ISTANBUL UNIVERSITY ENGINEERING FACULTY
JOURNAL OF ELECTRICAL & ELECTRONICS
YEAR : 2002 (601-609)
VOLUME : 2
NUMBER : 2

EXTRACTION OF FACIAL FEATURES USING GENETIC CELLULAR NEURAL NETWORKS

[1]Osman Nuri UCAN **[2]Erdem BILGILI** **[3]Ramazan COBAN**

[1]Istanbul University, Electrical -Electronics Engineering Department,
34850, Avcilar,Istanbul, Turkey (Corresponding address)

[2]Tubitak, MAM, MKTAE, TUOAL, Gebze ,Kocaeli

[3]Gebze Institue of Technology, Electronics Engineering Department, Gebze, Kocaeli

[1]E-mail: uosman@istanbul.edu.tr

ABSTRACT

This paper presents proper CNN templates to locate face region consisting of eyes,eyebrows, nose and mouth in frontal face images. Training of CNN is done by using genetic algorithms. At the begining, 72 random genes are selected. Each gene includes binary codes of the elements of the CNN templates. CNN templates are designed as symmetric to satisfy the stability. After the training process CNN templates are applied to various images.

Keywords: cellular neural network, Genetic Algorithm

1. INTRODUCTION

Most of the widely applied neural networks fall into two main classes: (1) memoryless neural networks and (2) dynamical neural networks. As in Hopfield networks (HN) and CNN, dynamical neural networks have usually been designed as dynamical systems where the inputs are set to some constant values and each trajectory approaches one of the stable equilibrium points depending upon the initial state. Cellular Neural Network is a large-scale non-linear analog circuits which processes signals in real time[1-2].

The network behaviour of CNN depends on the initial state of the cells activation, namely bias I and on weights values of A and B matrices which are associated with the connections inside the well-defined neighbourhood of each cell. There are many approaches in estimation of A,B, I matrices. Here we prefer genetic algorithm. Genetic algorithm is a learning algorithm based on the mechanism of natural selection and genetics, which have proved to be effective in a number of applications. It works with a binary coding of the parameter set, searches from a

Received Date : 17.6.2001
Accepted Date: 25.5.2002

number of points of the parameter space. It uses only the cost function during the optimization, it need not derivatives of the cost function or other information. [3-9].

In this paper, we focuse on face detection which is much significant in various application as virtual reality, intelligent human-computer interface and TV-conference, security system.[5,6,7]. First step in face dedection is locating face and facial features [5],[8]. Then, the detected faces have to be normalized and recognized by specially designed classifier. In this study, we find the facial features and face region using CNN templates estimated by using genetic algorithms.

2. Cellular Neural Networks

Like cellular automata, the CNN is made of a massive aggregate of regularly spaced circuits clones, called cells, which communicate with each other directly only through nearest neighbors. In Figure 2, each cell is modelled as squares. The adjacent cells can interact directly with each other. Cells not directly connected together may affect each other indirectly because of the propagation effects of the continous-time dynamics of cellular neural networks. An example of a two-dimensional CNN is shown in Figure 3. Now let us define the neighbourhood of C(i,j).

Definition : r-neighbourhood

The r-neighbourhood of a cell C(i,j) , in a cellular neural network is defined by,

$$N_r(i,j)=\{C(k,l)\,|\,\max\{|k-i|,|l-j|\}\le r\}$$
$$1\le k\le M;\,1\le l\le N \qquad (1)$$

where r is a positive integer number.

Figure 3 shows neighbourhoods of the C(i,j) cell (located at the center and shaded) with r=1 and 2, respectively. To show neighbourhood relations more clearly, the center pixel is coloured as black and related pixels in brown in Figure 3 and 4. Cells are multiple input-single output nonlinear processors all described by one, or one among several different, parametric functionals. A cell is characterized by a state variable, that is generally not observable as such outside the cell itself. It contains linear and non-linear circuit elements such as linear resistors, capacitors and non-linear controlled sources (Figures 4 and 5). Every cell is connected to other cells within a neighbourhood of itself. In this scheme, information is only exchanged between neighbouring neurons and this local information characteristic does not prevent the capability of obtaining global processing. The CNN is a dynamical system operating in continous or discrete time. A general form of the cell dynamical equations may be stated as follows:

$$\frac{dx_{ij}(t)}{dt}=-x_{ij}(t)+\sum_{kl\in N_r(ij)}A_{(i-k)(j-l)}(t)\,y_{kl}+\sum_{kl\in N_r(ij)}B_{(i-k)(j-l)}(t)\,u_{kl}+I$$

$$y_{ij}(t)=\frac{1}{2}\left(\left|x_{ij}(t)+1\right|-\left|x_{ij}(t)-1\right|\right) \qquad (2)$$

where x,y,u,I denote respectively cell state, output, input, bias and j and k are cell indices. CNN parameter values are assumed to be spaced-invariant and the nonlinear fuction is chosen as piece-wise linear (Figure 5).

Since we use discrete 2-D images, Equation (2) is rewritten as,

$$x_{ij}(n+1)=-x_{ij}(n)+\sum_{kl\in N_r(ij)}A_{(i-k)(j-l)}(n)\,y_{kl}+\sum_{kl\in N_r(ij)}B_{(i-k)(j-l)}(n)\,u_{kl}+I$$

$$y_{ij}(n)=\frac{1}{2}\left(\left|x_{ij}(n)+1\right|-\left|x_{ij}(n)-1\right|\right)$$

(3). with A, B and I being cloning template matrices that are identically repeated in the neighbourhood of every neuron as,

$$A=\begin{bmatrix}A_{-1,-1} & A_{-1,0} & A_{-1,1}\\ A_{0,-1} & A_{0,0} & A_{0,1}\\ A_{1,-1} & A_{1,0} & A_{1,1}\end{bmatrix},\quad B=\begin{bmatrix}B_{-1,-1} & B_{-1,0} & B_{-1,1}\\ B_{0,-1} & B_{0,0} & B_{0,1}\\ B_{1,-1} & B_{1,0} & B_{1,1}\end{bmatrix},\ I. \qquad (4).$$

The network behaviour of CNN depends on the initial state of the cells activation, namely bias I and on weights values of A and B matrices which are associated with the connections inside the well-defined neighbourhood of each cell. CNN's are arrays of locally and regularly interconnected neurons, or, cells, whose global functionality are defined by a small number of parameters (A,B, I) that specify the operation of the component cells as well as the connection weights between them. CNN can also be considered as a nonlinear convolution with the template. Cells can be characterized by a functional block diagram that is typical of neural network theory: Figure 4 depicts a two-stage functional block diagram of a cell, composed of a generalized weighted sum (in general nonlinear with memory) integration, output nonlinear function/functional. Data can be fed to the CNN through two different ports: initial conditions of the state and proper input u. Bias values I may be used as a third port.

The network behaviour of CNN depends on the initial state of the cells activation, namely bias I and on weights values of A and B matrices which are associated with the connections inside the well-defined neighbourhood of each cell. CNN's are arrays of locally and regularly interconnected neurons, or, cells, whose global functionality are defined by a small number of parameters (A,B, I) that specify the operation of the component cells as well as the connection weights between them. CNN can also be considered as a nonlinear convolution with the template. Since the introduction of Chua [1], CNN has attracted a lot of attention. Not only from a theoretical point of view these systems have a number of attractive properties, but furthermore there are many well-known applications like image processing, motion detection, pattern recognition, simulation. The reduced number of connections within a local neighbourhood, the principle of cloning template etc., turn out to be advantage of CNN's.

3.Genetic Algorithms

In the estimation of A.B. and I matrices of CNN, we use genetic algorithms. Genetic algorithm is a learning algorithm based on the mechanism of natural selection and genetics, which have proved to be effective in a number of applications. It works with a binary coding of the parameter set, searches from a number of points of the parameter space. It uses only the cost function during the optimization, it need not derivatives of the cost function or other information. [4,9]. Processes of natural selection cause chromosomes that encode successful structures to reproduce more often than those that do not. In addition to reproduction, mutations may cause the chromosomes of children to be different from those of their biological parents, and crossingover processes create different chromosomes in children by changing the some parts of the parent chromosomes between each other. Like nature, genetic algorithms solve the problem of finding good chromosomes by manipulating in the chromosomes blindly without any knowledge about the problem they are solving.[9]. The underlying principles of GA were first published by Holland in1962 ,[10] . The mathematical framework was developed in the 1960s and is presented in his pioneering book in 1975 [11]. In optimization applications, they have been used in many diverse fields such as function optimization, image processing, the traveling salesperson problem, system identification and control. A high-level description of GA has been done by Davis in 1991 as follows. [12]

