# Face Localization Using Backpropagation

**Abdul Bassut Kadhim shuker**

**Diyala University\College of Science\ Computer Science Department**

## Abstract

There is more than one type of neural net uses to classify complex pattern (face), and it is represent very suitable technique for face localization. We present backpropagation neural network-based face localization system. Face localization in image is a two class (face, nonface) and it is represent very an important first step for many systems ranging from supervision to human computer interface because of the computational model of face. The proposed system consist of two subsystem ,first subsystem is preprocessing operations on image to make it smooth and make next operation (training patterns in backpropagation neural net )easy as possible .Second subsystem is classifying the patterns produced by the preprocessing stage consists showing the patterns(an image) to backpropagation neural net to localize the face in an image .

## الخلاصة

عملية تحديد الوجه هي الخطوة الاولى والمهمة في منظومة متكاملة لتتميز الوجه بالإضافة إلى مجوعة الانظمة الهامة التي تعتمد عليها ،مثل انظمة المراقبة والتعداد الى وجهة بين المستخدم والحاسب .كذلك يعتبر من المنظومات الغير قابلة للتطفل لدرجة كبيرة .بسبب اسلوب البناء المتوازي وكفاءة العمليات الحسابية للشبكات العصبية الاصطناعية فقد اصبح استخدامها في معالجة الصور شائعا جدا ،تم في النظام المقترح تصميم نظام متكامل لتخصيص الوجه في الصورة ، يعتمد النظام المقترح على مجموعة عمليات معالجة مسبقة ،حيث تعتبر المشكلة الاكبر في عملية التدريب في الشبكات العصبية هي هل تحتوي المتجهات الناتجة من عمليات المعالجة المسبقة على معلومات كافية تمكن الشبكة من القدرة على التدريب على نماذج صور الوجه بصورة ناجحة . تم استخدام اكثر من مرشح والمقارنة بين هذه المرشحات لغرض الحصول المرشح الذي يجعل الصورة تحتوي على اكبر قدر من المعلومات يتم ادخالها الى الشبكة العصبية لغرض التدريب الناجح .بعد الحصول على معلومات من الصورة (خاصة المنطقة التي تحتوي العينيين والحاجبين والفم ) يتم ادخال المعلومات الى شبكة عصبية كفؤة لغرض التمييز.

**Key Word:** Face localization, Face localization, Face recognition, Expression Recognition,

## 1-1Introduction

This introduction describes the most important information that are considered as background for this paper.

## 1-2 Problem analysis

Face processing was not possible until 10 years ago because of the existing technology. Nowadays, there are several algorithmically techniques allowing face processing, but under several restrictions.[1] Defining these restrictions in a given environment is mandatory before starting the application development. Face detection problem consists of detecting presence or absence of face-like regions in a static image (ideally, regardless its size, position, expression, orientation and light condition) and their localizations. This definition agrees the ones in [4] [5]. Allowing image processing and face localization in a finite and short amount of time requires the image fulfills next conditions:

1. **Fixed size images**: The images have to be fixed in size. This requirement can be achieved by image preprocessing, but not always. If the input image has lower size than required, the magnification is inaccurate.

2. **Constant ratio faces**: The faces must be natural faces, around the correct proportions of an average face.

3. **Pose:** There are face poses techniques to find a rotated face in an image, it is achieved by harvesting face rotation invariant features. However, the neural network approach adopted in this document uses only simple features. This implies limited in-plane rotation (the faces must be looking at the direction normal to the picture at most)

4. **Distance:** The faces must be at such a distance that its size allows detection. It means about 20x20 pixels faces.

The output of face localization process is a set of normalized faces. The format of the normalized faces could be face images, positions of the faces in the original image, The main problems in face detection are related with the following factors:

1. **Face position**: Face localization is affected by rotation (in-plane and out-of-plane) and distance (scaled faces).

2. **Face expression**: There are facial expressions that modify the face form affecting the localization process.

3. Structural components: Moustaches, barb, glasses, hairstyle and other complements difficult the process.

4. **Environment conditions**: Light conditions, fog and other environmental factors affect dramatically the process if it is mostly based on skin color detection.

5. **Occultation**: Faces hidden by objects or partially out of the image represent a handicap in the process.

There are four approaches for the face localization problem [3,5]:

1. **Knowledge-based methods**: It uses rules based on human knowledge of what is a typical face to capture relations between facial features.

2. **Feature invariant approaches:** It uses structural invariants features of the **faces.**

**3. Template matching methods**: It uses a database of templates selected by experts of typical face features (nose, eyes, mouth) and compare them with parts of the image to find a face.

**4. Appearance-based methods**: It uses some selector algorithm trained to learn face templates from a data training set. Some of the trained classifiers are: neural network[1],The first and second approaches are used to face localization. Third works in localization and detection. And fourth is used mainly in detection.

