CHAPTER 4

MAMMOGRAM RETRIEVAL USING FUZZY NEURAL NETWORK CLASSIFIER TO DETECT BREAST CANCER

4.1 INTRODUCTION

To improve the performance accuracy of retrieval output and classification, in this chapter Modified Fisher’s Linear Discriminant Analysis (MFLDA) and Fuzzy Neural Network (FNN) based classification with CBIR proposed for prediction of breast cancer lesions. Initially, CBIR extracts features from a query image and computes a similarity measure, then gives results by detecting breast cancer lesions.

4.2 PROPOSED FRAMEWORK

Content based retrieval system consists of two phases are training phase and testing phase. The proposed system and its component of mammogram retrieval is given in Figure 4.1. The proposed system comprises of image preprocessing, segmentation, feature extraction, feature selection and classification. The Wiener filter is used in a preprocessing stage to remove the unwanted pixels in the original image. Feature vector is formed using three types of feature extraction, such as GLCM, Gabor filter and LBP from the denoised image. Modified Fisher’s Linear Discriminant Analysis (MFLDA) is used to select the specified features from the obtained feature vector.
Figure 4.1 The proposed Framework and its components of Mammogram retrieval

Mahalanobis Distance is used to find the similarity between the feature vector from the training phase and feature selection from testing phase. The retrieved image from image database, even though it consists of some non-retrieved image. To improve the CBIR retrieved image accuracy, the FNN classification is introduced in this proposed framework. FNN classification is used to classify between query image feature vector and retrieved image features.

4.3 IMAGE PREPROCESSING

The original mammogram image consists of lighting illumination, poor contrast and noises; it is due to camera settings. To alleviate these problems wiener filter is proposed in this framework. The Wiener filter is used
to remove the noise in the given image database and also for contrast enhancement. It is helpful to segment the mammogram images clearly and detect the retrieved images properly.

In preprocessing stage, artifacts removal is done. High intensity radio opaque artifacts could lead to misclassification and affect the detection of breast density. Therefore, before segmentation method is carried out the mammogram images are pre-processed for decreasing the misclassification rate and used for further processing. In this proposed work, the preprocessing using wiener filters for mammogram images are used.

The Wiener filter tries to make an optimum estimate of the initial image by implementing a minimum Mean Square Error (MSE) constraint between the estimate and original image. It is mentioned an optimum filter. The target of a wiener filter is to minimize the MSE. A Wiener filter has the ability of managing both the degradation function as well as noise. In the proposed method, the noisy image is pre-processed to remove the noise. It is mentioned as

\[ M(i, j) = x(i, j) + y(i, j) \quad (4.1) \]

Where \( M(i, j) \rightarrow \) the noisy measurement, \( x(i, j) \rightarrow \) is the noise free image and \( y(i, j) \rightarrow \) is additive Gaussian noise. The goal is to remove noise, or denoised \( M(i, j) \), and to obtain a linear estimate \( \hat{x}(i, j) \) of \( x(i, j) \) which minimizes the mean squared error

\[ MSE(\hat{x}) = \frac{1}{N} \sum_{i,j=1}^{N} (\hat{x}(i, j) - x(i, j))^2 \quad (4.2) \]

Where \( N \rightarrow \) number of elements in \( x(i, j) \).
When $x(i,j)$ and $y(i,j)$ are stationary Gaussian processes the Wiener filter is the optimal filter. Specifically, when $x(i,j)$ a Gaussian process is the proposed wiener filter has a very simple scalar form

$$
\hat{x}(i,j) = \frac{\sigma_x^2(i,j)}{\sigma_x^2(i,j) + \sigma_y^2(i,j)} [M(i,j) - \mu_x(i,j)] + \mu_x(i,j)
$$

(4.3)

Where $\sigma^2 \rightarrow$ the signal variances and $\mu \rightarrow$ mean, and where normally assume the mean of the noise to be zero.

Figure 4.2 illustrates the preprocessing results. Figure 4.2(a) gives the original mammogram image and it is affected by noise through camera settings. So the noisy pixels from the original mammogram are removed using wiener filter and are shown in Figure 4.2(b).