Given a way or a method of encoding solutions of problem into the form of chromosomes and given an evaluation function that returns a measurement of the cost value of the following steps:

Step1:Initialize a population of chromosomes
Step2:Evaluate each chromosomes in the population.
Step3:Create new chromosomes by mating current chromosomes ; apply mutation and recombination as the parent chromosomes mate.
Step4: Delete members of the population to make room for new chromosomes.
Step5: Evaluate the new chromosomes and insert them into the population.
Step6:If the stopping criterion is satisfied, then stop and return the best chromosome; otherwise , go to step3

4. Locating Facial Features by CNN

In this work, proper CNN templates are described to locate face and facial features in image by using genetic algorithms. For this aim, CNN templates are designed so that they satisfy the stability. So, A and B templates are selected as symmetric. Because of selecting size of templates as 3*3, totally 11 template parameters are searched. One of these parameters is offset, five of them belongs the A matrix, and the other five parameters belongs the B matrix. Each parameter are encoded by 16 bits in chromosomes. So,the length of chromosomes has been selected as 176 bits.

In training process, 72 chromosomes are constructed as initial population randomly. The number of population is kept constant as 72 during the algorithm. Mutation probability p_m has been set % 1 . Training process includes these steps as follows.

(i).Construct initial population ; A matrix is constructed called as population matrix. Each row of the population matrix represents chromosomes. Because of selecting number of chromosomes is 72, there are 72 rows in population matrix. Number of colomns of this matrix is 176, because there are 176 bits in each chromosome. At the begining this matrix is constructad randomly.

(ii). Extract the CNN template: Chromosomes represents the binary codes of the elements of the CNN template A, B, I . In this step, each chromosomes are decoded the elements of the CNN are computed in [-8,8] interval. Since each element is coded as 16 bits, each parameter can take 2^{16} different value in [-8,8] interval. In each chromosomes first 11 bits represents first bits of the template elements. And second 11 bits of chromosomes represents the second bits of the template elements and so on. These elements are

$$S = \left[A_{1,1}, A_{1,2}, A_{1,3}, A_{2,1}, A_{2,2}, B_{1,1}, B_{1,2}, B_{1,3}, B_{2,1}, B_{2,2}, I \right] \quad (5)$$

(iv). Evaluate cost function value for each chromosomes; In this step, an image which was selected as training image is given as input to CNN. Normally in this gray-level image, brightness varies in 0 (black) through 1(white) interval. To fit this image to CNN operation, brightness of the image is converted from [0,1] to [-1,1]. According the same rule, brightness of the CNN output image is converted from [-1,1] to [0,1].Then CNN works with templates belonging with first chromosome. After the CNN output appears as stable, cost function is computed between this output image and target image which we want to obtain. This process is repeated with template sets belongs each chromosomes in the population. Cost function has been selected in this study as follow

$$\cos t(A, B, I) = \sum_{i}^{m} \sum_{j}^{n} P_{i,j} \oplus T_{i,j} \quad (6).$$

where A, B, I represents CNN templates, m, n represents number of pixels of the image, P and T represent input and target image, respectively, notation $\oplus$ represents XOR operation between each elements of the P and T .

After the finding the cost function, fitness function is evaluated for each chromosome according this rule;

$$fitness(A, B, I) = m * n - \cos t(A, B, I) \quad (7).$$

Another definition has been defined for stopping criterion as follows;

$$stcriterion = 0.99 * m * n \quad (8).$$

where m represents the number of rows of the image matrix and n represents the number of the coloumns of the image matrix.