## 2. Proposed system

The details of proposed system can be shown in figure (2.1) it fist applies a set of filters, because the bigger problem in the training process is to know if grayscale image contains enough information by themselves to allow classifier training successfully. the first component of our system that receives as input a 20*20 pixel region of the image ,and generate an output (0 or 1 ).First two filter represents smoothing filters ,while the other edge detection filters.[6]

1- Mean filter.
2- Median filter.
3- Sobel filter.
4- Prewitt filter.

5- Canny filter.

6- Log filter.

These preprocessing operations includes (smoothing operation ,edge localization and resize image) are required for making an image easy to the next level( training neural on face patterns ).These filter operations on the original images are part of the preprocessing step in the whole face localization system.

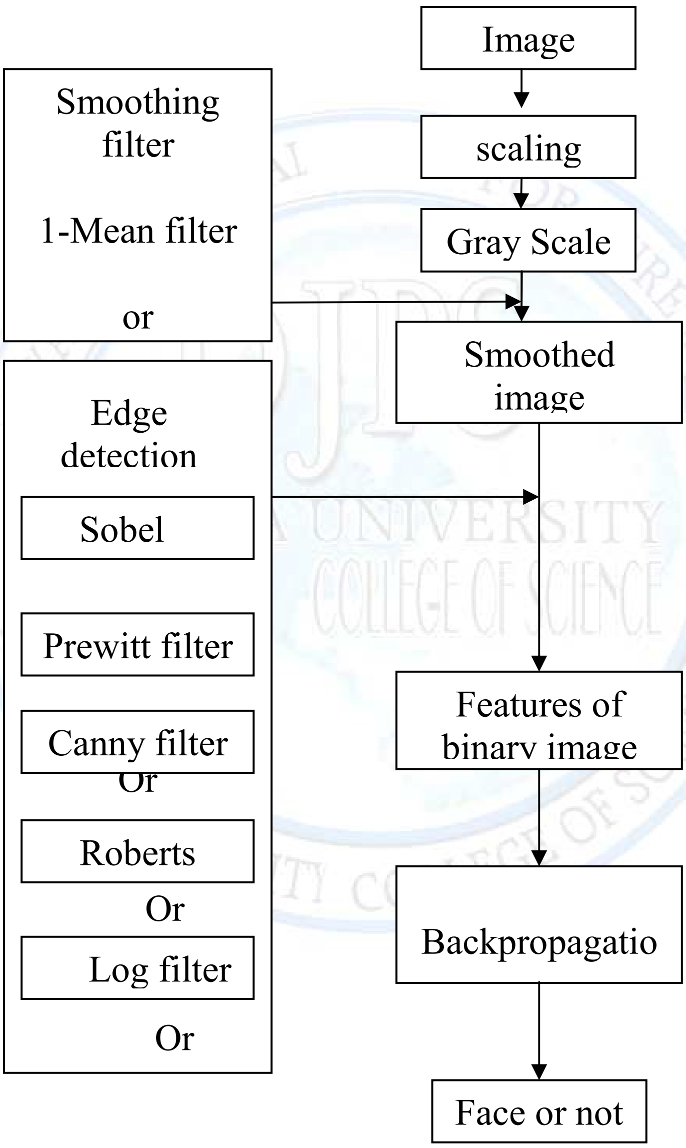



**Figure (2.1): The proposed system operations (preprocessing, neural net)**

## 2.1 Reading Image

To process an image by digital computer, an image function f(u,v) must be digitized both spatially and in amplitude. Digitization of the spatial coordinate (u,v) will be referred to as image sampling while amplitude digitization will be called gray –level quantization [ 7 ] .Each pixel represent brightness of the image at any point in an image .We can use digital camera or scanner to convert photograph an image to an array .An array contains data of an image .to read data of an image :

TO READ AN IMAGE
Image = I(r,c);
r= image width
c=image height
1-Read the a image width
2-Read the in image height
For the an image width
For the an image height
Read an image
3-End

## 2.2 Scaling

When we deal with an images with different sizes, it is very important to good training resize each pattern (face image) to fixed size. In our proposed system we scaling an face image with different size to [64, 64] an image size.

TO RESIZE AN IMAGE
1- Read an image
2- Read an image width
if image width >64
make image width =64 and
if image width < 64
make image width = 64

3- Read an image height
   if image height >64
   make image height =64 and
   if image height < 64
   make image height = 64
4- Save the results
5- End

## 2.3 Filtering

Smoothing filters are used to make the face reliable in an image, it is used for blurring and for noise reduction .blurring is used in preprocessing steps such as removal of small details from an image prior to large object extraction ,and brighing of small gaps in lines or curve . Neural network training can be made more efficient if you perform certain preprocessing steps on the training pattrens. This section describes several preprocessing filters that you can use it. [8 ] ,there two filters used to smooth an image .

## Median filter

The gray level of each pixel is replaced by the median of the gray levels in a neighborhood of that pixel. This method is particularly effective when the noise pattern consists of strong characteristics to be preserved in edge sharpness. It is set of values such that half of values less than certain value and the other half values more than that value. Median filtering is a nonlinear operation often used in image processing to reduce noise. A median filter is more effective than convolution when the goal is to simultaneously reduce noise and preserve edges.