4.4 THE MORPHOLOGICAL APPROACH TO SEGMENTATION-A WATERSHED ALGORITHM

Watershed segmentation is a gradient-based segmentation technique. In this process, the gradient map of the image considered as a relief map and the image segmented as a dam. This segmented regions mentioned as catchment basins. A variety of image segmentation problems are solved by
Watershed segmentation. It is suitable for the images that have higher intensity value. It is caused over segmentation and it controlled by marker controlled watershed segmentation is used. For edge detection, Sobel operator is suitable. In marker controlled watershed segmentation, sobel operator is used to distinct the edge of the object. The sobel masks in matrix form are as follows

\[
M_x = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{bmatrix}, M_y = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 0 \\
-1 & 0 & 2 \\
\end{bmatrix}
\]

The equation of gradient magnitude used in marker controlled watershed segmentation is

\[M = \sqrt{M_x^2 + M_y^2}\]

\[\text{Angle, } \theta = \tan^{-1} \frac{M_y}{M_x}\]

### 4.5 FEATURE EXTRACTION

The feature extraction is the key to the success; it extracts the relevant information of the mammogram images from the segmented part. The Extracted information of the mammogram image is useful for further processing of classification and retrieval function. There are three types of feature extraction are used in this chapter are GLCM, Gabor filter and LBP. The reason for choosing these features is given below:

- GLCM provides the texture properties of a mammogram image.
- Gabor filter used selectivity for orientation, spectral bandwidth and spatial extent of the mammogram image.
- LBP used to obtain the binary features with their histogram of a mammogram image.
4.5.1 Feature Extraction Using GLCM

Gray-Level Co-Occurrence Matrix (GLCM) defines the spatial association among each intensity quality with considering changes between gray levels $i$ and $j$ at a particular displacement distance $d$ and at an exacting angle $\theta$. Seven properties from GLCM are computed such as energy measuring uniformity of local gray scale distribution, inertia measuring the local variations, correlation measuring the joint probability of occurrence, entropy measuring randomness, homogeneity measuring the closeness of the distribution and the inverse difference moment measuring local minimal changes, cluster shade measuring a group of pixels that have parallel gray level values. Here, extract 28 features as well as exploit the standard values of all angles to obtain 7 features. The detailed explanation of GLCM features is given in chapter 3 from equation (3.15 to 3.40).

4.5.2 Gabor Filters for Feature Extraction

Gabor filter is used in many applications such as computer vision (Porat & Zeevi 1998), texture analysis (Jain & Farrokhnia 1991; Turner, 1986) and face recognition (Lampinen & Oja 1995). An important factor of Gabor filter is that they have joint localization, or resolution in both spatial and the spatial-frequency domains (Daugman 1985).

Basically the 2-D Gabor filter is entered at $(0, 0)$ in the spatial domain and it is defined as:

$$G(x, y, \xi_x, \xi_y, \sigma_x, \sigma_y, \theta) = \frac{1}{\sqrt{\pi \sigma_x \sigma_y}} e^{-\frac{1}{2} \left[ \left( \frac{R_1}{\sigma_x} \right)^2 + \left( \frac{R_2}{\sigma_y} \right)^2 \right]} e^{i(\xi_x x + \xi_y y)} \quad (4.4)$$

Where $R_1 = xcos\theta + ysin\theta$ and $R_2 = -xsin\theta + ycos\theta$, spatial frequency is represented as $\xi_x$ and $\xi_y$, standard deviation of an elliptical
Gaussian is represented as $\sigma_x, \sigma_y$ along x and y axes, orientation of the Gabor filter is denoted by symbol $\theta$.

\[ \xi_x = \omega \cos \theta \text{ and } \xi_y = \omega \sin \theta, \quad \text{Where } \omega = \sqrt{\xi_x^2 + \xi_y^2} \]

Based on the above two relations, equation (4.4) changes into

\[ G(x, y, \omega, \sigma, r, \theta) = \frac{1}{\sqrt{\pi \sigma}} e^{-\frac{1}{2} \left( \frac{R_1}{\sigma_x} \right)^2 + \left( \frac{R_2}{\sigma_y} \right)^2} e^{i\omega R_1} \]  \hspace{1cm} (4.5)

The feature extraction process can be written as a correlation of input image $I$ with Gabor kernel $G(r, \omega)$ with resolution $r$ and orientation $\omega$.

\[ IG_{(r, \omega)} = I \ast G_{(r, \omega)} \]  \hspace{1cm} (4.6)

The Gabor feature vector consists of all relevant information extracted from different frequencies and orientations of all areas. Hence, it is very useful for the breast cancer recognition.

