A PROJECT REPORT ON

ARTIFICIAL NEURAL NETWORK BASED TEXTURE ANALYSIS

SUBMITTED BY

AMITKUMAR MANTRI	B3023043
NIRMAL VIJAYAN	B3023048
YOGENDRA PATIL	B3023058

PROJECT GUIDE

PROF. ALWIN D. ANUSE

DEPARTMENT OF
ELECTRONICS & TELECOMMUNICATION

MAHARASHTRA INSTITUTE OF TECHNOLOGY
PUNE - 411033

2006 - 07

B.E. Project Report

On

ARTIFICIAL NEURAL NETWORK BASED TEXTURE ANALYSIS

Submitted by,
Amitkumar Mantri(B3023043)
Nirmal Vijayan (B3023048)
Yogendra Patil(B3023058)

Project Guide
Prof. Alwin D. Anuse

Year: 2006-2007

Department of Electronics and Telecommunication,

Maharashtra Institute of Technology, Pune – 38.

CERTIFICATE

MAEER's

MAHARASHTRA INSTITUTE OF TECHNOLOGY, PUNE.

This is to certify that the project entitled

Artificial Neural Network Based Texture Analysis

has been carried out successfully by

Amitkumar Mantri (B3023043)
Nirmal Vijayan (B3023048)
Yogendra Patil (B3023058)

during the Academic Year **2006-2007** in partial fulfillment of their course of study for Bachelor's Degree in ***Electronics and Telecommunication*** as per the syllabus prescribed by the **University of Pune.**

G. N. Mulay

Prof. Alwin D. Anuse
Internal Guide

Prof. Dr. G. N. Mulay
Head Of Department
(Electronics & Telecommunication)
MIT, Pune

ACKNOWLEDGEMENT

It gives us great pleasure in expressing our deepest gratitude towards all those who have helped us to materialize this project and for providing us with never-ending motivation.

First of all, we are very thankful to our project guide **Prof. Alwin D. Anuse** for reviewing our work, suggesting corrections and for being guiding light to us. His valuable guidance has helped us at all stages of our work. Without his novel ideas and concepts this project would not have been possible.

We would like to thank **Prof. Vibha Vyas, COEP, Pune** for her guidance in Image Processing. She helped us clear our doubts and gave us ideas on how to approach the tasks regarding the project. We also express our deep gratitude to **Mr Vijay Kumar, AFMC, Pune** for giving us valuable suggestions regarding various aspects of project.

However, last but by no stretch of imagination the least, we would like to thank our honorable **HOD, Dr. Prof. G.N.Mulay**, for creating an environment of technical excellence in MIT PUNE, which has helped us and many others to prosper and bloom through his encouragement and support in this most important endeavor of our engineering lives.

INDEX

LIST OF FIGURES

LIST OF TABLES

Chapter 1

OVERVIEW OF PROJECT

CHAPTER 1

OVERVIEW OF THE PROJECT

1.1. INTRODUCTION:

Texture provides a rich source of information about the natural scene. It is an important component in image analysis for solving a wide range of applied recognition, segmentation and synthesis problems, but also it provides a key to understanding basic mechanisms that underlie human visual perception. Various textures are analyzed by applying various methods like co-occurrence matrix, entropy based method, etc. to extract features such as entropy, energy, contrast, correlation, etc. and are further used as training samples to the Artificial Neural Network which then classifies them appropriately to the corresponding classes.

Artificial Neural Network is a set of processing units which when assembled in a closely interconnected network offers a structure exhibiting some features and attributes of the biological neural network. There are various ANN algorithms depending upon the learning method used. E.g. Back propagation, K-means clustering, etc. This project deals with back propagation and k-means clustering algorithms as the learning methods for the ANN.

Such an entire structure has many applications in the branches of industrial and biomedical activities such as MR brain imaging, road surface skidding estimation, work piece surface monitoring, detection of masses in mammograms and diseases in plants.

1.2. SCOPE OF THE PROJECT:

The project deals with implementing:

1. *Classification of Basic Textures.*

- This involves classifying textures into 4 classes viz. fabric, marble, leaves and pcbs.

2. *Detection of Masses in Mammograms.*
3. *Classification of Wood Surfaces (Real Time Application).*

All the above are based on a software structure implemented in Matlab under Windows platform. In the implementation of this system we presume images are coloured images with depth 24bit. The classification is done with only 4 features out of the many available. For simple classification approach the neural network is trained first with the training samples and then tested with features extracted from the test image.

Chapter 2

LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1. TEXTURE

The word texture refers to the surface characteristics and appearance of an object given by the size, shape, density, arrangement, proportion of its elementary parts. A texture is usually described as smooth or rough, soft or hard, coarse or fine, matt or glossy, and etc.

Textures might be divided into two categories, namely, *tactile* and *visual* textures. Tactile textures refer to the immediate tangible feel of a surface. Visual textures refer to the visual impression that textures produce to human observer, which are related to local spatial variations of simple stimuli like colour, orientation and intensity in an image. This report focuses only on visual textures, so the term `texture' thereafter is exclusively referred to `visual texture' unless mentioned otherwise.

Figure 2.1.(a) and 2.1.(b) shows a few natural and man-made textures, respectively, which could be met in daily life.

Although texture is an important research area in computer vision, there is no precise definition of the notion texture. The main reason is that natural textures often display different yet contradicting properties, such as regularity versus randomness, uniformity versus distortion, which can hardly be described in a unified manner.

Texture is considered as an "organized area phenomenon" which can be decomposed into 'primitives' having specific spatial distributions. This definition, also known as *structural approach*, comes directly from human visual experience of textures. For instance, each texture in Figs 2.1.(a) and 2.1.(b) is composed of particular texture elements, e.g., objects (windows), shapes (jigsaw pieces), or simply colour patterns. Meanwhile, these primitives are organized in a particular spatial structure indicating certain underlying placement rules. The other feature that describes a texture is the tone of the primitive. Tone is nothing but the pixel intensity properties of the primitive.

Figure 2.1.(a): Examples of natural textures.

Figure 2.1.(b): Examples of artificial regular textures.

Figure 2.1.(c): Examples of natural regular textures.

By spatial homogeneity, textures can be classified into *homogeneous*, *weakly-homogeneous*, and *inhomogeneous* patterns. Specifically, homogeneous texture contains ideal repetitive structures, and such uniformity produces idealized patterns. Weak homogeneity involves local spatial variation in texture elements or their spatial arrangement, which leads to more or less violates the precise repetitiveness (See Fig 2.1.(d)). An `inhomogeneous texture' mostly refers to an image where repetition and spatial self-similarity are absent. Since spatial homogeneity is considered below as an essential property of a texture, an inhomogeneous image is not treated in this thesis as a `texture'.

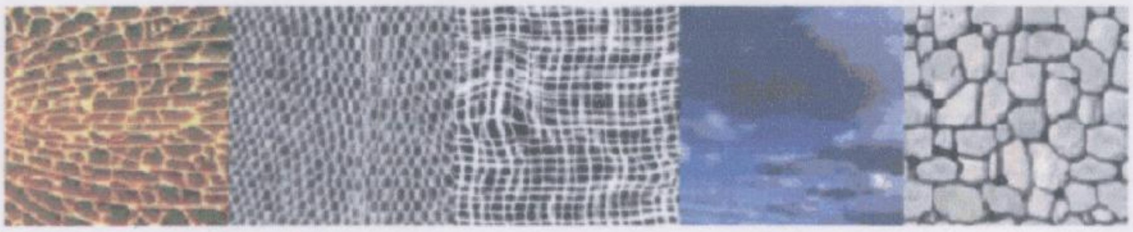


Fig.2.1.(d): Examples of weakly-homogeneous textures

2.1.1. TEXTURE ANALYSIS

Major goals of texture research in computer vision are to understand, model and process texture, and ultimately to simulate human visual learning process using computer technologies.

A typical computer vision system can be divided into components such as the ones shown in Fig 2.1.1.(a). Texture analysis might be applied to various stages of the process. At the preprocessing stage, images could be segmented into contiguous regions based on texture properties of each region. At the feature extraction and the classification stages, texture features could provide cues for classifying patterns or identifying objects.

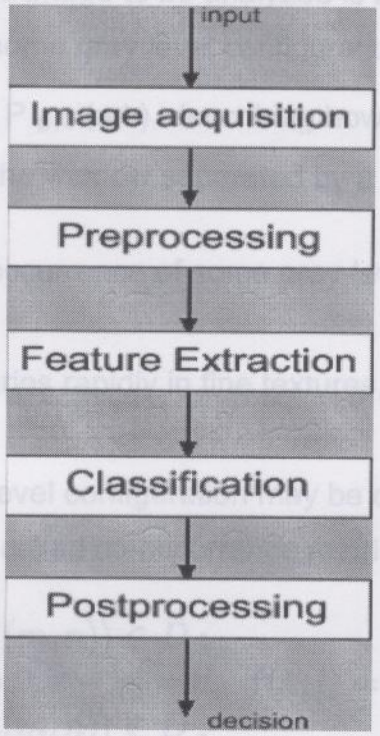


Figure 2.1.1.(a): The components of a typical computer vision system.

2.1.2. CO OCCURRENCE MATRIX AS FEATURE EXTRACTION

There are many methods on feature extraction like co-occurrence matrix, sum and difference histograms, entropy based method, etc. We have used co-occurrence matrix for extraction of features. Detailed description of the method is as given below.

Co-Occurrence Matrices

The Co-occurrence matrix method of texture description is based on the repeated occurrence of some gray-level configuration in the texture: this configuration varies rapidly with distance in fine and slowly in coarse textures. Suppose the part of textured image to be analyzed is an MxN rectangular window. An occurrence of some gray level configuration may be described by a matrix of relative frequency $P_{Ø,d}(a,b)$, describing how frequently two pixels with gray levels (a,b) appear in the window separated by a distance d in direction Ø.