## Mean filter

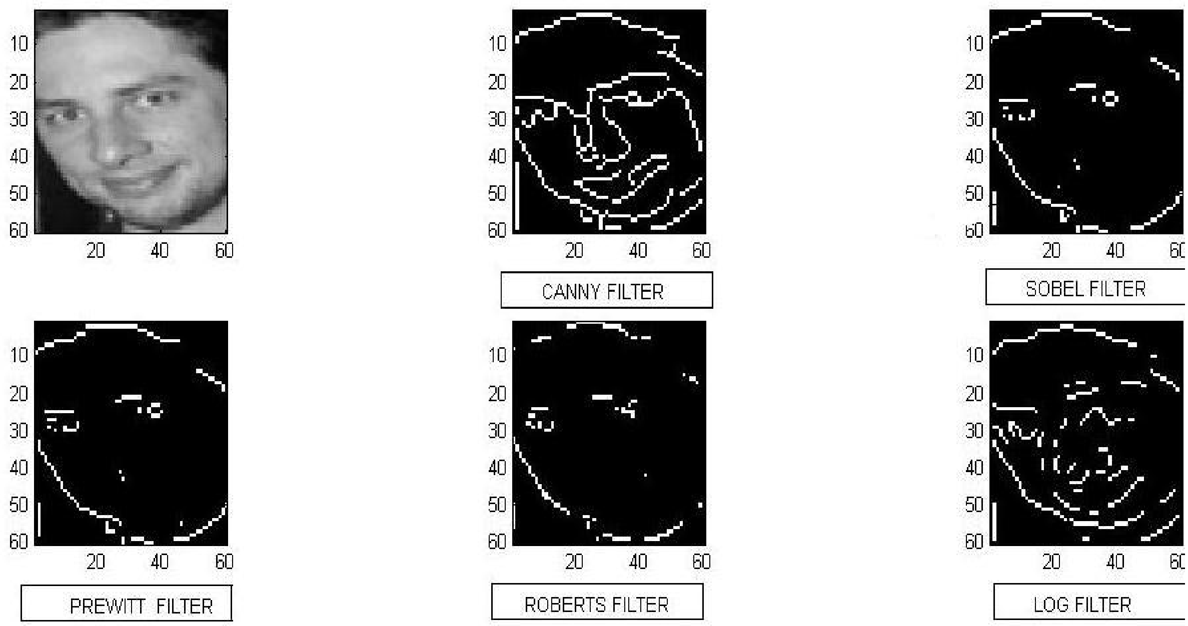
A mean filter is average of image pixels (by using mask(3*3 window at over all image pixels and put the average of mask in the center of mask a(i,j),were:

mean =1/9

| | | |
|---|---|---|
| i-1,j-1 | i-1,j | i-1,j+1 |
| i,j-1 | i,j | i,j+1 |
| i+1,,j-1 | i+1,j | i+1,j+1 |

## Edge Detection Filter

Edge takes intensity or a binary image I as its input, and returns binary image of the same size as I, with 1's where the function finds edges in I and 0's elsewhere. More then one filter used in proposed system to find edges in intensity image .The Sobel, prewitt and Roberts's fitter's method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of intensity is maximum. [6]. Sobel,Roborts and Prewitt methods specifies the sensitivity threshold and ignores all edges that are not stronger than threshold .The Laplacian of Gaussian method(log) finds edges by looking for zero crossings after filtering I with a Laplacian of Gaussian filter. The Canny method finds edges by looking for local maxima of the gradient of I. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore less likely than the others to be "fooled" by noise, and more likely to detect true weak edges (see figure). Canny filter specifies sensitivity thresholds for the canny method. Threshold is a two-element vector in which the first element is the low threshold, and the second element is the high threshold. If you specify a scalar for threshold, this value is used for the high threshold and 0.4*threshold is used for the low threshold.



all filtered image (canny filtered image , Sobel filterd image, prewitt filtered image, robots filtered image and log filtered image) are treated by other filter . This operation (all filtered image treated with one filter) can be used to recognize among filtered image, that's lead to choose most information filtered image. In our proposed system the canny filtered image have been chosen, it is clearly canny filtered image has more information in the regions of face than other (as illustrated in the figure above).In all filtered image we need deploring operations to delete noise and bridging the lines and curves in the regions of face except canny filtered image.