### 4.5.3 Local Binary Pattern (LPB) based Feature Extraction Method

LBP method is very much interested in image processing and computer vision. It is one of the non parametric methods and its principle is comparing each pixel with its neighboring pixels in an efficient manner. LBP algorithm is developed for texture analysis and it is a powerful technique to describe the local structure (Ojala et al. 2002). It is used in many applications such as face image analysis, image and video retrieval (Huijsmans & Sebe 2003), object modelling, visual inspection, motion analysis, biomedical image analysis (Oliver et al. 2007), remote sensing.

The original LBP operator labels the pixels of an image with decimal numbers, called Local Binary Patterns or LBP codes, which encode the local structure around each pixel. It proceeds thus, as illustrated in Figure 4.3.
In Figure 4.3, the binary value is obtained using each pixel in the neighborhood pixels are subtracted with the center pixel. In the Figure 4.3, the original pixels are 5,9,1,4,4,6,7,2,3 and the center pixel is 4. The binary value is obtained using below equation:

\[
LBP_{M \times N} = \sum_{i=0}^{M} \sum_{j=0}^{N} P(i, j) - \text{center pixel} \quad (4.7)
\]

For example

\[
LBP_{M \times N} = 5 - 4 = 1 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 9 - 4 = 5 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 1 - 4 = -3 \{\text{binary value is 0}\}
\]

\[
LBP_{M \times N} = 4 - 4 = 0 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 4 - 4 = 0 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 6 - 4 = 2 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 7 - 4 = 3 \{\text{binary value is 1}\}
\]

\[
LBP_{M \times N} = 2 - 4 = -2 \{\text{binary value is 0}\}
\]

\[
LBP_{M \times N} = 3 - 4 = -1 \{\text{binary value is 0}\}
\]

In the above example obtained the result of two values, i.e., one value is positive and the other value is negative. So the positive value is encoded with binary value of 1 and the negative value is encoded with the binary value of 0.
Finally obtained the binary value by concatenating all these binary codes in a clockwise direction starting from the top-left one and its corresponding decimal value is used for labelling.

**Figure 4.3 A basic example of LBP structure**

Table 4.1 Binary value with their respective decimal number

<table>
<thead>
<tr>
<th>Binary values</th>
<th>Decimal number</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4.1 represents the binary numbers with their respective decimal number. The above table is used for the conversion of decimal into binary value. But in our condition is the conversion of a binary value into decimal numbers.
The main limitation of the basic LBP operator is that small $3 \times 3$ neighborhood cannot retrieve the specified features from the large scale structure. To extract the texture features with large scale structure was obtained by increasing the neighborhood sizes (Ojala et al. 2002).

A local neighborhood is defined as a set of sampling points evenly spaced on a circle which is centered at the pixel to be labelled, and the sampling points that do not fall within the pixels are interpolated using bilinear interpolation, thus allowing for any radius and any number of sampling points in the neighborhood. Figure 4.4 shows some examples of the extended LBP operator, where the notation $(S, r)$ denotes a neighborhood of $S$ sampling points on a circle of radius of $r$.

Let us consider the given pixel in a circle is represented as $x_c, y_c$ and the resulting LBP decimal form is obtained using below equation:

$$LBP_{S,r}(x_c, y_c) = \sum_{S=0}^{S-1} b(i_p - i_c)2^S$$  \hspace{1cm} (4.8)
In the above equation, $i_p$ and $i_c$ represents the gray values of the center pixels and $S$ represents the surrounding pixels in the circle with the radius $r$ and the function $b(x)$ is defined as:

$$b(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (4.9)$$

The above equation defines that LBP operator is invariant to monotonic gray scale transformations preserving pixel intensity order in the local neighborhoods. The texture features over the region is obtained by applying histogram to LBP labels.

The operator $LBP_{(S,r)}$ gives $2^S$ different output values for corresponding patterns $2^S$ different patterns formed by $S$ pixels in the neighborhood. After obtained the different output value, the image is rotated along the perimeter of the circle gives different LBP value. The below equation defines the rotation-invariant LBP is implemented to remove the rotation effect is given as:

$$LBP_{r}^{S,r} = \min\{ROR(LBP_{S,r}, i) | i = 0, 1, \ldots, S-1\} \quad (4.10)$$

The ROR $(y, i)$ defines the circular bit-wise right shift on the $S$ sampling $s$ bit number $y$ times. The above equation LBP operator quantifies the number of occurrences of invariant rotation corresponding to the features in the image.