If the maximum fitness value of the chromosomes is greater than stop criteron, algorithms is stopped and the chromosome whose fitness value is the maximum fitness in the population is selected. The templates which has been extracted from this selected chromosomes are the most proper the templates which satisfy the task we wanted to realize.

(v). Creating new generation; Before creating next generation, fitness values of the population are sorted by descendent order. And all of the fitness values are normalized related to the sum of the fitness values of the population. A random number r between 0 and 1 is generated. Then the first population member is selected whose normalized fitness, added to normalized fitnesses of the proceedings population members, is greater than or equal to r . This operation is repeated 72 times. So, the cromosomes whose fitness are bad are deleted from the population. This procedure mentioned above is called reproduction process in genetic algorithms. Reproduction process does not generate new chromosomes. It elect the best chromosomes in the population and increases the number of the chromosomes whose fitness values are relatively greater than the others.

After the reproduction 36 pairs of chromosomes are selected as parents randomly. Two numbers $s1, s2$ between 1 and length of chromosomes,176 are generated. The bit strings between s1 and s2 are called crossover site. During the crossingover process, bit strings in crossover site in each pair of chromosomes are interchanged. Then two new chromosomes are created from a pair of old chromosomes. At the last, 72 new chromosomes which are called children are generated to build new population. Over these chromosomes, mutation operation is processed. Since mutation probability has been set to %1, 126 bits are selected randomly in the population and they are inverted. And the chromosome whose fitness value was the best before the reproduction process is added instead of deleted a chromosome which randomly selected in the population that was obtained after the mutation process population to save the best chromosome. This new population is the next generation population. After the obtaining the new generation searching procedure goes to second step and goes on untill the stopping criterion was happened in the second step.

At the end of the training process following templates were found after the 223 generation.

$$A = \begin{bmatrix} 2.7612 & 7.3152 & 1.7566 \\ 1.5916 & 8.5273 & 1.5916 \\ 1.7566 & 7.3152 & 2.7612 \end{bmatrix} \quad (9).$$

$$B = \begin{bmatrix} -6.1912 & 2.8350 & -7.9270 \\ 1.3044 & -2.7349 & 1.3044 \\ -7.9270 & 2.8350 & -6.1912 \end{bmatrix} \quad (10).$$

and

$$I = 0.4414 \quad (11).$$

The best chromosome which gives these templates is;

BC={
111101 0 1010100100 00101 1001010000 0
11111001 1010110110 0001110011 011100
0000000010 1010011001 0000110010 0000
1111000100 111010110001 1 01101 111110
0 01 1110100100 10110111 1000111001 110
101100
}. (12)

5. Experiments

CNN templates are trained using Figure 5 as input and Figure 6 as an desired set. The optimum A,B,I matrices are found to be as in Equation (9-11) and the best suited chromosome giving these values are given in Equation 12. Then we check our optimised values by applying genetic CNN to the pictures given in Figure 7. At the output of CNN, in all these pictures background ,eyes and eyesbrows appears to be as black and face region appears to be as white. Since the brightness of the bottom parts of the background of the Figure 7 (e) and Figure 7 (k) , output of these images include white regions in the background area as shown in Figure 7 (f) and Figure 7(l), respectively. The images in Figure 7(g) and Figure 7 (i) have background with high brightness related to the training image. When CNN operates over these figures, background of the output image appeared as white. But, we can detect face region by using any simple filter ,easily. Because inside of the face region is most important in face detection algorithms.

6.Conclusions

Describing the location of the facial organs and the face in an image is very important for face recognition application. In this study CNN templates are obtained, which they extract the facial organs; eyes, eyesbrow and face region when they can be applied to an image . Facial organs are represented as black colored pixel and face is region represented white colored pixels. The pixels along the face boundary are black at the output of the image. A little numbers of them of them may be white when the brightness of the backround is very high. If the brightness of the background is low value, CNN with these templates satisfy very good performance.