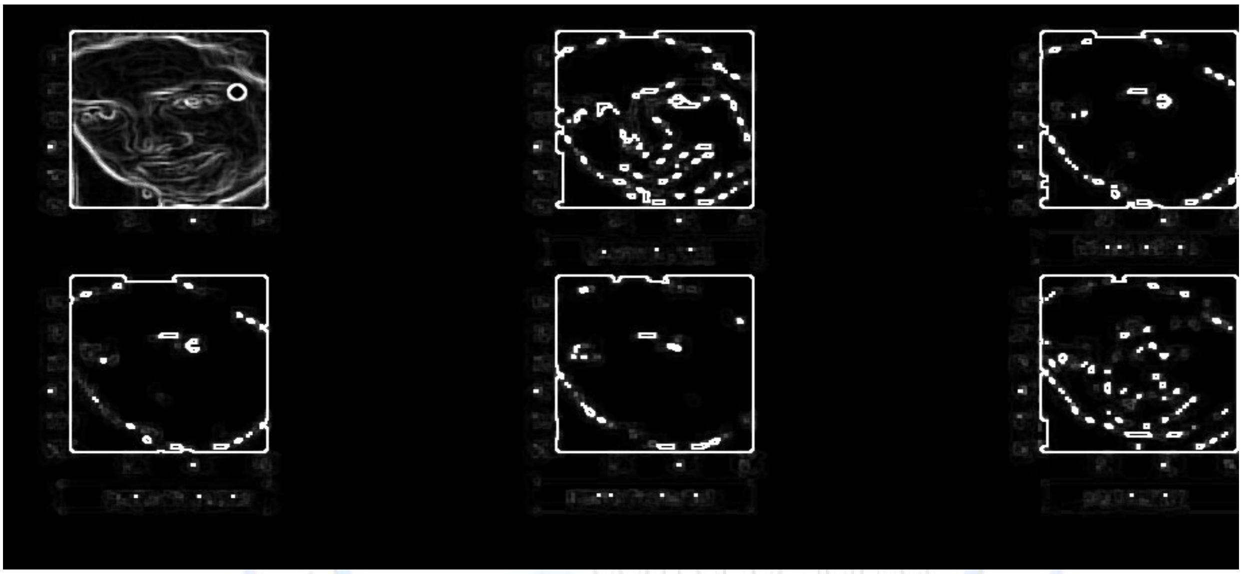


**Figure (2-2):(original image in the upper left corner ,canny filtered image on the right ,sobel filtered image in the right upper corner , prewiit filtered image in the lower left corner ,robrts filtered image on the right and log filtered image in the lower right corner )**

## Final Stage (Backpropagation Neural Stage)

Neural networks are powerful tools that can be trained to perform a complex and various functions in computer vision applications, such as preprocessing (boundary extraction, image restoration, image filtering), feature extraction (extract transformed domain features), associative memory (storing and retrieving information), and pattern recognition [9]. All the previous operations in the preprocessing stage is to make the image face easy to the training patterns in the backpropagation neural net. Every stage receives, as its input, the output data from previous stage .Classifying the patterns produced by the preprocessing stage consist of showing the patterns to the backpropagation neural net and inspecting it output. The output neuron show the certainty that the pattern face region in image.

## Algorithm of Classification

1. From 1 to the number of patterns do the fallowing steps
2. Divided input filtered image to regions, such that

Z= y01,y02,y03,y04
y05,y06,y07,y08
y09,y10,y11,y12
y13,y14,y15,y16

where:

y01 :represents first vector consist of (1……256] values.

y2 : represents second vector consist of (1……256] values. and so on.

3. Divided each region of pattern to vectors, such that each vector:

Y01= x1*1,x1*2,x1*3,………. …… ….x1*16
X2*1,x2*2,x2*3………………….x2*16
X3*1,x3*1,x3*2………………….x3*16
X4*1,x4*1,x4*2………………….x4*16
X5*1,x5*1,x5*2………………….x5*16
|
|
X16*1,16*2,16*3……………….x16*16

For 1 to 16 :

Training the vectors by the backpropagation neural net;

Get the output of the 16 neural net:

4.Input the output of 16 neural to last neural ; then do the fallowing steps

4.1 Apply the input vector, Xp=(Xp1 , Xp2 , …….. , Xp16 )24 to the input units .

4.2 Calculate the net- input values to the hidden layer units:

neth pj=(Σ Wh ji Xpi ) (A.1)

4.3 Calculate the outputs from the hidden layer:

$$i_{pj} = f_{hj}(net_{h} \cdot pj) \quad (A.2)$$

4.4 Move to the output layer. Calculate the net-input values to each unit:

$$neto_{pk} = (\Sigma WO_{kj} \cdot i_{pj}) \quad (A.3)$$

4.5 Calculate the outputs:

$$O_{pk} = f_{oj}(neto_{pk}) \quad (A.4)$$

4.6 Calculate the error terms for the output units:

$$\delta o_{pk} = (Y_{pk} - O_{pk}) \cdot fo'_{k}(neto_{pk}) \quad (A.5)$$

Where,

$$fo'_{k}(neto_{pk}) = fo_{k}(neto_{pk}) \cdot (1 - fo_{k}(neto_{pk})) \quad (A.6)$$

4.7 Calculate the error terms for the hidden units

$$\delta h_{pj} = f_{kj}(neth_{pi} \cdot \Sigma \delta o_{pk} \cdot Wo_{kj}) \quad (A.7)$$

4.8 Update weights on the output layer

$$Wo_{kj}(t+1) = Wo_{kj}(t) + (\eta \cdot \delta o_{pk} \cdot i_{pj}) \quad (A.8)$$