![Image](image.png)

Figure 4.5(a) Gabor feature result image (b) LBP feature result image
Figure 4.5 provides the Gabor filter and LPB output, they both feature extraction techniques used for extraction of mammogram image features. Figure 4.5(a) shows the orientation of Gabor filter output, Figure 4.5(b) gives the binary values of the mammogram image using the LBP operator.

4.6 FEATURE SELECTION USING MFLDA

The reason of LDA is to find features which gather the features from the same class and also enlarges the margin of samples from different classes. Exactly, this objective can be achieved by maximizing the Fisher criterion as the ratio of the class scatter to the within class scatter. The feature extraction set from the $k$-th class $X^k = \{X^k_1, X^k_2, \cdots, X^k_{N_k}\}$, where $Nk \rightarrow$ the number of features in the $k$-th class, between class scatter and within the class scatter are calculated as

$$S_b = \frac{1}{N} \sum_{i=1}^{C} N_i (M^i - M)(M^i - M)^T$$ (4.11)

$$S_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{N_i} (M^i - M)(M^i - M)^T$$ (4.12)

Where $N = \sum_{i=1}^{C} N_i \rightarrow$ the total number of Samples, $C \rightarrow$ number of classes, $M^i = \frac{1}{N_i} \sum_{i=1}^{N_i} X_i \rightarrow$ the mean vector class $i$ and $M = \frac{1}{N} \sum_i^{C} \sum_{j=1}^{N_i} X^i_j \rightarrow$ the mean vector over the whole sample set. The MFLDA is used to find projective vector $W$ that increases the Fisher separation criterion $J$ is given below

$$J = \frac{|W^T S_b W|}{|W^T S_w W|}$$ (4.13)

The obtained solution of $W$ by solving the generalized eigen problem of $S_b W = \lambda S_w W$ with its leading eigen values
In standard Fisher’s LDA, extract features only one less than the number of classes, because of the rank between-class scattering matrix. It’s not a big problem in distance-based pure pixel classification methods. But, based on linear mixture model for the sub pixel classification, the number of features cannot be less than the number of classes, which is known as band number constraint (Ren & Chang 2004). To overcome this problem, modify Fisher’s LDA algorithm is proposed.

In this feature selection problem, the original feature set considered as \( \{f_1, f_2, \ldots, f_n\} \), where is the dimension of the original feature set, the purpose of feature selection is to select \( d \) features \( F^d = \{f_{v(1)}, f_{v(2)}, \ldots, f_{v(d)}\} \) from \( n \) original features which have the largest Fisher separation value. In this work, \((i)\) is the \( i \)-th feature index in the selected feature subset. Denoting between class scatter and within class scatter computed based on the selected feature set \( F^d \) as \( S_b(F^d), S_w(F^d) \) and the Fisher separation criterion as \( (F^d) \), the objective of selecting \( d \) features based on the Fisher separation criterion can be formulated as

\[
F^d = \arg \max_{F^d} J(F^d)
\]

(4.14)

Where \( F^d = J(f_{v(1)}, f_{v(2)}, \ldots, f_{v(d)}) \) is defined as

\[
(F^d) = \frac{|W^T S_b(F^d) W|}{|W^T S_w(F^d) W|}
\]

(4.15)

The solution to Eq. (4.15) is equivalent to solving the following generalized eigen value problem

\[
S_w^{-1} S_b W = \lambda W
\]

(4.16)

The features extracted by modified LDA are the eigen vectors of \( S_w^{-1} S_b \). Because, the total-class and within-classs cattering matrixes are all
calculated by training samples, providing number of training samples more than the number of bands, and the rank of $S_w^{-1}S_b$ will most likely be full rank. In this way, since this work has more features than the number of classes, the linear mixture problem can be solved.

4.7 SIMILARITY MEASURES WITH MAHALANOBIS DISTANCE

The similarity measurement is used to retrieve the images between the query image and the number of images in the given database. There are different types of similarity measure are available, in this chapter Mahalanobis distance is used for similarity measure.