- Based on repeated occurrence of some gray-level configuration in the texture.
- This configuration varies rapidly in fine textures, more slowly in coarse textures
- Occurrence of gray-level configuration may be described by matrices of relative frequencies, called co-occurrence matrices.

$$\begin{aligned}
P_{0^\circ,d}(a,b) &= |\{((k,l),(m,n)) \in D : \\
&\quad k-m=0, |l-n|=d, \;\; f(k,l)=a, f(m,n)=b\}| \\
P_{45^\circ,d}(a,b) &= |\{((k,l),(m,n)) \in D : \\
&\quad (k-m=d, l-n=-d) \; OR \; (k-m=-d, l-n=d), \\
&\quad f(k,l)=a, f(m,n)=b\}| \\
P_{90^\circ,d}(a,b) &= |\{((k,l),(m,n)) \in D : \\
&\quad |k-m|=d, l-n=0, \;\; f(k,l)=a, f(m,n)=b\}| \\
P_{135^\circ,d}(a,b) &= |\{((k,l),(m,n)) \in D : \\
&\quad (k-m=d, l-n=d) \; OR \; (k-m=-d, l-n=-d), \\
&\quad f(k,l)=a, f(m,n)=b\}|
\end{aligned}$$

Example image:

$$\begin{matrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{matrix}$$

Gray level image.

Corresponding co-occurrence matrices:

$$P_{0^\circ,1} = \begin{vmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{vmatrix} \qquad P_{135^\circ,1} = \begin{vmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{vmatrix}$$

Texture classification can be based on criteria (features) derived from the co-occurrence matrices.

- Energy:

$$\sum_{a,b} P^2_{\phi,d}(a,b)$$

- Entropy

$$\sum_{a,b} P_{\phi,d}(a,b) \log_2 P_{\phi,d}(a,b)$$

- Maximum probability

$$\max_{a,b} P_{\phi,d}(a,b)$$

- Contrast (typically, kappa=2, lambda=1)

$$\sum_{a,b} |a-b|^{\kappa} P^{\lambda}_{\phi,d}(a,b)$$

- Inverse difference moment

$$\sum_{a,b;a\neq b} \frac{P^{\lambda}_{\phi,d}(a,b)}{|a-b|^{\kappa}}$$

- Correlation

$$\frac{\sum_{a,b}\left[(ab)\,P_{\phi,d}(a,b)\right]-\mu_x\mu_y}{\sigma_x\sigma_y}$$

Where,

$$\mu_x = \sum_a a \sum_b P_{\phi,d}(a,b)$$

$$\mu_y = \sum_b b \sum_a P_{\phi,d}(a,b)$$

$$\sigma_x = \sum_a (a-\mu_x)^2 \sum_b P_{\phi,d}(a,b)$$

$$\sigma_y = \sum_b (b-\mu_x)^2 \sum_a P_{\phi,d}(a,b)$$

There are methods like sum and difference histogram, entropy based, that can be used for feature extraction. But at some point of time, these methods fail to classify the textures properly.

In case of sum and difference histogram, it is possible that two or more textures can have similar histograms, resulting in misclassification of the same. In case of entropy based method, the entire method is solely depended on a single parameter. Due to noise corruption, the textures tend to have similar entropy values and so the method fails to classify.

2.2. *ARTIFICIAL NEURAL NETWORK*

An artificial neural network is an information processing system that has certain performance characteristics in common with the biological neural network. It is a system loosely modeled on the human brain.

To develop an artificial neural network, one develops a mathematical model that best describes the biological systems functionality. A computer can then simulate the model fairly quickly, and some degree of confidence may be gained with regard to its operation and functionality. Changes may then be made to the model either to enhance its performance or to simplify it.

In operation, each unit of an A.N.N. receives inputs from the connected units and/or from an external source. A waited sum of the inputs is computed at a given instant of time. The activation value determines the actual output from the output function unit.

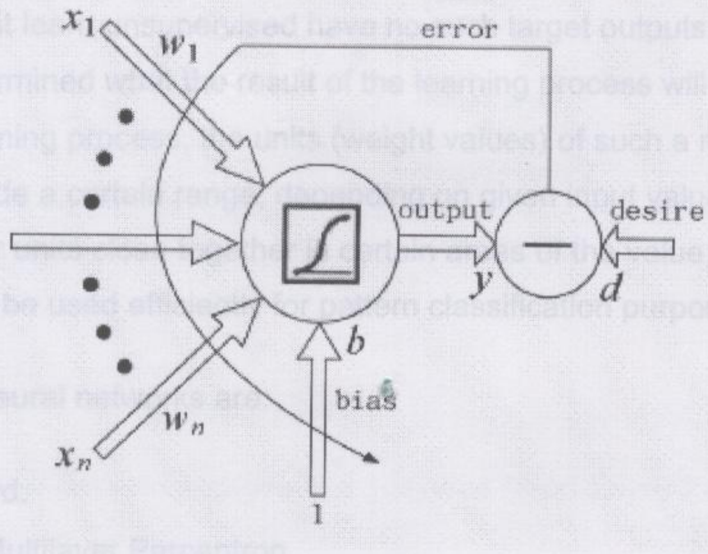


Fig.2.2.(a) Single Input Neuron

Typically the transfer function is chosen by the designer and then the parameters 'W' and 'b' will be adjusted by some learning rule so that the neuron input/output relationship meets some specific goals.

Supervised and Unsupervised learning

The learning algorithm of a neural network can either be supervised or unsupervised.

A neural net is said to learn supervised, if the desired output is already known. While learning, one of the input patterns is given to the net's input layer. This pattern is propagated through the net (independent of its structure) to the net's output layer. The output layer generates an output pattern which is then compared to the target pattern. Depending on the difference between output and target, an error value is computed.
This output error indicates the net's learning effort, which can be controlled by the "imaginary supervisor". The greater the computed error value is, the more the weight values will be changed.

Neural nets that learn unsupervised have no such target outputs.
It can't be determined what the result of the learning process will look like. During the learning process, the units (weight values) of such a neural net are "arranged" inside a certain range, depending on given input values. The goal is to group similar units close together in certain areas of the value range.
This effect can be used efficiently for pattern classification purposes.

Few types of neural networks are:

1. Feed Forward.
2. Single and Multilayer Perceptron.
3. Adaline
4. Radial Basis Function (RBF).

The project focuses on MLP using back propagation algorithm.

2.2.1. BACK PROPAGATION

Back propagation is a supervised learning algorithm and is mainly used by Multi-Layer-Perceptrons to change the weights connected to the net's hidden neuron layer(s).The back propagation algorithm uses a computed output error to change the weight values in backward direction.

To get this net error, a forward propagation phase must have been done before. While propagating in forward direction, the neurons are being activated using the sigmoid activation function.

The formula of sigmoid activation is:

$$f(x) = \frac{1}{1 + e^{-input}}$$

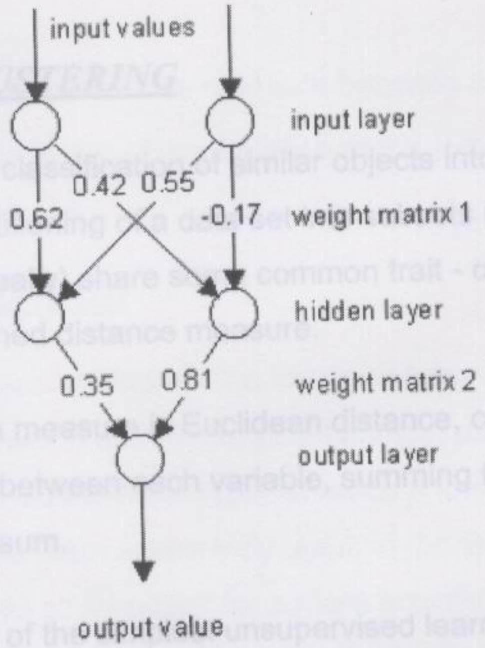


Fig.2.2.1.(a) Backpropagation in a 3-layered Multi-Layer-Perceptron

1. The first input pattern had been propagated through the net.
2. The same procedure is used for the next input pattern, but then with the

changed weight values.

3. After the forward and backward propagation of the second pattern, one learning step is complete and the net error can be calculated by adding up the squared output errors of each pattern.

4. By performing this procedure repeatedly, this error value gets smaller and smaller.

5. The algorithm is successfully finished, if the net error is zero (perfect) or approximately zero.

Note that this algorithm is also applicable for Multi-Layer-Perceptrons with more than one hidden layer.

If all values of an input pattern are zero, the weights in weight matrix 1 would never be changed for this pattern and the net could not learn it. Due to that fact, a "pseudo input" is created, called Bias that has a constant output value of 1.

2.2.2. K-MEANS CLUSTERING

Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure.

A more common measure is Euclidean distance, computed by finding the square of the distance between each variable, summing the squares, and finding the square root of that sum.

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the *n* data points from their respective cluster centres.

2.3. PRESENT SCENARIO

Medical diagnosis is based on information obtained from various sources such as results of clinical examination and histological findings, patient's history, and other data the physician considers in order to reach a final diagnostic decision. Imaging techniques have been extensively used, in the last decade, as a valuable tool in the hands of an expert for a more accurate judgment of patients' condition. In addition, approaches using medical images for automatic detection of lesions have been proposed. Such methodologies can increase expert's identification ability while decreasing the need for aggressive intervention. Moreover the short comings of biopsies, such as discomfort for the patient, delay

in the diagnosis, and limited number of tissue samples can also be minimized.
Some of the areas where texture analysis is used today:

- Diagnosis of the Thyroid Glands.
- Diagnosis of Spinal Cord for back ache problems.
- Detection of Multiple Sclerosis.
- Detection of Tumor in Endoscopic Images.

