Human Face Detection using Evolutionary Optimized Neural Network

Ali Al-Hamdi [*] and Salem Bajnuf [**]

[*] Sana'a University, Faculty of Engineering, Electrical Department, Sana'a
[**] University of Science and Technology, Faculty of Computer Science and Information, Sana'a,
e-mail: nalhamdi@yemen.net.ye

ABSTRACT

Face detection is a challenging computer vision problem. Given a still image or a sequence of images, the goal of face detection process is to locate all regions that contain a face regardless of any three dimensional transformation and lighting condition. In this paper, we propose an optimized Neural-Network with a multi-layer using evolutionary algorithm for human face detection. The proposed method integrates the neural network and genetic algorithm in order to reduce time detection. The influence of the number of hidden units in a network topology was also investigated. To evaluate our system and to investigate the effect of the optimizing algorithm, a number of tests were performed in a large number of images. Results obtained show that this method has speeded up face detection process with high detection rate.

1. INTRODUCTION

Nowadays, research on computer vision is spreading enormously so that it is almost impossible to itemize all of its subtopics. One of these subtopics is human face recognition which witnessed a significant interest in many applications such as computer human interaction, crowd surveillance, content-based image retrieval, etc… All of these applications require automatic face recognition in which face detection represents a corner stone and can be viewed as a preliminary step in order to be able to recognize the face. In fact, good face detection will simplified the next steps of an automatic face recognition system such as facial feature location and face recognition.

Depending on utilizing face knowledge, face localization techniques can be classified into two broad categories approaches*: feature-based approaches* and *Image-based approaches* [1, 2]. In general, image-based approaches rely on machine learning and statistical analysis. Hence, image-based approaches utilize more complex techniques particularly learning algorithms such neural network [3, 4] and statistical methods [5] to overcome the high dimensionality of the problem space. Generally, one can select neural network technique due to its high accuracy for face detection and effectiveness in discrimination between face and non-face patterns when trained with huge data. However, the main drawback of NNs is the computational cost i.e. time consuming, due to their architecture tuning specially for process all possible positions with varied resolutions of a raw image. So, in order to overcome this drawback, one can think in using evolutionary algorithms [6, 7]. In this paper we propose an efficient face detection method, which uses evolutionary algorithm (genetic algorithms) to optimize the neural network topology, hence speed up the human face detection process in an image.

In addition to this introduction, this paper is organized in four sections. In the next section an overview of our method is presented. Section 3 describes the optimization evolutionary algorithm for NNs structure. The experimental results are discussed in section 4. In section 5 conclusions and future work will be drawn.

2. SYSTEM OVERVIEW

Our implementation, which can be seen in figure 1, is a simulated version neural network-based face detection system developed by Rowley et al [4]. The main difference between Rowley's work and our work is in the network architecture (topology) and the learning algorithm. From topological perspective, Rowley used 20x20 nodes in input layer and 26 nodes in hidden layer in his system. From algorithm point of view, Rowley used back-propagation as learning algorithm. In our system, we used 25x25 nodes in input layer, 50 nodes in hidden layer for the network structure and we used genetic.

As shown in figure 1, each image is subjected to a pre-processing phase before being fed to the neural network. The preprocessing phase plays an essential role for preparing the input image to be classified by the neural network. This phase includes different steps of image processing such as image scaling, image chromaticity conversion, image enhancement such as illumination correction and histogram equalization.

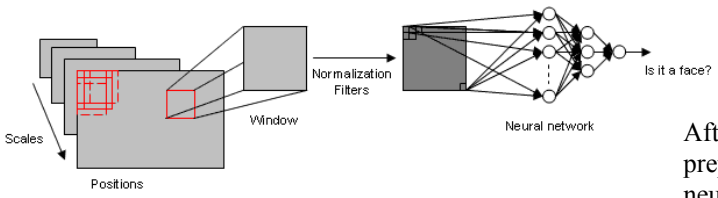

Figure 1: a short description about our approach.

Image rescaling which resample the raw image for providing scale invariant face detection in a fixed size window. Thus, to detect larger or smaller faces than the window size, the input image is repeatedly reduced or increased in size (by sub-sampling) respectively, and the window is applied at each size. To detect faces anywhere in the image, the window is applied at every location in the image, we put an important parameter (windows interval) to control moving this window throw image, we can choose number of pixels to move this window. Changing of this parameter is manually, we decrease it if there is no positive detection in the image.