Mahalanobis in 1936 introduced a distance called as Mahalanobis distance and it is a distance measure. It is based on correlations between variables by which different patterns can be determined and evaluated. It $c$ gauges similarity of an unknown sample set to a known one.

$$ D(I, J) = \sqrt{(F_I - F_J)^T C^{-1} (F_I - F_J)} $$ (4.16)

Where $D (I, J) \rightarrow$ the distance measure between the query images $I$ and the image $J$ in the database,$F (I) \rightarrow$ the number of pixels in bin $i$ of $I$. If feature dimensions are independent means, the Mahalanobis distance can be simplified. In that case, a variance of each feature component $c_i$ is needed

$$ D(I, J) = \sum_{i=1}^{N} \frac{(F_I - F_J)^2}{c_i} $$ (4.17)

4.8 FUZZY NEURAL NETWORK (FNN)

Nowadays Neural Network (FNN) based fuzzy logic system was introduced in the field of soft computing and pattern recognition. FNN is based
on the fusion of fuzzy and NN and it has an advantage of both fuzzy control system and NN system. In NN has an advantage of learning abilities, optimization abilities, and connectionist structure, whereas fuzzy control system has an advantage of human-like IF-THEN rule.

4.8.1 Basic Introduction about Algebraic Definition of Fuzzy Relation Neural Network

A fuzzy set in the universe of discourse U is defined by a set of ordered pairs and is given as

\[ A = \{(x, \mu_A)(x) \mid x \in U\} \] (4.18)

In the equation, \( \mu_A \) defines the membership function of A and is the grade of memberships of x in A. It represents the degree that x belongs to A and plays the role of the set of truth table (Zadeh 1965).

If any algebraic manipulation is too performed, then the set of truth table should be equipped with an algebraic structure. The below equation represents the complete residuated lattice is an algebra.

\[ L = \langle L, \land, \lor, \otimes, \to, 0, 1 \rangle \] (4.19)

Where

\[ \langle L, \land, \lor, \otimes, 0, 1 \rangle \] defines the complete lattice consists of least and greatest elements of ‘0’ and ‘1’. \( \land \) defines the infima of subsets of L exist and \( \lor \) defines the suprema of subset of L exists.

\[ \langle L, \otimes , 1 \rangle \] represents the commutative monoid, i.e., \( \otimes \) indicates the associative and commutative. If it is associative means \( x \otimes (y \otimes z) = (x \otimes y) \otimes z \), commutative \( (x \otimes y = y \otimes x) \) and the identity \( x \otimes 1 = x \) holds.
⊗, → satisfy the adjointness property, i.e., \( x \leq y \rightarrow z \) if \( x \otimes y \leq z \) holds.

### 4.8.2 Logical Operations and IF-THEN Rules

The important thing in fuzzy system is keeping the fuzzy values into 1 (True) or 0 (False). In order to find out the fuzzy logic operations, first have known about the crisp set. Elementary functions of the crisp set operations are union, intersection, and complement, which are correspond to OR, AND, and NOT operations.

Let us consider \( A \) and \( B \) are the two subsets of \( U \). The Union of \( A \) and \( B \) is represented as \( A \cup B \), in this union function, it contains either \( A \) or \( B \) and it is defined by below equation

\[
\mu_{A \cup B}(y) = 1, y \in A \text{ or } y \in B
\]  
(4.20)

The intersection of \( A \) and \( B \) is represented as \( A \cap B \), the intersection functions are described as it contains all the elements simultaneously and it is defined by

\[
\mu_{A \cap B}(y) = 1, y \in A \text{ and } y \in B
\]  
(4.21)

The complement of \( A \) is representing as \( \bar{A} \) and it describes that all elements in \( A \) is not present in the complement of \( A \) and it is defined as

\[
\mu_{\bar{A}}(y) = 1, y \notin A \\
\mu_{\bar{A}}(y) = 0, y \in A
\]  
(4.22)

To define the fuzzy logic operators, we have to find the corresponding operators that preserve the results of using AND, OR, and NOT operators. The answer is min, max, and complements operations. These operators are defined by below equations
Fuzzy union and fuzzy intersections are defined by below equations

\[
\mu_{A\cup B}(y) = \max[\mu_A(y), \mu_B(y)] \quad (4.23)
\]

\[
\mu_{A\cap B}(y) = \min[\mu_A(y), \mu_B(y)] \quad (4.24)
\]

\[
\mu_A(y) = 1 - \mu_A(y) \quad (4.25)
\]