4.9 Update weights on the Hidden layer

$$Wh_{ji}(t+1) = Wh_{ji}(t) + (\eta \cdot \delta h_{pj} \cdot X_{i}) \quad (A.9)$$

4.10 Get the output and save the result.

## 3 Results

It can be supposed that the union dataset of face-like and no face-like patterns is a non-linearly separable set, so a non-linear discriminator function should be used. Artificial neural networks in general, and a multilayer feed forward perceptron with back propagation learning rule in particular fit this role.The classifier training process consists of a supervised training. The patterns and the desired output for each pattern are showed to the classifier sequentially. It processes the input pattern and produces an output. If the output is not equal to the desired one, the internal weights that contributed negatively to the output are changed by the back propagation learning rule; it is based on a partial derivates equation where each weight is changed proportionally to its weight in the final output. In this way, the classifier can adapt it neural connections to improve its accuracy from the initial state (random weights) to a final state. In this final state the classifier should be able to produce correct (or almost correct) outputs. The network performance is measured by the Mean Squared Error (MSE). MSE is the sum of the squared absolute values of the difference between network outputs and desired outputs.

$$MSE_t = \frac{1}{patterns} \left( \sum_{i=1}^{patterns} \sum_{i=1}^{patterns} \right) (n_{ki} - d_{ki})^2$$

k is the pattern number and goes from 1 to the number of patterns (patterns in the formulae); i is the number of the output neuron; n is the computed output and d is the desired output.

This section summarizes the results:

1- The tables

- (3-1,3-2) represents the values of face vector.
- (3-3) represents the values of non face vector. 3-experiment shows the ability of the backpropagation to localize:
- Face (rotated to left).
- Frontal face.
- Ellipse face.
- Rotated face to right.
- Corrupted face.

**Table (3-1): Shows Backpropagation Discriminate Values for (Run 1-8)**

| Output of the training set (faces ) | | | | | | | |
|---|---|---|---|---|---|---|---|
| 9.1082 | 5.1778 | 6.8765 | 8.6766 | 5.9932- | 4.0199 | 0.8370 | 2.0759 |
| 7.4338- | 3.3666 | 5.3436- | 1.4581- | 2.4432- | 1.5558- | 2.9670 | 4.6349 |
| 5.1986- | 1.4159 | 5.9710- | 1.8645 | 1.4990- | 8.9547- | 2.2653 | 1.3254 |
| 2.6946 | 0.8107 | 0.5397- | 2.7092 | 0.9876- | 0.6317 | 0.1495 | 3.2881 |
| 5.3280- | 1.3646- | 6.2553- | 3.7234 | 4.3994 | 1.6844- | 7.9960 | 5.6919 |
| 4.9281- | 2.0647 | 1.5376- | 1.2055- | 0.0172 | 0.6175 | 9.5390 | 3.8593- |
| 3.0892- | 9.7508 | 7.4582 | 3.8267- | 3.5235- | 0.7369 | 3.2998 | 1.5593 |
| 5.3130 | 0.0502- | 0.6253 | 6.6213- | 4.2588- | 1.3483- | 2.6895- | 0.8745 |
| 4.5787- | 3.6585- | 0.1086 | 6.4493- | 3.6336 | 2.5496- | 5.4201- | 4.3798 |
| 7.1256 | 3.2137 | 5.2360 | 6.3741 | 5.6986- | 2.9081 | 5.2129 | 3.9076- |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4.7894- | 5.2889- | 0.8292 | 6.8601 | 4.3214- | 0.3608 | 2.1000- | 5.9666 |
| 0.6460- | 0.0961- | 4.2058- | 5.7518 | 3.9482 | 2.2941 | 2.6876- | 0.1827- |
| 6.1083 | 4.1646 | 7.8713 | 9.0274 | 1.5645 | 2.6083 | 3.2571 | 0.5248 |
| 1.9365- | 7.9680 | 0.0720- | 3.3207- | 5.7292 | 7.5132- | 5.6454- | 2.1550- |
| 8.0129 | 9.9501 | 1.0571- | 2.0206 | 3.3940- | 2.4844 | 7.3423 | 1.3386- |
| 9.1355 | 1.7169 | 3.2534 | 7.6873 | 3.9649 | 3.7712- | 0.3554 | 7.1295- |
| 0.5665- | 7.4399- | 5.4512- | 1.2123 | 1.4716 | 2.3659 | 3.5190 | 0.6467 |
| 2.8900 | 2.1273 | 5.2755- | 1.0809 | 4.9709- | 3.2071- | 2.6470- | 1.1033 |