To avoid image variation such as the diversity in race, color, genre, different lighting conditions, etc… each fixed window is pre-processed before being fed to the classifier. First a color to gray-scale image conversion is made using standard system. Next, image illumination correction is performed via a best-fit brightness plane which is subtracted from the window pixel values. In our work, the size of the standard window is 25x25 pixels. As in the previous work [3], [4], we fit a function which varies linearly across the window to the intensity values in an oval region (mask) inside the window. Pixels outside the oval may represent the background, so those intensity values are ignored in computing the lighting variation across the face. Finally, the resulted images are stretched by histogram equalization in order to compensate differences in brightness, camera responses, skin color etc… The results of these steps of preprocessing to obtain a normalized image are given in figure 2.

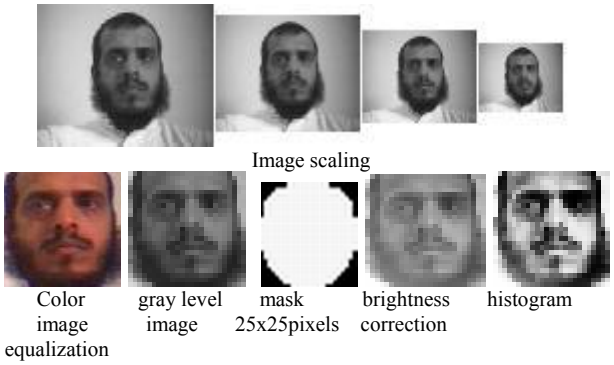

Figure 2: image preprocessing steps.

3. MULTILAYER NN TOPOLOGY OPTIMIZATIO USING GENETIC ALGORITHM

After the first phase in which an input image is preprocessed, a window 25x25 pixel is passed into a neural network to classify the input data as face or non-face. Artificial Neural Networks are popular approach for the problem of pattern classification. Very important features of ANNs are their adaptive quality and their ability to generalize, which allow them to be effective in classification task particularly when huge data are dealt. However, one of the main problems of ANNs is the definition of a good or optimal topology to adequately solve the problem in hand. In fact, the quality of the solution offered by ANNs strongly depends on the number of layers, the number of neurons (nodes), and their organization (in terms of connection) as well. Another important consideration in ANNs is the use of appropriate learning or training algorithm. There are hundreds of them, but it is best to use a well-known and well-characterized one. The basic learning algorithm used in ANNs is Back Propagation (BP). However, this algorithm has some drawbacks: the most important of them is time-consuming. In fact, learning neural network weights can be considered a hard optimization problem for which the learning time scales exponentially becoming prohibitive as the problem size grows [8]. Therefore it is evident to find the best learning technique that gives the best structure in order to overcome the time-consuming phase. So, one can think of using the evolutionary algorithms.

In fact, evolutionary algorithms have been applied successfully as optimization techniques in many applications in general [6] and in particular for the topology of neural networks for face detection [7]. Our approach is to use evolutionary algorithm to yield the network architecture optimization and contemporaneously choose the best technique to update the weights of the BP optimizing the related parameters. In this context, an automatic procedure is presented to train the neural network and to optimize its topology simultaneously. The optimization loop of our hybrid genetic algorithm is given in subsection 3.2.

The goal of our work is to find the most appropriate neural network topology in a minimum time of network training (iteration number). To do so, the network optimization is characterized by determining two parameters: number of hidden layers NHL and number of layer nodes LN. We gave NHL a maximum value of 2 i.e. two types of topologies hopping that “the smaller the network the better the generalization” [8]. The first topology is a single hidden layer and the second is double hidden layers. For these two topologies, the value of LN is allowed to vary from 10 to 100 and from 5 to 50 respectively.