Fuzzy union and fuzzy intersections are defined by below equations

\[
\mu_{A\cup B}(y) = \mu_A(y) + \mu_B(y) - \mu_A(y)\mu_B(y) \quad (4.26)
\]

\[
\mu_{A\cap B}(y) = \mu_A(y)\mu_B(y) \quad (4.27)
\]

Generally the intersection of two fuzzy sets A and B is specified by binary mapping T that aggregates two membership functions

\[
\mu_{A\cap B}(y) = T(\mu_A(y), \mu_B(y)) \quad (4.28)
\]

In the above equation, binary operator T represents the multiplication of fuzzy set. These fuzzy intersection operators are referred as T-norm (triangular norm) operators, and they meet the following basic requirements,

\[
\text{boundary: } T(0,0) = 0, T(a,1) = T(1,a) = a \quad (4.29)
\]

\[
\text{monotonicity: } T(a,b) \leq T(c,d) \text{ if } a \leq c \text{ and } b \leq d \quad (4.30)
\]

\[
\text{commutativity: } T(a,b) = T(b,a) \quad (4.31)
\]

\[
\text{associativity: } T(T(a,b),c) = T(T(a,c),b) \quad (4.32)
\]

Equation (4.29) ensures the correct generalization of crisp sets. The equation (4.30) defines that if the membership function of A and B is decreased, then the membership function of the intersection of A and B also decreased. The equation (4.31) defines that the operation is insensitive to the order in which fuzzy sets are combined. The equation (4.32) describes that
enable by taking different number of fuzzy sets with different number of grouping.

4.8.3 Introduction about Neuron Model in Neural Network

A single input neuron is illustrated in Figure 4.6, it shows that scalar input $p$ is multiplied with weight $w$ and is sent to the summer $\Sigma$. The other input $1$ is multiplied with bias $b$ and is sent to the summer $\Sigma$. The summer output is derived by $n$ and is referred as a net input and it passes into transfer function $f$, it gives the scalar neuron output $a$. The actual output of neuron model is based on the transfer function and their chosen parameter $n$.

Figure 4.6 Single-Input Neuron
### Table 4.2 Transfer Function

<table>
<thead>
<tr>
<th>Name</th>
<th>Input/output Relation</th>
<th>Icon</th>
<th>MATLAB FUNCTION</th>
</tr>
</thead>
</table>
| Hard Limit                | \[a = 0 \quad n < 0 \]
                          | \[a = 1 \quad n \geq 0\]                |        | Hardlim         |
| Symmetrical Hard Limit    | \[a = -1 \quad n < 0\]                       |        | Hardlims        |
|                           | \[a = +1 \quad n \geq 0\]                   |        |                 |
| Linear                    | \[a = n\]                                   |        | Purelin         |
| Saturating Linear         | \[a = 0 \quad n < 0\]                       |        | Satlin          |
|                           | \[a = n \quad 0 \leq n \leq 1\]             |        |                 |
|                           | \[a = 1 \quad n > 1\]                      |        |                 |
| Symmetric saturating Linear | \[a = -1 \quad n < -1\]                   |        | Satlins         |
|                           | \[a = n \quad -1 \leq n \leq 1\]           |        |                 |
|                           | \[a = 1 \quad n > 1\]                      |        |                 |
| Log-sigmoid               | \[a = \frac{1}{1 + e^{-n}}\]               |        | Logsis          |
| Hyperbolic Tangent Sigmoid | \[a = \frac{e^n - e^{-n}}{e^n + e^{-n}}\] |        | Tansig          |
| Positive Linear           | \[a = 0 \quad n < 0\]                       |        | Poslin          |
|                           | \[a = n \quad 0 \leq n\]                   |        |                 |
| Competitive               | \[a = 1 \quad \text{neuron with max n}\]  |        | Compet          |
|                           | \[a = 0 \quad \text{all other neurons}\]  |        |                 |

The transfer function has different types and it may be linear or a nonlinear function of net input. Table 4.2 provides the list of transfer function.
where the particular transfer function is chosen to satisfy some specifications of the problem that the neuron is attempting to solve. If a neuron has more than one input and is called multiple neuron and is illustrated in Figure 4.7.

The Figure 4.7 shows that the neuron has R inputs or individual inputs are \( p_1, p_2, \ldots, p_R \) and each input is weighted by \( w_{1,1}, w_{1,2}, \ldots, w_{1,R} \) of the weight matrix \( W \).