References

[1] L.O. Chua, L. Yang : ‘‘Cellular Neural Networks: Theory’’, IEEE Trans. On Circuit and Systems, Vol.35, pp. 1257-1272, 1988

[2] L.O. Chua, L. Yang : ‘‘Cellular Neural Networks: Application’’, IEEE Trans. On Circuit and Systems, Vol.35, pp. 1273-1290, 1988

[3] T. Kozek, T. Roska, L.O. Chua, : ‘‘Genetic Algorithms for CNN template Learning’’, IEEE Trans. On Circuit and Systems, Vol.40, No.6 pp. 392-402, 1988

[4] T. Sziranyi, M. Csapodi, ‘‘ Texture Classification and Segmentation by Cellular Neural Network Using Genetic Learning’’, Research Report ,Budapest,Hungary, November,1994

[5] J.Hu, H.Yan, M. Sakalli, ‘‘Locating head and face boundaries for head-shoulder images’’, Pattern Recognition 32 (1999) 1317-1333

[6] A. Samal, P.A. Iyengar, ‘‘Automatic recognition and analysis of human faces and facial expressions : a survey’’, Pattern Recognition 25 (1) (1992) 65-77

[7] R. Chellappa, C.L. Wilson, S. Sirohey, ‘‘Human and machine recognition faces: a survey’’, Proc. IEEE 83 (5) (1995) 705-740.

[8] S.H. Jeng, H.Y.M. Liao, C.C.Han, M.Y. Chern, Y.T.Liu ‘‘Facial feature dedection using geometrical face model: an efficient approach’’

[9] C.T. Lin, C.S. George Lee ‘‘ Neural Fuzzy Systems ’’ , Prentice-Hall Inc., New Jersey, 1995.

[10] J.H. Holland, ‘‘Outline for a logical theory of adaptive systems.’’, J. Assoc. Computing. Mach. 3:297-314,

[11] J.H. Holland, ‘‘Adaptation in neural and artificial systems’’, Ann Arbor, MI: University of the Michigan Press, 1975

[12] L. Davis,‘‘Handbook of Genetic Algorithms’’ New York:Van Nostrand, Reinhold,1991.

LIST OF FIGURES

Figure 1. General CNN Neighbourhood Structure.

Figure 2. Representation of Neighbourhood Relation of CNN.

Figure 3. CNN Model Input-Output Relationship

Figure 4. Piece-wise Linear Output Characteristics of CNN cell.

Figure 5. Input Image used during genetic CNN training

Figure 6. Target Image used for genetic CNN training

Figure 7. Examples of genetic CNN application: Input (a), (c),(e), (g),(i),(k);Genetic CNN CNN Output (b),(d),(f),(h),(j),(l)

FIGURES

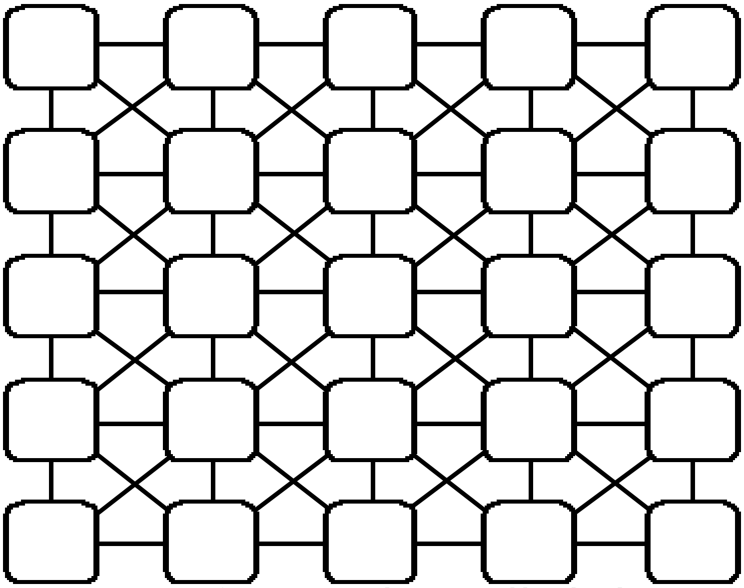
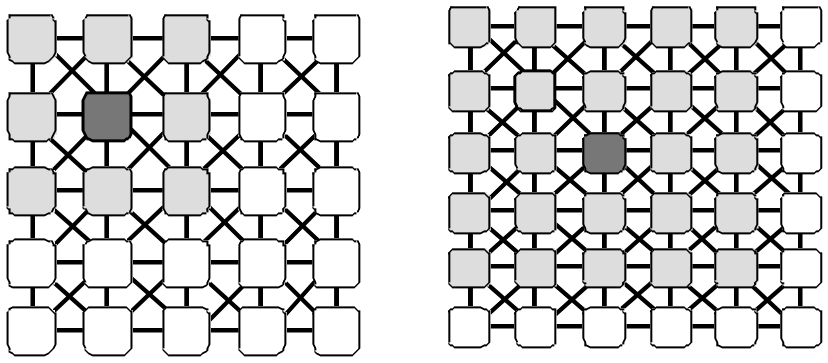