3.1. Fitness function

The goodness of an individual is evaluated by training the network with a fixed number of patterns and then evaluated according to determined parameters. These parameters which seem to better describe the fitness of a network topology are the mean square error, E at the end of training and the number of epochs ep needed for the learning as stated in [9]. While the objective of the first parameter is to determine the optimal network structure, the second objective is to minimize the training time. Hence, it is desirable to attain both parameters E and ep as low as possible with the following limits for each one.

$$0 < ep < ep_{max}. \quad (1)$$
$$0 \leq e_{min} \leq E. \quad (2)$$

Since the heuristic methods are implemented for the minimization, the fitness function has been chosen as a function increasing when ep and E increase. Specifically, the fitness function is :

$$f(x) = \frac{ep}{ep_{\max}} + E \quad (3)$$

The above equation can take 1 if and only if the learning phase is such as that the minimum error required is reached by the network within a number of epochs lower than ep_{max}. Note that e_{min} is generally chosen equal to 10^{-a} with $a > 2$.

3.2. Optimization loop

The optimization loop of our evolutionary network is adopted from the work of [9] using direct encoding, nested learning, and Lamarckian inheritance is the following:

Given a pattern set p for the network training;
Procedure Evolutionary Algorithm
Begin
Randomly initialize a population of μ neural network structures;
while (termination criterion not fulfilled) do
train x_i by means of MLP (x_i) on *P* **;**
evaluate the trained network x_i ;
save the best trained network in the new population;
select the best λ neural network configurations;
For *i*=1 to μ - 1do
Create one offspring;
Od
Update variables for termination;
Od
End

The procedure is constituted by an evolutionary algorithm and by a procedure for training the MLP encapsulated in the main program.

4. EXPERIMENTAL RESULTS AND DISCUSSION

To train a neural network for face detection task, a large number of face and non-face samples are needed. It may be easy to collect a representative set for face patterns, but unfortunately the same is not true for non-face patterns.

For the face samples, we collected more than 100 images for training data set from our environment and from the Internet. These samples contain faces of different sizes, orientations, intensities, and positions. In each sample, the eyes, nose, and right, left and center of the mouth were hand cropped. Then each one is normalized into 25x25 to give face sample.

We collected more than 400 for non-face images from tow sources with simple and complex background. The first source images were gathered from the Web. However, due to the difficulty of obtaining this type of samples and to avoid huge data storage, the second source of this type of samples is obtained using bootstrapping technique. During training the face images, the non-face detection samples were collected and added to the train sets.

In this work we intend to study how different network topologies affect the classification performance so as to have the best topology. In our system we initialized the parent population with a fixed individual of 50 that represent a 652-50-1 network structure. The operators are the crossover and mutation with probability of 0.8 and 0.6 respectively. The vale of e_{min} has been specified to 10^{-5} in order to obtain the most ideal results. Ten trials of genetic algorithm have been executed with a termination criterion of 100 as a maximum number of generations. The value of ep_{max} has been set equal to 1000 in order to allow the genetic algorithm to cover a limited number of generations, so to reduce the network error at the end of training phase.

As first step, we conducted a number of tests to investigate the influence of the number hidden layers and hidden neurons in order to obtain the optimal topology. First a neural network with only one hidden layer is exploited. The number of nodes in the single hidden layer is varied from 10 to 100 as mentioned earlier and the mean squared error of the training set is recorded over the whole run. The results (table 1) obtained from this experiment showed that the best structure is 652-50-1 with value of E equal to 0.02 followed by 625-40-1 and 625-20-1 with values of E equal to 0.021 and 0.03 respectively. Then, we tried to exploit the multiple hidden layers and see if increasing the number of layers gives us better performance. In this structure the number of units in the second hidden layer varies from 5 to 50 as stated in section 3. The best topology obtained is 625-20-5-1 and 625-50-10-1 with

values of E equal to 0.025 and 0.04 respectively. Tables 2 and 3 show the results obtained using these topologies.

Table 1: Variation of training error with varied number of nodes in a network with single hidden layer.

LN	10	20	30	40	50	60	70	60	90
E	0.15	0.03	0.05	0.021	0.02	0.035	0.045	0.9	0.86

Table 2: Variation of training error with varied number of nodes in a network with double hidden layer (525-20-x-1).

NHL	5	10	15	20	25	30	35	40	45
E	0.025	0.039	0.5	0.4	0.74	0.67	0.45	0.78	0.82

Table 3: Variation of training error with varied number of nodes in a network with double hidden layers (525-50-x-1).

