:10.1016/j.engappai.2016.06.006.

[23] M. Yuan, Field Programmable Gate Array Implementation of Active Control Laws for Multi- mode Vibration Damping, Int. J. Eng. Trans. B Appl. 29 (2016) 229–235.

[24] M.H. Zivarian, H., Soleimani, M., Doost Mohammadi, FPGA – based Implementation of an Improved Algorithm for Objects Distance Measurement, Int. J. Eng. (IJE), Trans. A Basics. 30 (2017) 1874–1881.

[25] BioID AG, BioID face database, (2001) 1–4. http://www.bioid.com/support/downloads/software/bioid-face-database.html (accessed January 1, 2016).

[26] Xilinx EDK tools Free Web pack, Xilinx Inc. (2013). https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/design-tools.html (accessed January 1, 2016).

[27] M. Turk, A. Pentland, Eigenfaces for Recognition, J. Cogn. Neurosci. 3 (1991) 71–86. doi:10.1162/jocn.1991.3.1.71.

[28] M.S.Bartlett, H.M.Lades, T.J.Sejnowski, Independent Component Representation for Face Recognition, in: Proc. SPIE Symp. Electron. Imaging Hum. Vis. Electron. Imaging, Vol. 3299, 1998: pp. 528–539.

[29] J. Ng, H. Cheung, Dynamic Local Feature Analysis for Face Recognition, Biometric Authentication. (2004) 1–13. http://www.springerlink.com/index/CTY76AA2LXLP15G5.pdf.

[30] H.-Y. Leung, L.-M. Cheng, X.Y. Li, A FPGA implementation of facial feature extraction, J. Real-Time Image Process. 10 (2012) 135–149. doi:10.1007/s11554-012-0263-8.

[31] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, {T}he {FERET} {D}atabase and {E}valuation {P}rocedure for {F}ace {R}ecognition {A}lgorithms, Image Vis. Comput. 16 (1998) 295–306. doi:doi:10.1016/S0262-8856(97)00070-X.

[32] T. Ahonen, S. Member, A. Hadid, S. Member, M. Pietika, Face Description with Local Binary Patterns: Application to Face Recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 2037–2041. http://www.computer.org/portal/web/csdl/doi/10.1109/tpami.2006.244.

[33] A. Bhuiyan, C.H. Liu, On Face Recognition using Gabor Filters, 2 (2007) 837–842.