Chapter 3

SYSTEM BLOCK DIAGRAM AND SPECIFICATIONS

CHAPTER 3

SYSTEM SCHEMATIC AND SPECIFICATIONS

3.1. BLOCK DIAGRAM FOR BASIC TEXTURE CLASSIFICATION

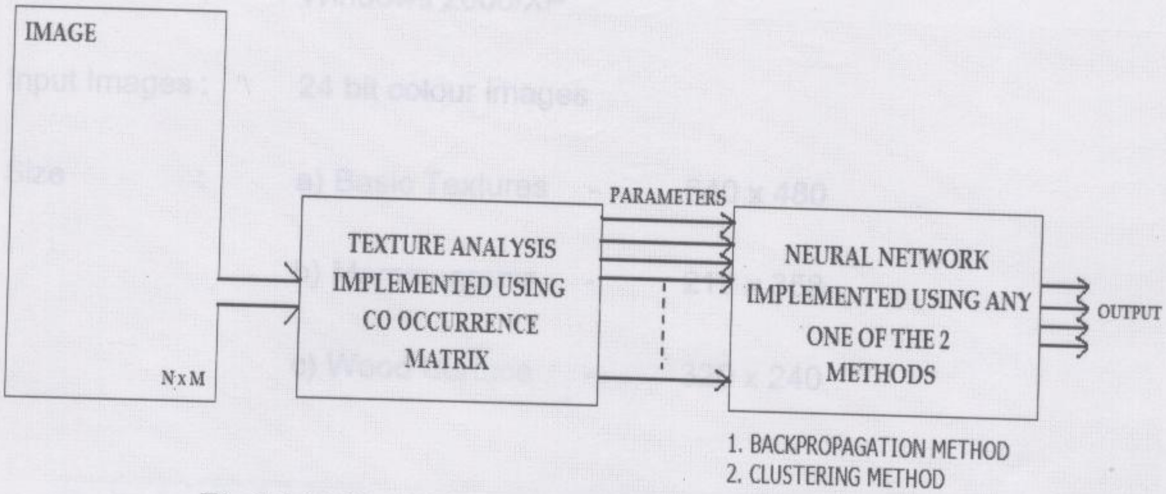


Fig 3.1.(a) Basic Texture Classification Block Diagram

3.2. BLOCK DIAGRAM FOR DETECTION OF MASSSES IN MAMMOGRAMS

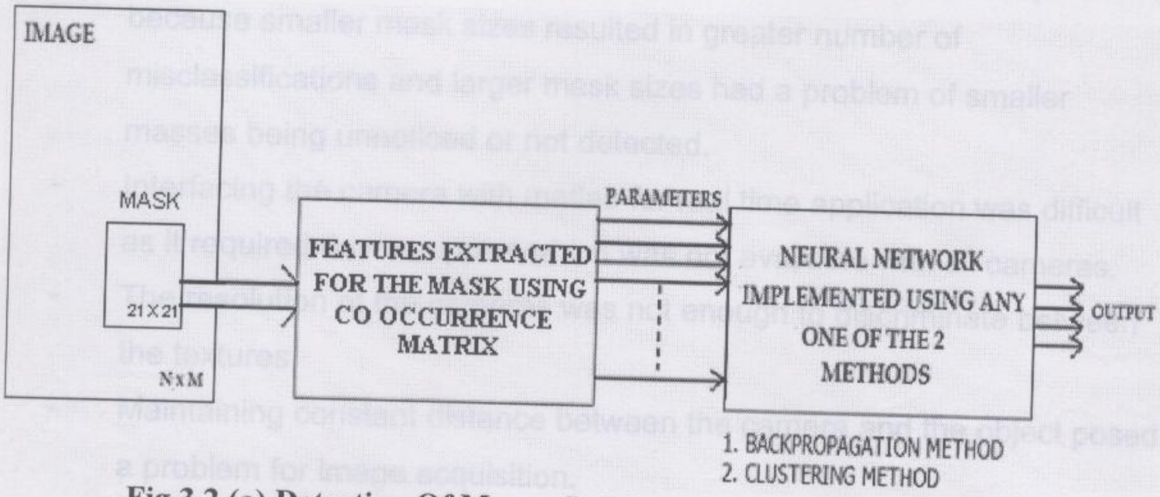


Fig 3.2.(a) Detection Of Masses In Mammograms Block Diagram

3.3. SPECIFICATION

Processor	:	Pentium III onwards		
Platform	:	MATLAB 7.0		
OS	:	Windows 2000/XP		
Input Images	:	24 bit colour images		
Size	:	a) Basic Textures	-	640 x 480
		b) Mammograms	-	212 x 358
		c) Wood Surface	-	320 x 240

3.4. COMPLEXITIES INVOLVED

- Images were not easily available because of patient confidentiality issues.
- Deciding the mask size in the detection of masses was not easy because smaller mask sizes resulted in greater number of misclassifications and larger mask sizes had a problem of smaller masses being unnoticed or not detected.
- Interfacing the camera with matlab for real time application was difficult as it required driver support which was not available with all cameras.
- The resolution of the cameras was not enough to discriminate between the textures.
- Maintaining constant distance between the camera and the object posed a problem for image acquisition.
- Image acquisition was not easy due to improper lighting.
- Deciding the centroid in K-means Clustering

Chapter 4

SYSTEM ALGORITHMS AND FLOWCHARTS

CHAPTER 4
SYSTEM ALGORITHMS

4.1. ALGORITHMS

4.1.1. Co Occurrence Matrix Algorithm

1. Get image matrix in 'D'.
2. Initialize 'd' to '1'.
3. For matrix at 0deg.,

   apply,

$$\begin{aligned} P_{0^\circ,d}(a,b) \quad &= \quad |\{((k,l),(m,n)) \in D : \\ &k-m=0, |l-n|=d, \quad f(k,l)=a, f(m,n)=b\}| \end{aligned}$$

Where (k,l) and (m,n) are two pixel locations in the same row separated by 1 column. 'a' and 'b' are the pixel intensities at these locations respectively.'$P_{0,d(a,b)}$ is the final co occurrence matrix at 0deg.

4. For matrix at 45deg.,

   apply,

$$\begin{aligned} P_{45^\circ,d}(a,b) \quad &= \quad |\{((k,l),(m,n)) \in D : \\ &(k-m=d, l-n=-d) \ OR \ (k-m=-d, l-n=d) \\ &f(k,l)=a, f(m,n)=b\}| \end{aligned}$$

Where (k,l) and (m,n) are two pixel locations at an angle of 45deg with respect to each other. 'a' and 'b' are the pixel intensities at these locations respectively.'$P_{45,d(a,b)}$ is the final co occurrence matrix at 45deg.

5. For matrix at

   90deg.,apply,

$$\begin{aligned} P_{90^\circ,d}(a,b) \quad &= \quad |\{((k,l),(m,n)) \in D : \\ &|k-m|=d, l-n=0, \quad f(k,l)=a, f(m,n)=b\}| \end{aligned}$$

Where (k,l) and (m,n) are two pixel locations at an angle of 90deg with respect to each other. 'a' and 'b' are the pixel intensities at these locations respectively.'$P_{90,d(a,b)}$ is the final co occurrence matrix at 90deg.

6. For matrix at

135deg.,apply,

$$\begin{aligned} P_{135^o,d}(a,b) \;=\; & |\{((k,l),(m,n)) \in D : \\ & (k-m=d, l-n=d) \;\; OR \;\; (k-m=-d, l-n=-d), \\ & f(k,l)=a, f(m,n)=b\}| \end{aligned}$$

Where (k,l) and (m,n) are two pixel locations at an angle of 135deg with respect to each other. 'a' and 'b' are the pixel intensities at these locations respectively.'$P_{135,d(a,b)}$ is the final co occurrence matrix at 135deg.

4.1.2. Back Propagation Algorithm

Step 0: Intialize weights.(Set to small random values.)

Step 1: While stopping condition is false,do steps 2-9.

Step 2: For each training pair,do steps 3-8.

Feedforward:

Step 3: Each input unit(X_i i = 1…….n) receives input signal x_i and broadcasts

this signal to all units in the hidden layer above (the hidden units.

Step 4: Each hidden unit(Z_j j = 1……p) sums its weighted input signals.

$$Z\_inj = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

applies its activation function to compute its output signal.

$$Z_j = f(Z\_inj)$$

And sends this signal to all units in the layer above(output units).

Step 5: Each output unit (Y_k =1……m) sums its weighted input signals.

$$Y\_ink = w_{ok} + \sum_{i=1}^{p} z_j w_{jk}$$

And applies its activation function to compute its output signal.

Back propagation of error.

Step 6: Each output unit receives a target pattern corresponding to the input training pattern, computes its error information term.

$$\delta_k = (t_k - y_k)\, f'(y\_in_k)$$

calculates its bias correction term(used to update w_{jk})

$$\Delta w_{jk} = \alpha \delta_k z_j$$

calculates its bias correction term

$$\Delta w_{ok} = \alpha \delta_k$$

and sends δ_k to units in the layer below.

Step 7: Each hidden unit sums its delta inputs (from above in the layer),

$$\Delta\_inj = \sum_{k=1}^{m} \delta_k w_{jk},$$

Multiplies by the derivative of its activation function to calculate its error information term.

$$\Delta_j = \delta\_inj'(z\_inj),$$

Calculates its weight correction term

$$\Delta v_{jk} = \alpha \delta_k x_j$$

and calculates its bias correction term

$$\Delta v_{oj} = \alpha \delta_j$$

Update weights and biases

Step 8: Each output unit updates its biases and weights

$$w_{jk}(new) = w_{jk(old)} + \Delta w_{jk}$$

each hidden unit updates its biases and weights

$$v_{jk}(new) = v_{jk(old)} + \Delta v_{jk}$$

Test stopping condition.

4.1.3. K-means Clustering Algorithm

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.

Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

4.2. FLOWCHART

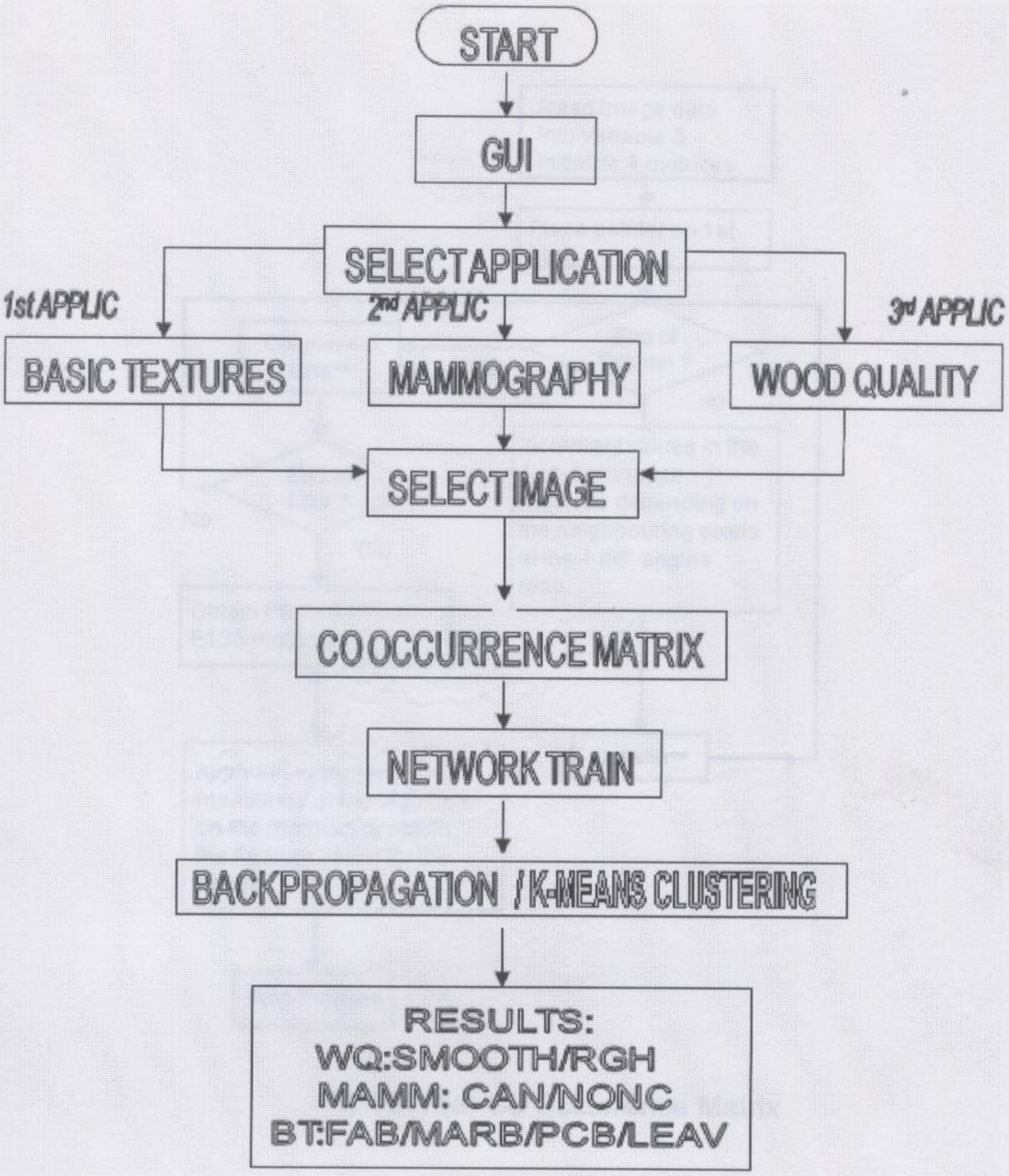


Fig 4.2.(a) Flow Chart

4.2.1. FLOWCHART FOR CO OCCURRENCE MATRIX

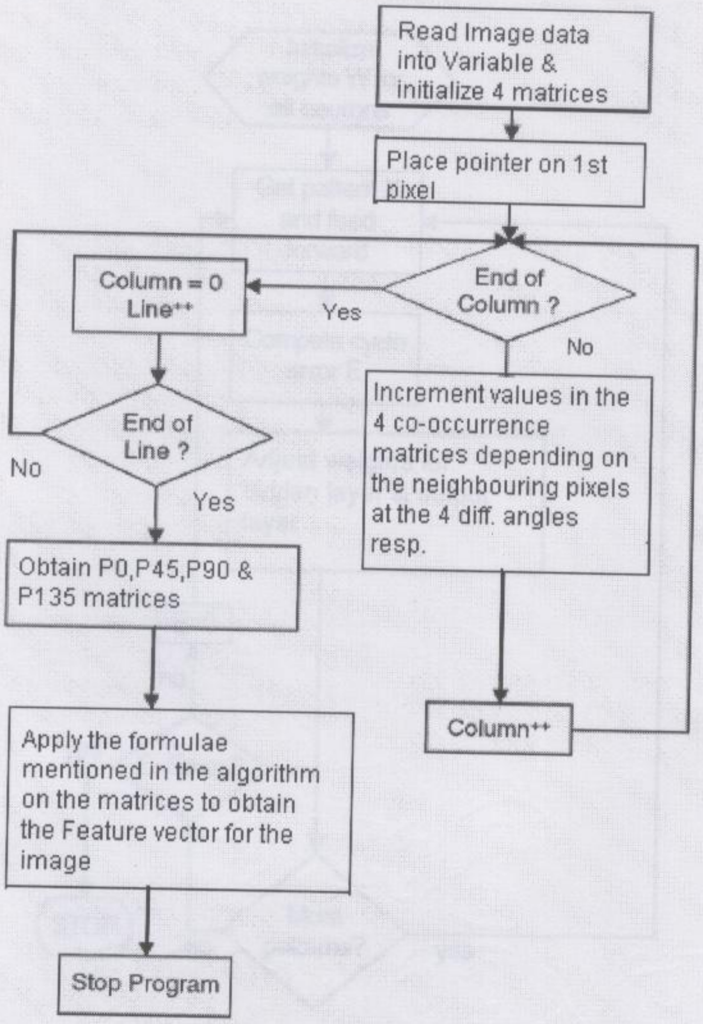


Fig 4.2.1.(a) Co Occurrence Matrix

4.2.2. FLOWCHART FOR BACK PROPAGATION

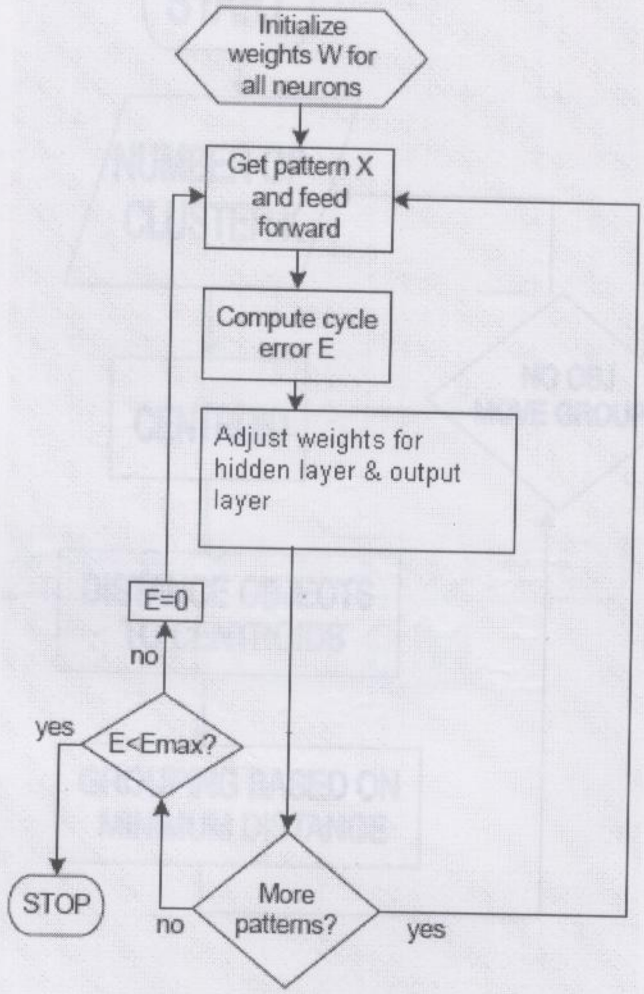


Fig 4.2.2.(a) Back Propagation

4.2.3. <u>FLOWCHART FOR K-MEANS CLUSTERING</u>

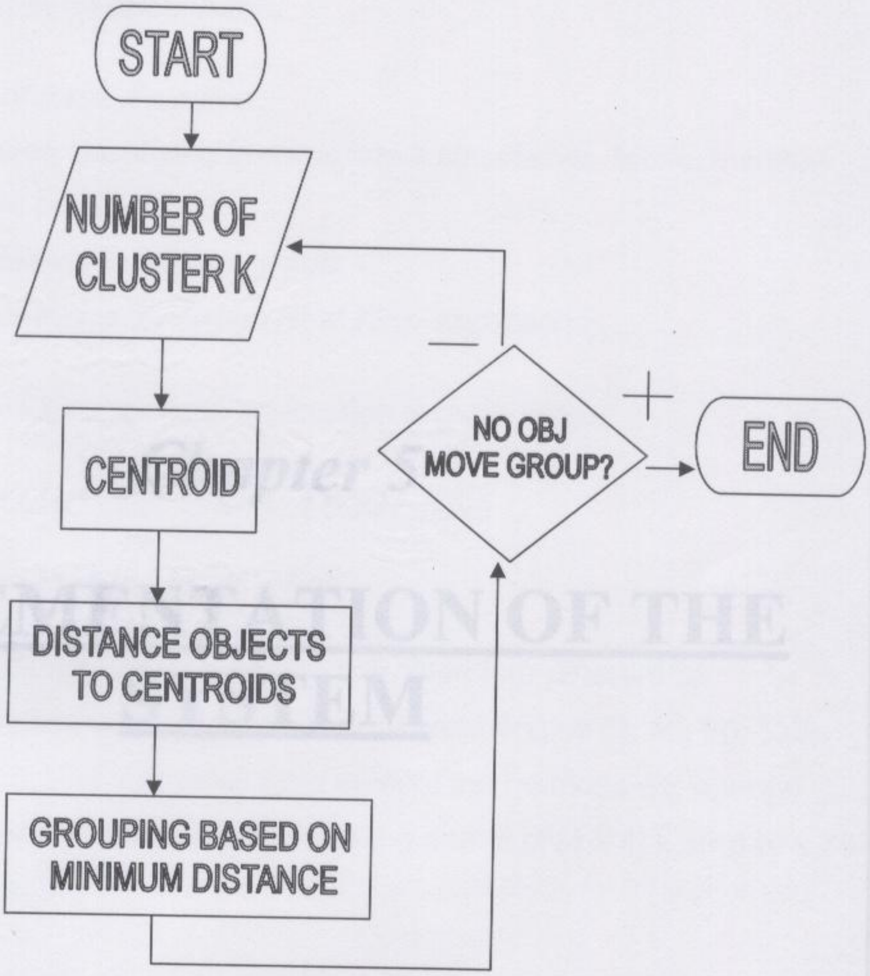


Fig 4.2.3.(a) K-Means Clustering

Chapter 5

IMPLEMENTATION OF THE SYSTEM

CHAPTER 5

IMPLEMENTATION OF THE SYSTEM

Our project deals with three applications:

1. *Classification of Basic Textures.*
   - This involves classifying textures into 4 classes viz. fabric, marble, leaves and pcbs.
2. *Detection of Masses in Mammograms.*
3. *Classification of Wood Surfaces (Real Time Application).*

The detailed implementation for each application is given below.

5.1. CLASSIFICATION OF BASIC TEXTURES

5.1.1. Co Occurrence Matrix Construction

The image is first acquired into a variable. Then four different co occurrence matrices are constructed for different angles (0, 45, 90, 135). Four features are then extracted from each of the matrices viz. energy, entropy, homogeneity, correlation, so giving a total of [4 x 4] x 3(r,g,b) = 48 features per image.