**Table (3-2) Shows Backpropagation Discriminate Values for (Run 9-16)**

| **Output of the training set (faces )** | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7.2220- | 3.6371- | 4.6505- | 0.6502- | 3.2044- | 1.6299- | 1968.- | 6.8945- |
| 0.0388- | 0.6306- | 7.0338- | 7.1430 | 2.4503 | 0.6056- | 7.0143 | 3.9210 |
| 5.0222- | 4.6661 | 4.3383 | 6.4372 | 0.8532- | 1.6864 | 1.4375 | 1.0411- |
| 5.4577- | 0.4140 | 5.0595- | 0.2114- | 5.6995- | 3.2700- | 1.7241- | 1.8994- |
| 2.2693- | 4.4206- | 0.0823- | 1.7415 | 4.2918- | 0.1655 | 14.8879- | 0.3584 |
| 0.2704 | 0.1139 | 6.3374- | 6.1980- | 0.1330- | 3.3332- | 1.3267 | 1.5600- |
| 3.8422- | 8.7518- | 0.3489- | 2.0846- | 3.9721- | 3.6808- | 4.6817 | 11.5318 |
| 0.9306- | 2.2307 | 0.4650 | 6.2853- | 3.5938- | 2.3792- | 1.9101 | 4.3135- |
| 1.6662 | 3.2408 | 1.1355 | 7.1960- | 4.5624 | 3.3112- | 0.1739- | 3.7628 |
| 3.2351 | 4.7162- | 1.5281 | 1.3249 | 5.2452- | 7.7218 | 5.4058- | 3.4962- |
| 2.2949 | 6.0218- | 4.4279 | 4.7990 | 4.9641 | 1.8644 | 1.7719 | 7.2207 |
| 3.3418- | 3.1312- | 0.5088- | 0.9876- | 7.7125 | 1.6609 | 3.2499 | 4.2935 |
| 0.7075 | 1.5621 | 1.9664 | 4.7086- | 2.7549 | 2.8419 | 0.8857 | 7.8639 |
| 7.6771- | 4.8074 | 5.7631- | 0.7458- | 1.7809- | 2.4028- | 6.4725- | 8.9445- |
| 0.8338- | 2.2558 | 5.8730- | 3.9192- | 11.1699- | 4.3069- | 0.4739 | 6.8454- |
| 0.2668- | 3.6799 | 0.0762- | 3.9143 | 12.8335- | 6.6248- | 5.9838- | 6.5941- |
| 1.0997 | 1.3669- | 4.3614 | 5.9289- | 4.8665 | 4.8351 | 1.7450 | 0.9925 |
| 7.3518- | 1.1626- | 7.9469 | 4.0538 | 1.7818 | 3.9704- | 6.0053- | 1.8378 |

**Table (3-3)Shows Backpropagation Discriminate Values for (Run 9-16)**

| Output of the training set (non-faces ) | | | | | |
|---|---|---|---|---|---|
| 7.4542 | -3.5599 | -4.9223 | 4.4798 | 8.6173 | 4.9943 |
| 4.4835 | 7.7375 | -1.4135 | -2.5981 | -3.3930 | 4.8136 |
| 6.2480 | 3.3400 | -1.2879 | 4.9482 | -3.8225 | 3.6476 |
| -5.3025 | -0.2041 | -3.1338 | -4.3903 | -0.8226 | -5.4382 |
| 6.2220 | 5.1152 | 4.5892 | -0.1106 | 0.3793 | 7.3384 |
| -3.5043 | -6.7863 | -4.5754 | -6.8516 | 0.2095 | 2.2131 |
| -1.3627 | 0.0621 | -0.5090 | 4.1705 | 0.5370 | -4.9176 |
| 1.3543 | 3.2827 | -3.9222 | 6.0012 | -6.3833 | 1.1700 |
| 2.2844 | 6.4575 | 5.9172 | 0.5446 | -4.0681 | -10.4199 |
| -6.5918 | 7.6932 | -0.1348 | 1.8987 | -4.8762 | 3.3048 |
| 5.6691 | 5.3939 | 7.6555 | -1.5871 | 3.9891 | 3.7102 |
| 0.3031 | -3.1702 | 7.9176 | 4.6972 | 0.1565 | 6.7903 |
| 2.0917 | 4.6827 | -0.4889 | -1.9104 | -1.2294 | 2.5211 |
| -5.6461 | -9.6464 | -0.0473 | -7.6679 | -4.4836 | -6.2172 |
| -5.6765 | -4.7624 | -1.1324 | -7.0013 | 3.4544 | -0.4621 |
| -2.1347 | -5.2260 | -3.3698 | -2.0654 | 5.4498 | -2.8707 |
| 0.8478 | -0.7669 | -0.2821 | 2.2388 | 0.9194 | -7.6551 |
| -2.4566 | 2.8069 | -1.6701 | 6.3500 | -4.8800 | 0.2673 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |
| -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 | -1.0000 |