NHL	5	10	15	20	25	30	35	40	45
E	0.25	0.04	0.09	0.42	0..34	0.55	0.045	0.08	0.68

The second step of our work deals with training the optimal topologies obtained in the first step in order to investigate the optimal training time. So, each topology from the previous step has been trained for several epochs varying from 100 to ep_{max} in 10 steps. Ideally, we would want the number of nodes to be as small as possible for faster computation when we search an image for localization, but if the number of hidden layer nodes is too small the network might not be able to capture the desired characteristics completely.

The results obtained from this step showed that the best topology 625-50-10-1 with fitness value of 0.115 followed by 625-20-5-1 with fitness value of 0.302 for double layers network. Table 4 and table 5 show the variation of training error of double layers network.

Table 4: Variation of training error of network with double hidden layers (525-20-5-1).

ep	100	200	300	400	500	600	700
E	0.02	0.007	0.002	0.008	0.01	0.025	0.03

Table 5: Variation of training error of network with double hidden layers (525-50-10-1).

ep	100	200	300	400	500	600	700
E	0.015	0.04	0.08	0.07	0.06	0.065	0.07

The single hidden layer topologies, 625-20-1 and 625-50-1 resulted fitness values of 0.6002 and 0.5004 respectively as shown in figures 6 and 7.

Table 6: Variation of training error of network with single hidden layer (525-20--1).

ep	100	200	300	400	500	600	700	800
E	0.07	0.035	0.018	0.03	0.03	0.002	0.003	0.007

Table 7: Variation of training error of network with single hidden layer (525-50--1).

ep	100	200	300	400	500	600	700
E	0.01	0.001	0.0009	0.0002	0.00004	0.00005	0.00004

From these results, one can deduce that the argument "the smaller the network the better the performance" does not necessary hold as was stated in [7].

To evaluate our system and to investigate the effect of the genetic algorithm, a number of tests were performed in more than 150 images taken from real environment and downloaded from the WWW. Some detection examples of our work are depicted in figure 3. From this figure, we can see that the proposed system is quite robust against low image quality and variations of face scale, appearance, some orientation, acquisition conditions, etc…

Figure 3: detection examples of applying our approach.

5. CONCLUSION

In this paper, a neural network-based method is presented for human face detection. An evolutionary algorithm is used to classify face and non-face patterns in order to optimize both network topology and learning time. The proposed method has shown its performance on testing a number of images with different features.

Although the results are satisfied, one can think of further improvements in order to decrease the hidden nodes and reducing the learning time. Thus, more advanced evolutionary algorithms could be promising techniques [10].

REFERENCES

[1] M.-H. Yang, David J. Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey", IEEE JAN 2002, 34-58.

[2] E. Helomas and B. K. Low. "Face detection: A survey". *Computer Vision and Image Understanding*, 83:236–274, 2001.

[3] S.-H. Lin, S.-Y. Kung, and L.-J. Lin. "Face recognition/detection by probabilistic decision-based neural network". *IEEE Trans. Neural Networks*, 8:114–132, 1997.

[4] H. A. Rowley, S. Baluja, and T. Kanade. "Neural network-based face detection". *IEEE Trans. Pattern Analysis Machine Intelligence*, 20:23–38, 1998.

[5] H. Schneiderman and T. Kanade. "A statistical model for 3d object detection applied to faces and cars". *IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

6] X. Yao, "Evolving artificial NNs" Proceedings of the IEEE, 87 (9), pp: 1423-1447, 1999.

[7] Stefan Widgand *et al*, "Evolutionary optimization of Neural Networks for face detection", 12th European Symposium on Artificial Neural Networks (ESANN 2004), Evere, Belgium: d-side duplications, 2004.

[8] R Caruana *et al*, "Overfitting in neural networks: Backpropagation, conjugate gradient,and Early stopping" In Advances in neural Information Processing system ,volume 13 ,pages 402-408 ,Denver ,Colorado , 2001, MIT Press.

[9] Ivanoe De Falco, "Nonlinear ayatem identification by means of evolutionary optimized Neurla Networks", John Wiley & sons Ltd, pp: 367-391, 1997.

[10] D. Quagliarella *et al*, "Genetic Algorithms and Evolution Strategy in Engineering and Computer Science", JOHN WILEY & SON, Chichester, England, 1998.