5.1.2. Training of the net

We have used a total of 34 images belonging to different classes for training using back propagation algorithm. The features are normalized and then fed into a built-in toolbox of matlab. The structure of the net is as given below:

Input Nodes	-	48
Hidden Nodes	-	60

Output Nodes	-	4
No. Of Epochs	-	45
Learning Rate	-	0.65
MSE	-	0.0001

5.1.3. Testing of the net

Features are extracted from the test image and fed into the trained neural net which then classifies the image. We tested the net with 30 test images taken from the database.

5.2. ***DETECTION OF MASSES IN MAMMOGRAMS***

5.2.1. Co Occurrence Matrix Construction

The image is first acquired into a variable. A mask of 21 x 21 pixels is moved over the image. At each location four different co occurrence matrices are constructed for different angles (0, 45, 90, 135). Each of the matrices are segmented into five quadrants. Three features are then extracted from each of the quadrants viz. energy, entropy, homogeneity, so giving a total of [15 x 4] = 60 features per image.

5.2.2. Training of the net

The features are extracted from around 10 images which includes both mass and non mass. Sample templates of 21 x 21 pixels are extracted from the main image. We used total of 37 templates for training using back propagation algorithm. The features are normalized and then fed into a built-in toolbox of matlab. The structure of the net is as given below:

Input Nodes	-	60

and then fed into a built-in toolbox of matlab. The structure of the net is as given below:

Input Nodes	-	48
Hidden Nodes	-	15
Output Nodes	-	2
No. Of Epochs	-	200
Learning Rate	-	0.55
MSE	-	1e -11

5.3.4. Testing of the net

Images of new blocks are then used for testing. Features are extracted from the test image and fed into the trained neural net which then classifies the image.

5.4. *GRAPHICAL USER INTERFACE AS FRONT END*

A GUI was created using matlab which links all the program files created. Drop down menu is provided to select the application. The "Open Image" helps you to select the image and is displayed on window1. A push button "Rotate Image" option is provided to rotate the image to any angle required. The classification is shown on window2 after the "RUN" button is pushed.