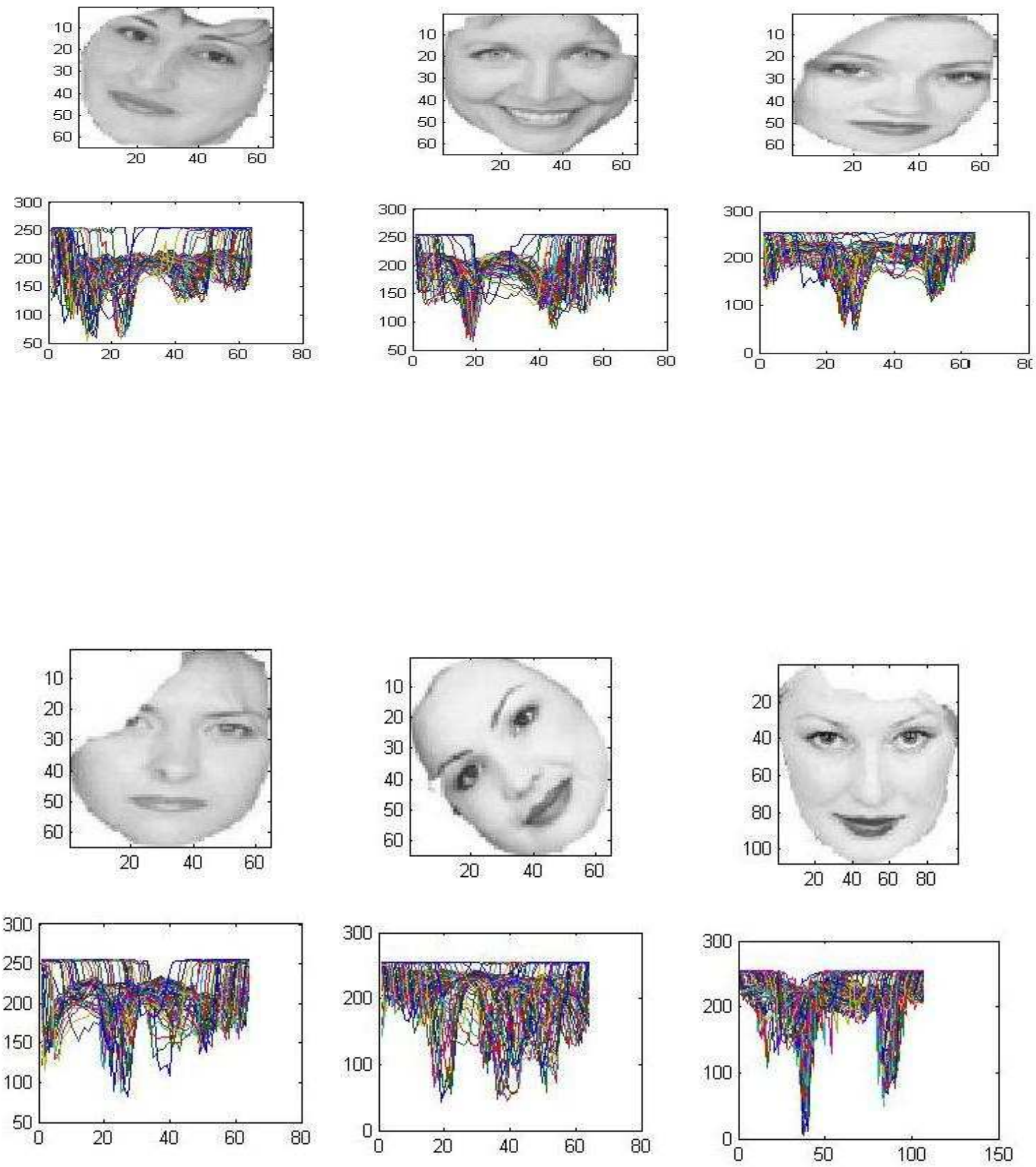


Figure (3.1a): Upper level The figure represents patterns in different poses of face: upper left corner image represent face semi frontal, in the upper middle the frontal face and upper right face is rotate face Lower level The of data to Semi frontal ,rotate left and corrupted face