Figure 1. General CNN Neighbourhood Structure.



r=1 r=2

Figure 2. Representation of Neighbourhood Relation of CNN.

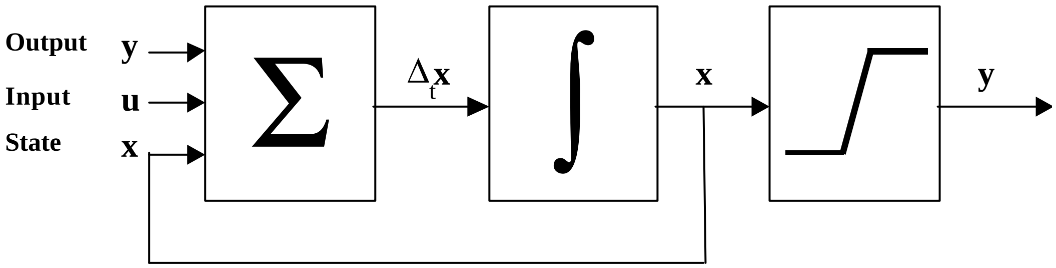


Figure 3. CNN Model Input-Output Relationship

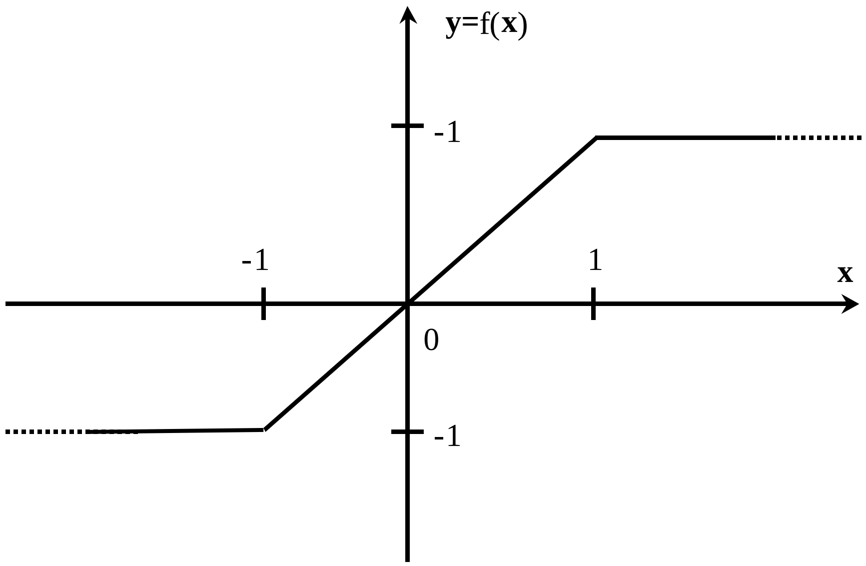


Figure 4. Piece-wise Linear Output Characteristics of CNN cell.