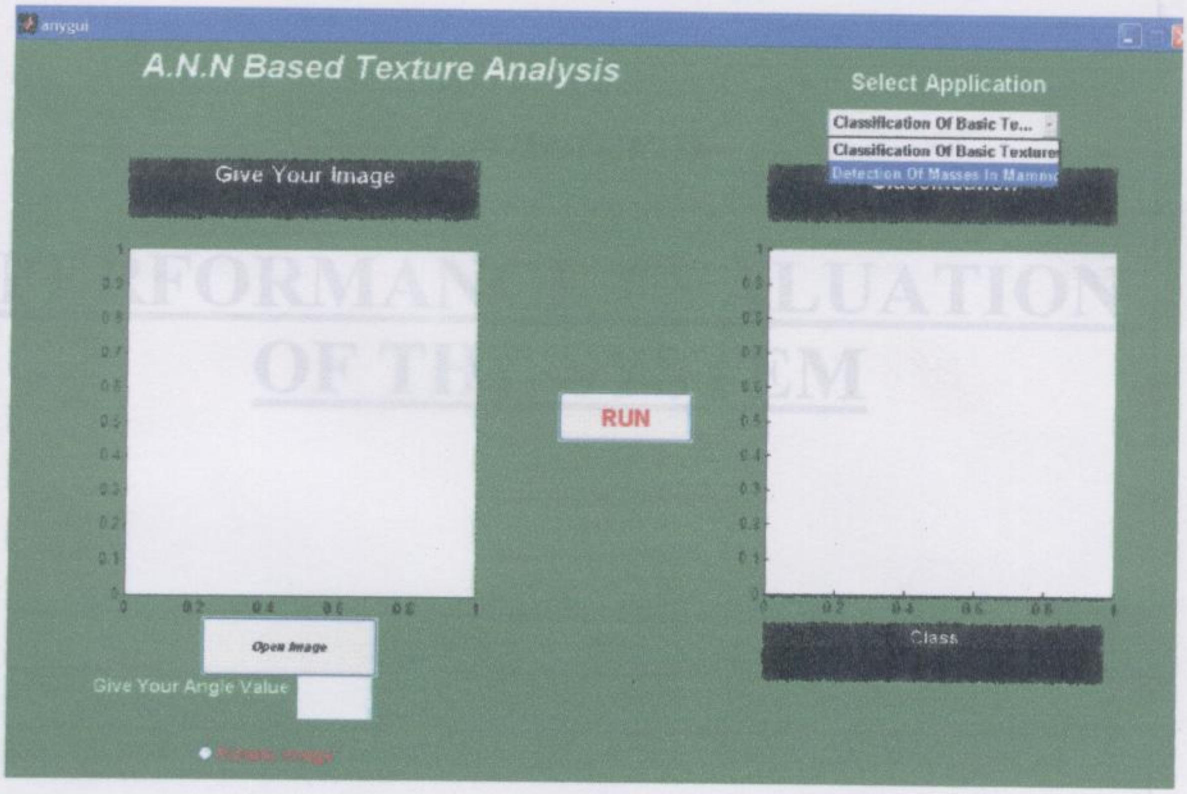


Fig 5.4.(a) GUI

Chapter 6

PERFORMANCE EVALUATION OF THE SYSTEM

CHAPTER 6

PERFORMANCE EVALUATION OF THE SYSTEM

6.1. STEP BY STEP RESULTS

SELECT APPLICATION – CLASSIFICATION OF BASIC TEXTURES

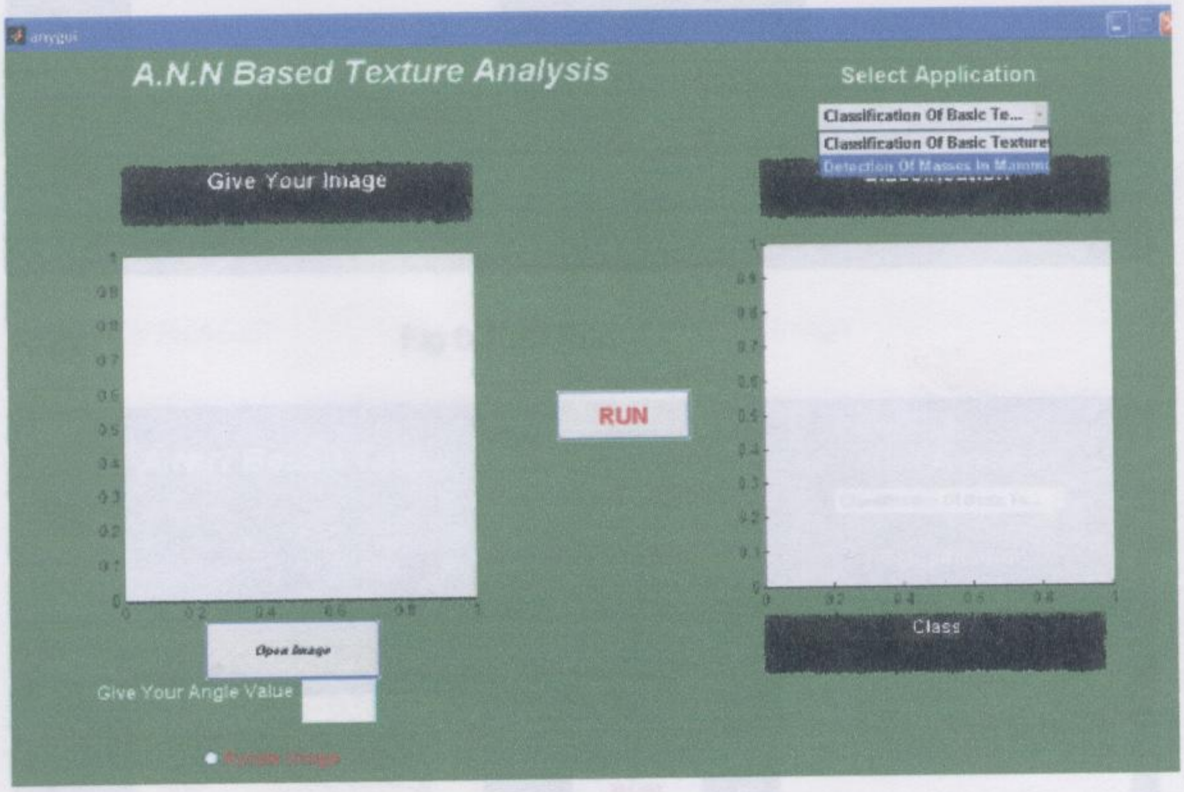
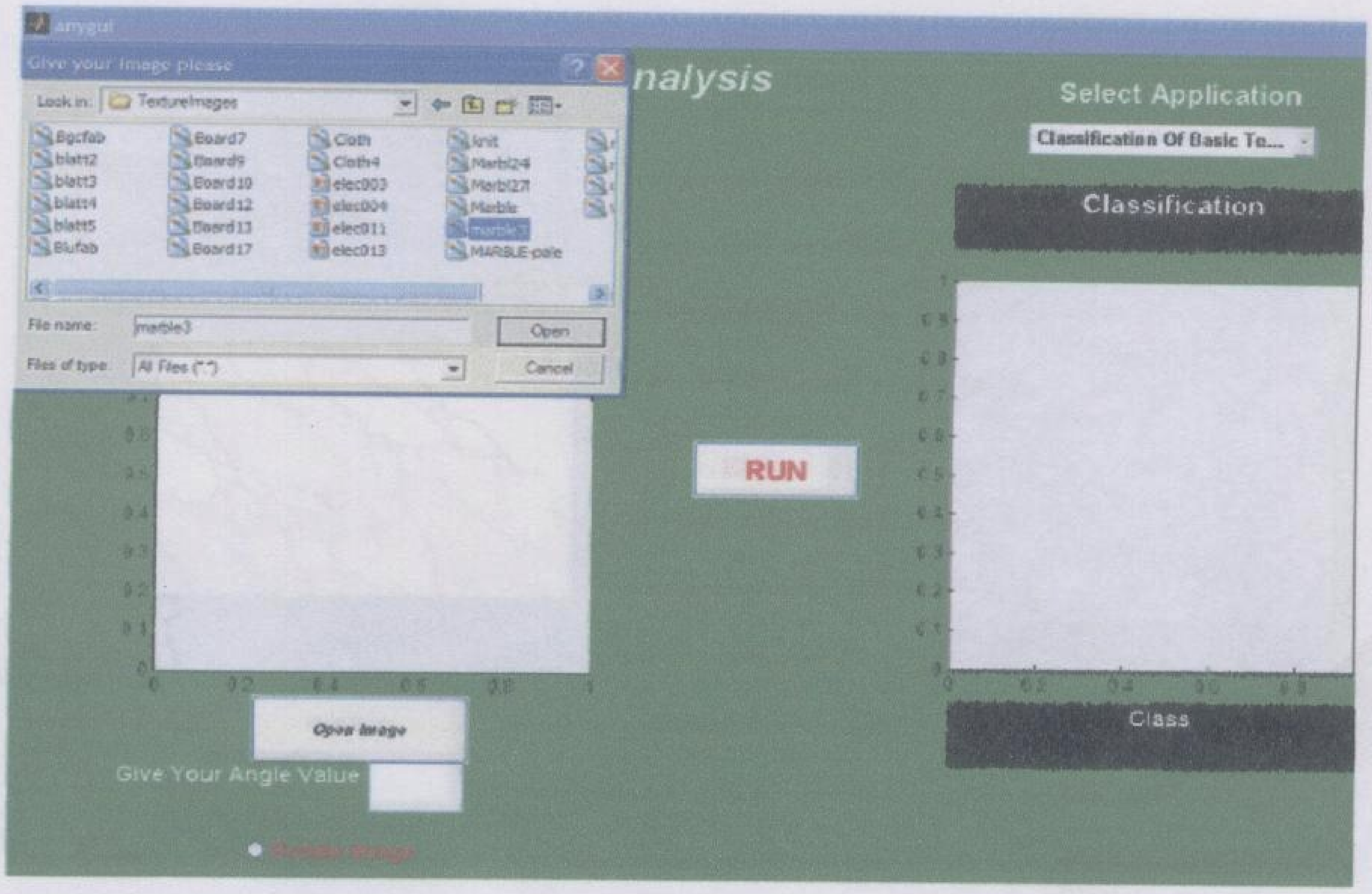
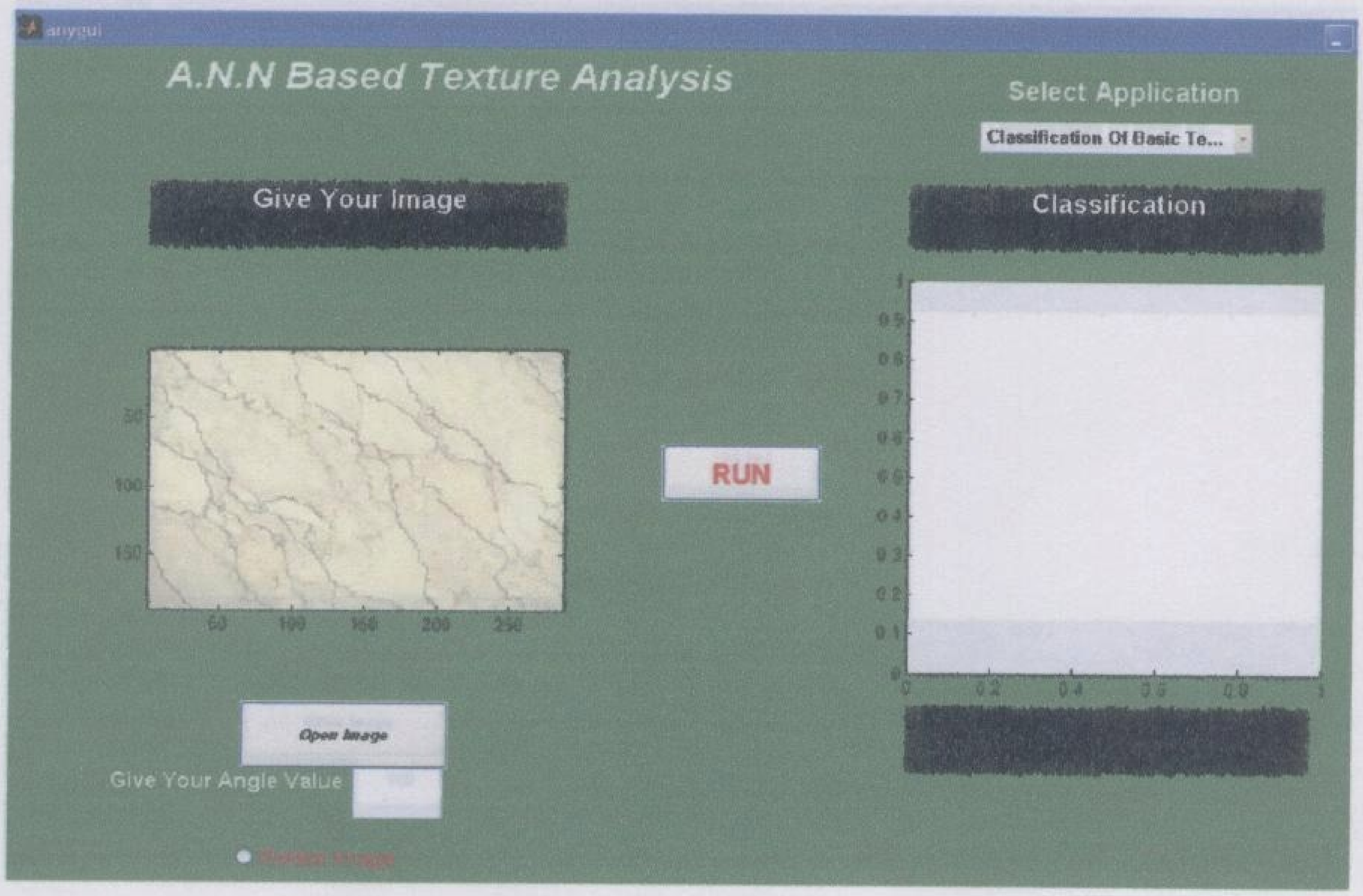