The neuron has a bias which is summed up with individual inputs and weighted inputs and produces net inputs are

\[
 n = w_{1,1}p_1 + w_{1,2}p_2 + \ldots + w_{1,R}p_R + b
\]  

(4.41)

The above equation can be written in the matrix form is given below:
\[ n = W_p + b \]  
(4.42)

In the above equation \( W \) represents the single neuron and it consists of only one row. The neuron output can be written as:

\[ a = f(W_p + b) \]  
(4.43)

Multiple-Input Neuron with abbreviated notation is illustrated in Figure 4.8

![Multiple-Input Neuron with Abbreviated Notation](image)

\[ a = f(W_p + b) \]

**Figure 4.8 Neuron with R Inputs, Abbreviated Notation**

Figure 4.8 illustrates that input R is a single vector consists of R rows and 1 column, whereas it enters into the weight matrix, where it has 1 rows and R columns. Here 1 as the input and is multiplied by the scalar bias b and it enters into net input n. The net input passes into transfer function f and produces the output as a scalar.

**4.8.4 Neuron Network Architecture**

A single-layer network of S neuron is illustrated in Figure 4.9. It shows that each element from the input vector is connected to the neuron. The
neuron consists of bias, transfer function and output function. Finally the output function provides the scalar output of $a$. The combination of input vectors and weight matrix is entering into the neuron is defined as:

$$
w = \begin{bmatrix}
w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\
w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\
\vdots & \vdots & \ddots & \vdots \\
w_{S,1} & w_{S,2} & \cdots & w_{S,R}
\end{bmatrix}
$$

(4.44)

Figure 4.9 Layer of S neurons
As noted previously, the row indices of the elements of the matrix indicate the destination neuron associated with that weight, while the column indices indicate the source of the input for that weight. Thus $w_{3,2}$, the indices say that this weight represents the connection to the third neuron from the second source. Let us consider the network with several layers and each layer has own bias, weight and transfer function and output vector. Figure 4.10 illustrates the Three-Layer Network Structure.

$$a^2 = f^1(W^1_p + b^1)$$
$$a^2 = f^2(W^2_p + b^2)$$
$$a^3 = f^3(W^3_p + b)$$

Figure 4.10 Three-Layer Network
4.8.5 General Architecture of Fuzzy Relation Neural Network (FRNN)

This architecture defines the design of FRNN for a complex fuzzy system. Let us consider the fuzzy system with multiple inputs and one output consists of the following fuzzy rules:

\[ R_1: \text{If } x_1 \text{ is } A_{11} \text{ and } x_2 \text{ is } A_{22} \text{ and } \ldots \text{ and } x_m \text{ is } A_{1m} \text{ then } y \text{ is } B_1 \]

\[ \text{else} \]

\[ \text{...} \]

\[ \text{else} \]

\[ R_n: \text{If } x_1 \text{ is } A_{n1} \text{ and } x_2 \text{ is } A_{n2} \text{ and } \ldots \text{ and } x_m \text{ is } A_{nm} \text{ then } y \text{ is } B_n \]

In the above rules \( A_{ij} \) and \( B_i \) are the fuzzy sets in \( U \subset \mathbb{R} \) and \( V \subset \mathbb{R} \) respectively and \( x_i \) and \( y \) are linguistic variables.

4.9 Classification Using FNN

In this FNN, use the simplified case of only two classes of normal and abnormal. Each class grouping of the FNN works the same way. More classes yield fuzzy memberships in more classes from which to select a maximum value winner at the final output node. The proposed system has \( N \) features in the input exemplar feature vectors. Here there are two classes in the training exemplar data \( \{(x^{(q)}, t^{(q)}): q = 1, \ldots, Q\} \), i.e., the \( t^{(q)} \) have two unique labels, so the proposed system use \( K = 2 \) class groups of hidden nodes where each such node represents a Gaussian function centered on an exemplar feature vector that has an associated label. Each Gaussian in a class group has a different center, but the same label. The first group of hidden nodes is considered for Class 1 in Figure 4.11.
In large number of cases, $K_p$ of feature vectors in Class $p$ ($p = 1, 2$ here), therefore eliminate those feature vectors that are close to another feature vector with the same label. It diminishes the number of centers, and thus Gaussians (attributes), that represent each Class $p$. Then, fuzzy truth that input vector $x$ is in the same class as $x^{(q)}$ is given by the Gaussian Fuzzy Set Membership Function (FSMF) centered on $x^{(q)}$. The $q^{th}$ Gaussian FSMF is the function.

$$x \rightarrow g(x, x^{(q)}) = \exp\left\{-\|x - x^{(q)}\|^2/(2\sigma^2)\right\}$$ \hspace{1cm} (4.45)

Where $\sigma$ can be taken to be one-half of the average distance between all pairs.