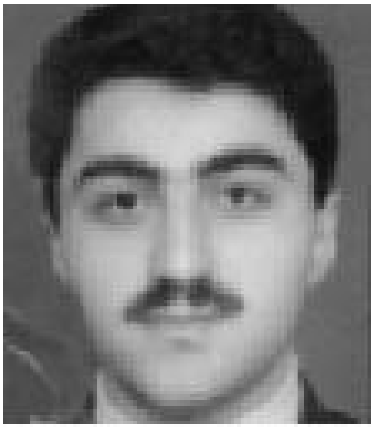
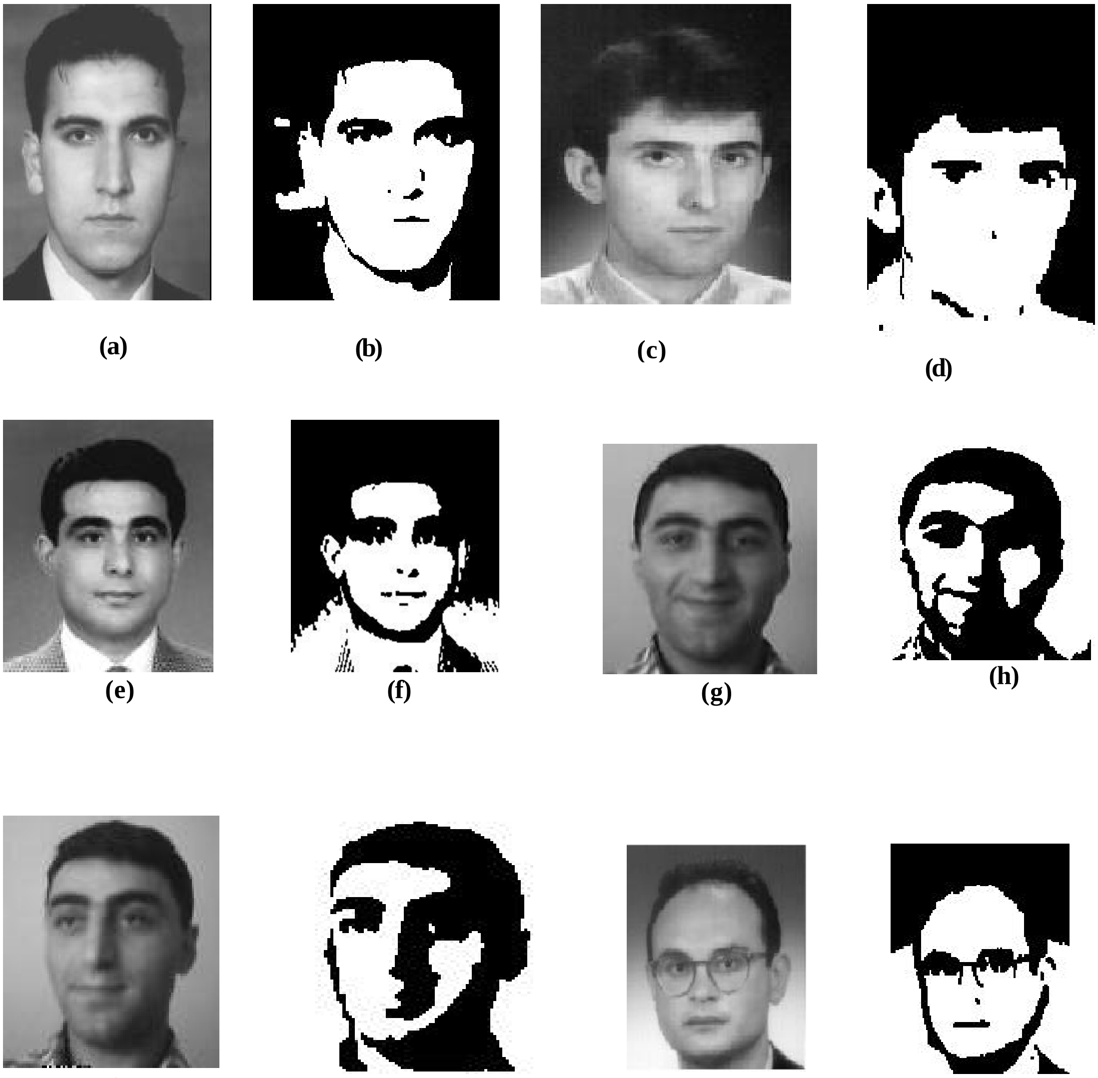


Figure 5. Input Image used during genetic CNN training

Figure 6. Target Image used during genetic CNN training



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** **(l)**

Figure 7. Examples of Genetic CNN application: Input data (a), (c),(e), (g),(i),(k); Genetic CNN Output data (b),(d),(f),(h),(j),(l)