Fig 6.1.(a) Select Application

OPEN IMAGE Fig 6.1.(b) Open Image

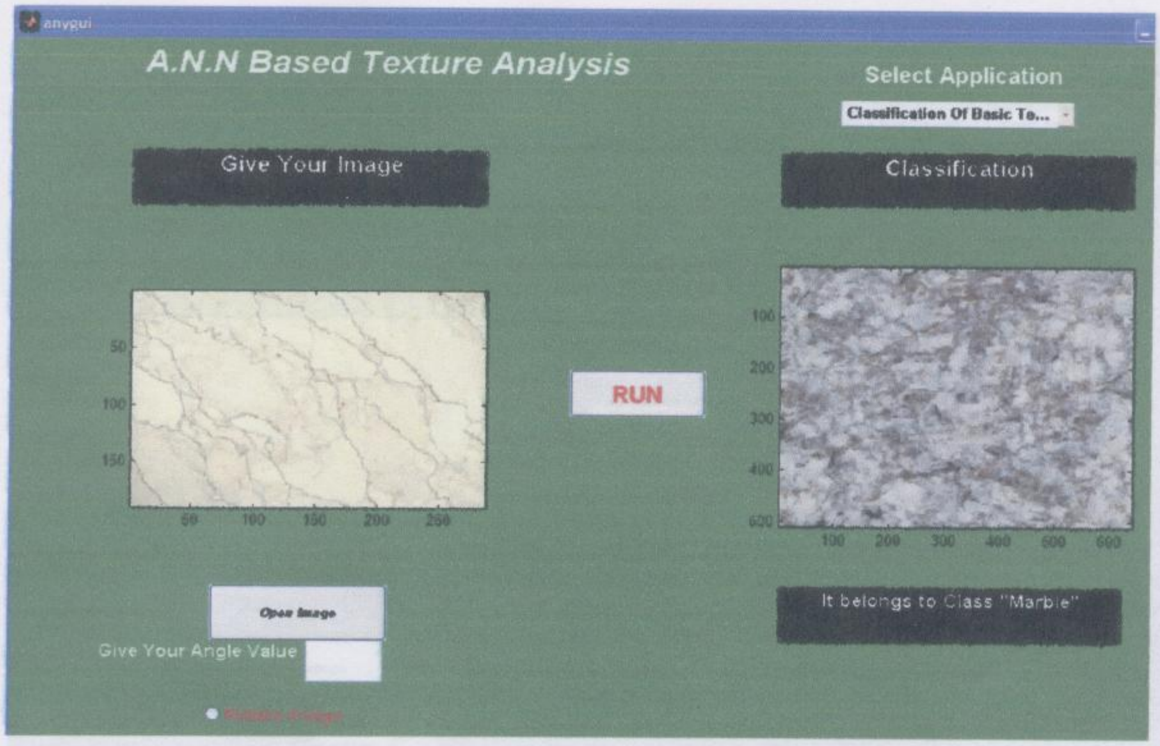


RUN Fig 6.1.(c) Run



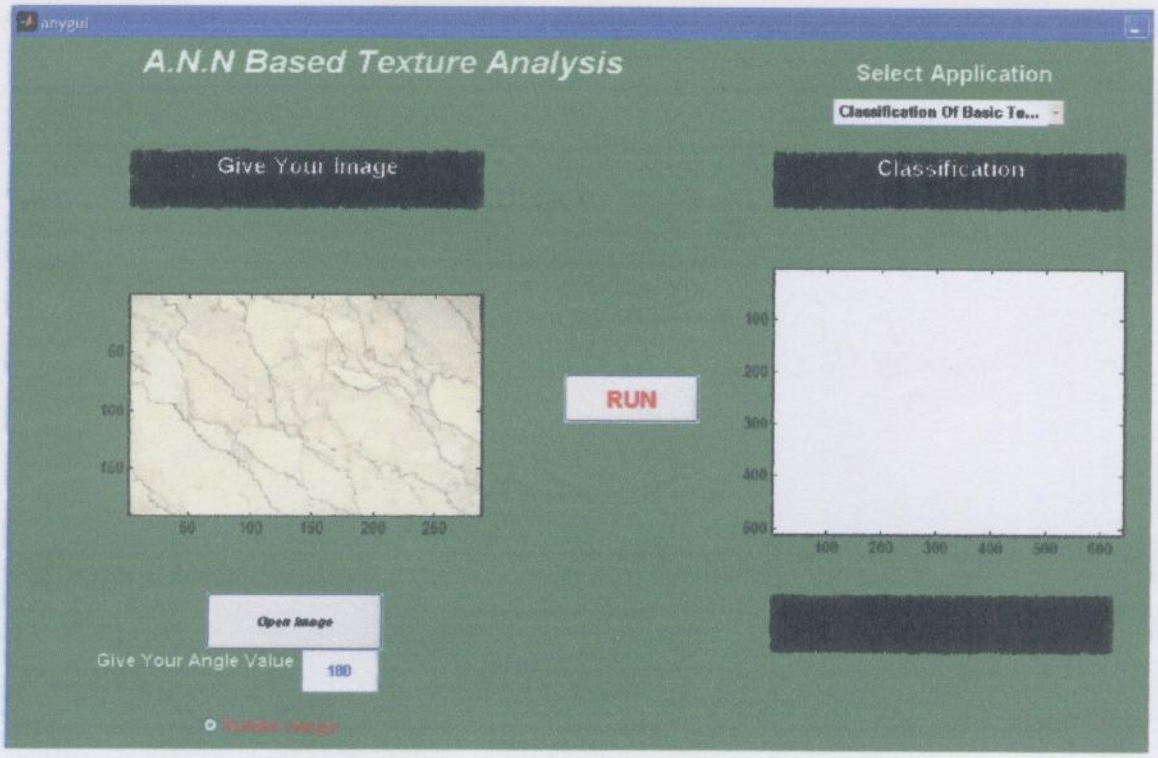
RESULT

Fig 6.1.(d) Result



ROTATE IMAGE

Fig 6.1.(e) Rotate Image



RESULT

Fig 6.1.(f) Result

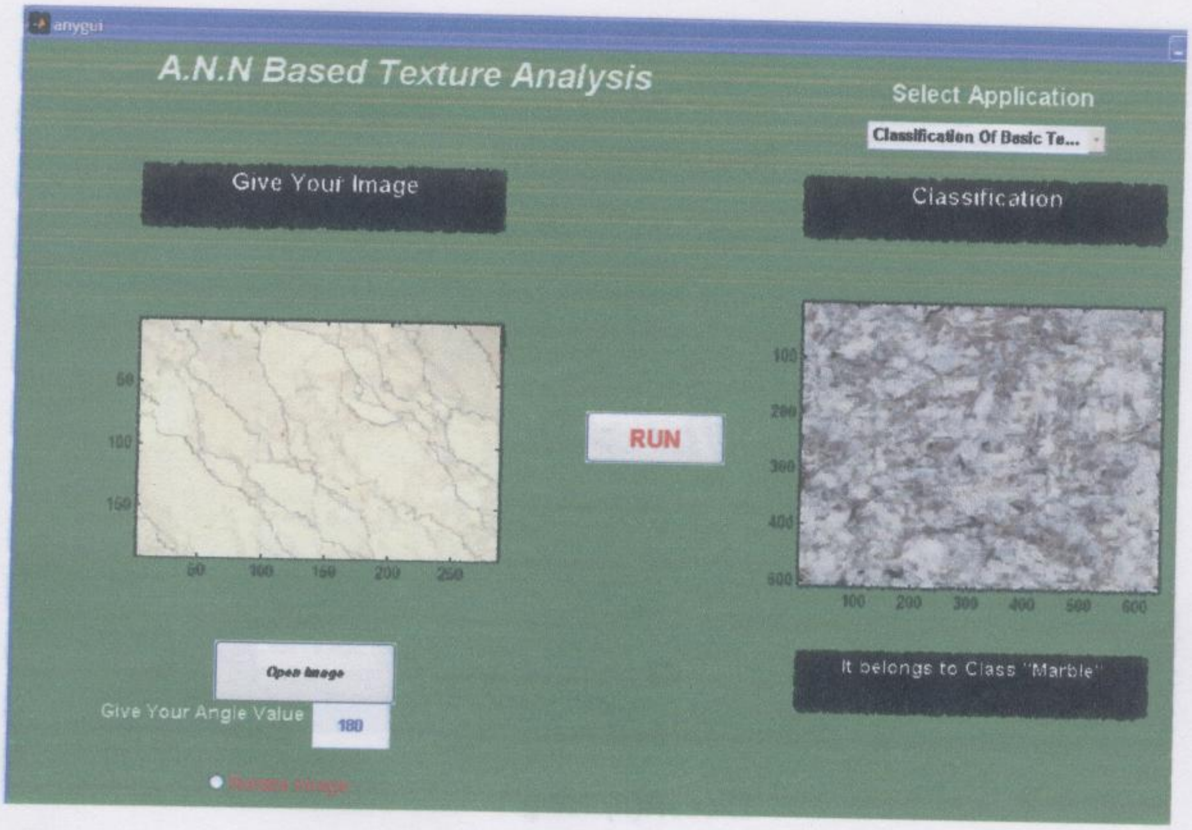


Fig 6.1.(g)Result1

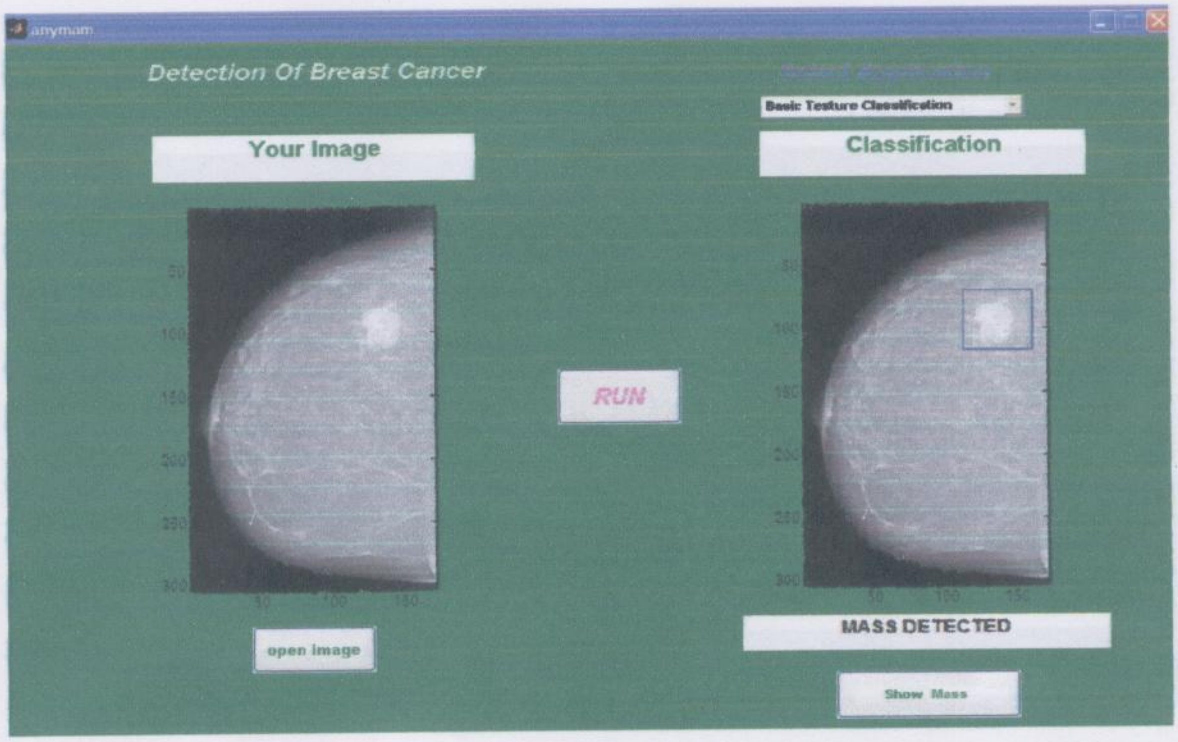


Fig 6.1.(h)Result2

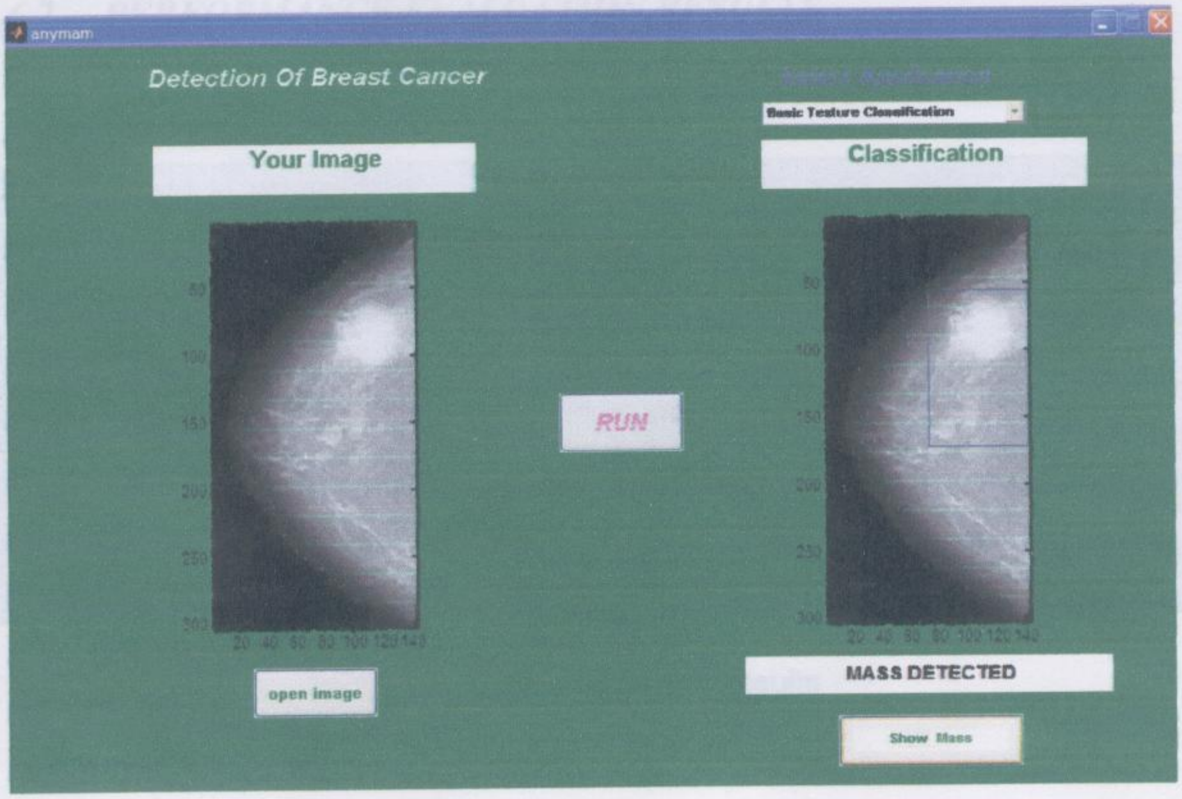
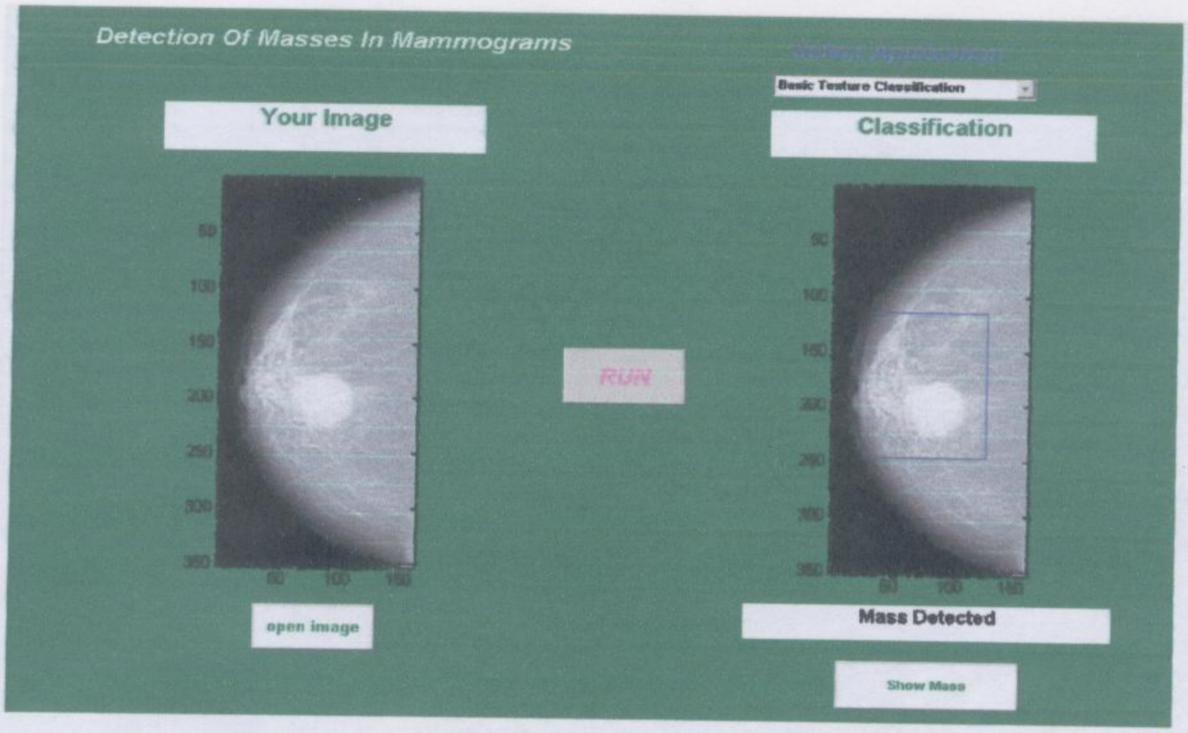