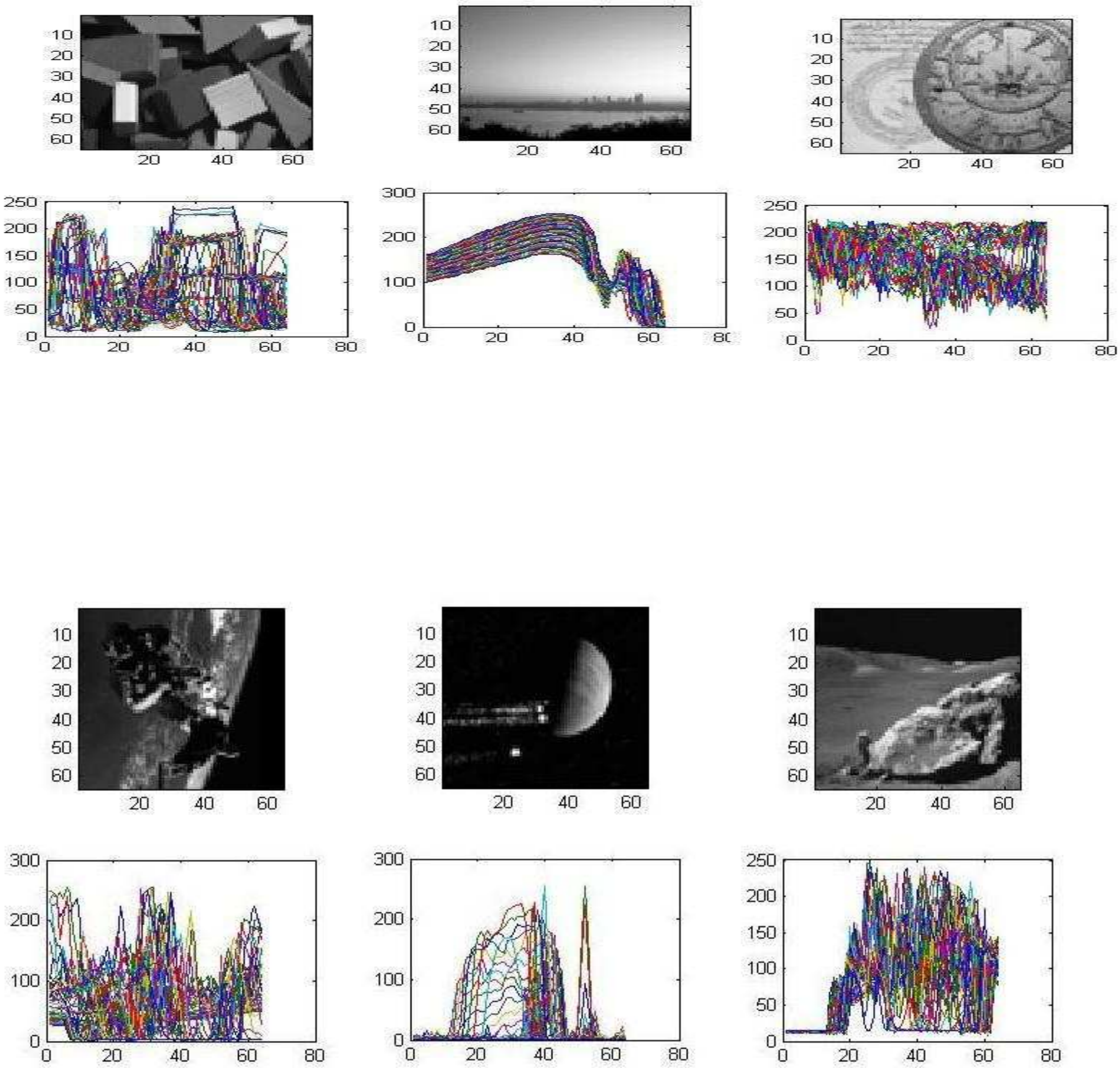


**Figure (3.1b): The figure Represents Patterns it Different Shapes of Non-face**

## Conclusions

1-The training method used was backpropagation. The problem here was to find a proper learning rate.

a-If set it too big(learning rate), the network never learned,

b- If it too small, the training took too long.

2-Main problem of the components –based neural approach is how to choose the set of discriminatory object parts

3-Possible improvements might be to increase the number of training examples.

4-Taking advantage of color information if applied on color images.

5-Although small values for the learning rate were used, the BP was able to train the neural

6- Network with much fewer numbers of epoch in BP

7-Canny filter can be represented as more efficient filter , because we get almost information in all regions of an image (face) after we apply the filter on an image ,when we lock at filtered an image treated with canny filter we see the components of face (eyes, nose ,moth ,eye borrow) appeases clearly.

## Reference

[1] David Suárez Perera1.Backpropagation neural network based face detection infrontal faces images .Neural & Adaptative Computation + Computational Neuroscience Research Lab Dept. of Computer Science & Systems, Institute for Cybernetic University of Las Palmas de Gran Canaria.Las Palmas de Gran Canaria, 35307, Spain, University of Applied Sciences,Brandenburg an der Havel, 14770.

[2] Rule or knearest neighbor based. David Suarez 3 21/09/2005

[3] Ming-Hsuan Yang. (2004). Recent Advances in Face Detection. IEEE ICPR 2004 Tutorial, Cambridge, United Kingdom, August 22, 2004.14.09.05. http://vision.ai.uiuc.edu/mhyang/papers/icpr04_tutorial.pdf

[4] Overview of face detection algorithms (proposal). Tao Wang,CMPUT 615.14.09.05. http://www.ee.ualberta.ca/~xiang/proposal2

[5] Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja.(2002). Detecting Faces in Images: A Survey.IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 24, No. 1, January 2002.((14.09.05. http://vision.ai.uiuc.edu/mhyang/papers/pami02a.pdf))

[6] Matlab help

[7] Scott, E Umbaugh, Ph.D. "Computer Vision and Image processing",Prentice Hall PTR.((http://www.prenhall.com)).

[8] Rafael C.Gonzalez ," digital image preprocessing ",1992.

[9] Abdual Radha Hanash Abbas " Face Detection, Localization and Recognition", Ms. thesis submitted to Baghdad University, College of Science, Computer Department, 2000.