Fig 6.1.(i)Result3



6.2. PERFORMANCE EVALUATION RESULTS

SYSTEM WITH BACK PROPAGATION ALGORITHM

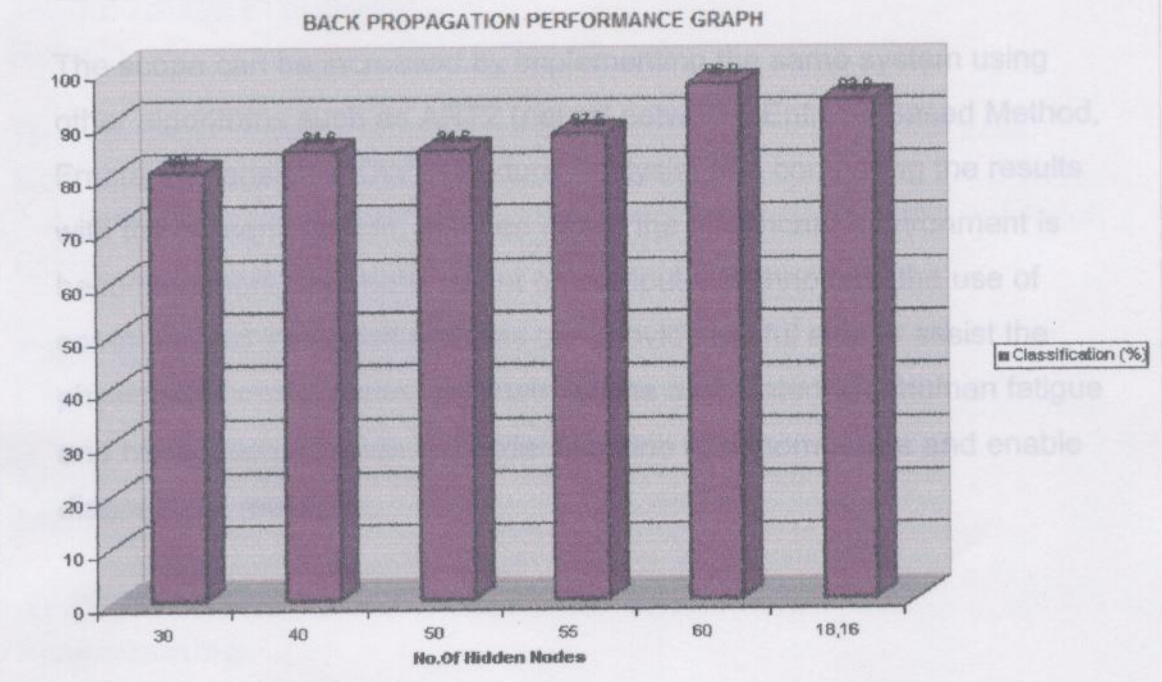
HIDDEN LAYER NODES		LEARNING RATE	CLASSIFICATION (%)
LAYER1	LAYER2		
55	NIL	0.45	87.5
60	NIL	0.65	96.9
30	NIL	0.45	80.7
50	NIL	0.45	84.6
40	NIL	0.53	84.6
18	16	0.45	93.9

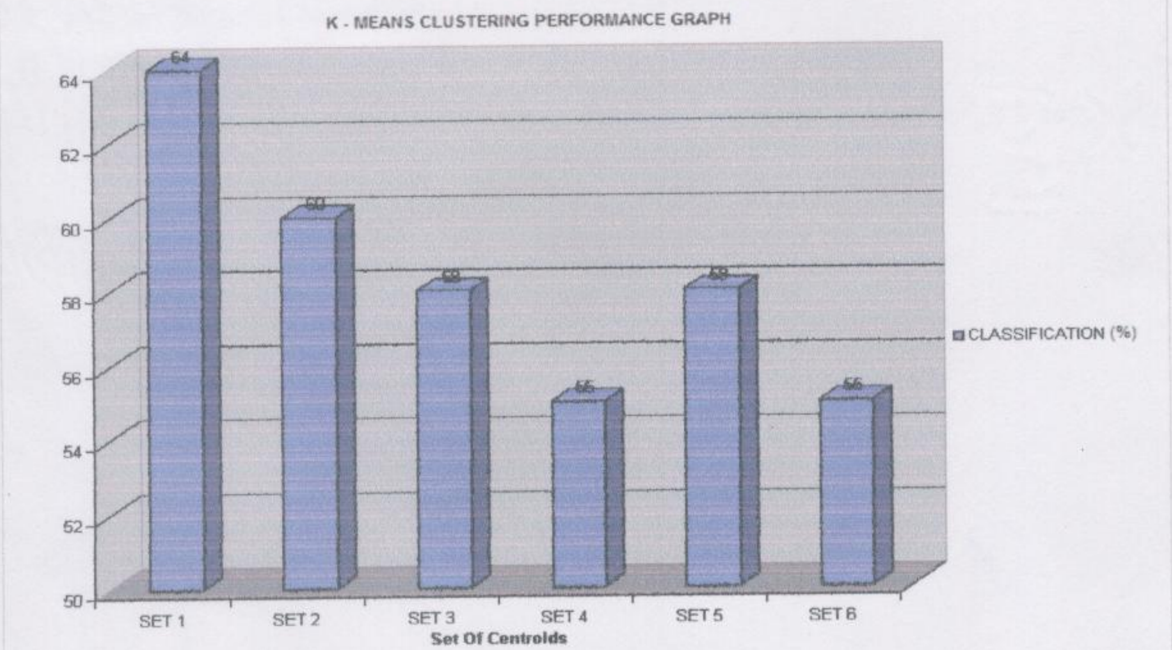
Tab 6.2.(a) Back Pro Results

CENTROIDS		CLASSIFICATION (%)
C1	SAMPLE 2	64
C2	SAMPLE 21	
C3	SAMPLE 25	
C4	SAMPLE 28	
C1	SAMPLE 2	60
C2	SAMPLE 16	
C3	SAMPLE 25	
C4	SAMPLE 30	
C1	SAMPLE 2	58
C2	SAMPLE 12	
C3	SAMPLE 25	
C4	SAMPLE 29	
C1	SAMPLE 2	55
C2	SAMPLE 14	
C3	SAMPLE 25	
C4	SAMPLE 29	
C1	SAMPLE 2	58
C2	SAMPLE 9	
C3	SAMPLE 25	
C4	SAMPLE 29	
C1	SAMPLE 2	55
C2	SAMPLE 21	
C3	SAMPLE 25	
C4	SAMPLE 28	

Tab 6.2.(b) K means Clustering Results

6.3. PERFORMANCE GRAPHS





FUTURE SCOPE

The scope can be increased by implementing the same system using other algorithms such as ART2 (neural network), Entropy Based Method, Fractal Dimension & DWT (Texture Analysis) and comparing the results with the present system. In times where the healthcare environment is becoming more and more reliant on computer technology, the use of computerized intelligent systems can provide useful aids to assist the physician in many cases, eliminate issues associated with human fatigue and habituation, provide rapid identification of abnormalities and enable diagnosis in real time.

REFERENCES

1. **Image Processing Analysis And Machine Vision**
   Milan Sonka, Vaclav Hlavac, Roger Boyle

2. **Digital Image Processing**
   Rafael C. Gonzalez, Richard E. Woods

3. **Detection Of Masses In Mammograms Using Texture Features**
   Keir Bovis and Sameer Singh
   P A " Research, Department of Computer Science, University of Exeter, Exeter, UK

4. **Fundamentals of Neural Network**
   Lauren Fausett

5. **Elements of Artificial Neural Network**
   Mehrotra, Mohan, Ranka

6. **Artificial Neural Network Algorithm Applications and Programming**
   Freeman

7. **Artificial Neural Network**
   B Yegnanarayan