The entire fuzzy truths for the centers of Gaussians in Class 1 (Figure 4.11) are feed from their Gaussian nodes to the maximizer node of the

![Figure 4.11 FNN architecture](image-url)
Class 1 fuzzy truth and they perform as a fuzzy OR node in selecting the agent (representative) center and fuzzy truth that x belongs to several $x^{(k)}$ for Class 1. The class 1 maximum fuzzy truth for x is now sent to the final output maximizer node as the Class 1 agent. The final output maximizer node also receives the Class 2 maximum fuzzy truth representative that x belongs to Class 2 and establishes the maximum of these fuzzy truths, and then the class that sent it is the output of classification. Thus the input x belongs to the output class determined by the label of the output Gaussian center vector. Step by step process of FNN is given below.

**Step 1:** Initially, the input data is read and it contains the number of features N, the number of feature vectors Q, the number K of classes, the dimension J of the labels, all Q feature vectors and all Q labels.

**Step 2:** Then minimal distance $D_{\text{min}}$ is calculated over all feature vector pairs

Mentioned $F = D_{\text{min}}/2$

Mentioned $G = Q$ //Starting no. Gaussian centers

**Step 3:** Two exemplar vectors of min. is founded distance d with indices $k_1$ and $k_2$

If $d < (\frac{1}{2})D_{\text{min}}$ //If vectors are close and

If label[$k_1$] = label[$k_2$]// have same label

Eliminate Gaussian center of $k_2$

$G = G - 1$

Go to **Step 2**

**Step 4:** Next unknown input x to FNN to be classified by...
For \( k = 1 \) to \( G \) do

\[
\text{Compute } g[k] = \exp\left\{-\frac{\|x - x^{(k)}\|^2}{2\sigma^2}\right\}
\]

Find maximum \( g[k^*] \), over \( k = 1 \ldots G \)

The maximum Output \( x \), \( \text{label}[k^*] \) where \( \text{label}[k^*] \) is class of \( x \)

**Step 5:** If classifying are done for all inputs means terminate the function.

Else

Go to Step 4.

### 4.10 SIMULATION RESULTS

In this section, the proposed MFLDA-FNN classification technique performance is evaluated. MATLAB 12 is employed to simulate planned algorithm. The Figure 4.12 illustrates the retrieval output of the MFLDA FNN classifier. The performance of the proposed approach of MFLDA-FNN is compared with hybrid classifier of MLFFB-ANN.

![Figure 4.12 Retrieval output using MFLDA FNN classifier](image-url)
4.10.1 Performance Analysis Criteria

For effective CBIR system, the paper estimates the retrieval performance of precision, accuracy and recall outlined as follows. The method for evaluating performance inherently considers every image as a query, measures each retrieval outcome and reports overall average

\[
\text{Precision} = \frac{\text{Number of Relevant retrieved images}}{\text{Total number of retrieved images}} \times 100
\]

\[
\text{Recall} = \frac{\text{Number of Relevant retrieved images}}{\text{Total number of relevant images in the database}} \times 100
\]

\[
\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}
\]

Where, TP \rightarrow true positive, FP \rightarrow false positive, FN \rightarrow false negative and TN \rightarrow true negative.

- True Positive refers to the correctly identified abnormal image,
- True Negative correctly identified the normal image as a normal image
- False Positive wrongly identified a normal image as an abnormal image
- False Negative refers wrongly identified abnormal image as normal image.
Figure 4.13 Comparison of Precision between Hybrid classifier and Fuzzy Neural Network classifier

Figure 4.14 Comparison of Recall and Accuracy between Hybrid classifier and Fuzzy Neural Network classifier
Figure 4.15 Comparison of Execution time between Hybrid classifier and Fuzzy Neural Network classifier

Figure 4.13-4.15 depicts the recall, precision, accuracy and execution time comparison between proposed fuzzy neural network and hybrid classifier. From this figure analyzed that proposed MFLDA-FFN classifier provides better performance in terms of precision, recall and accuracy compared to proposed hybrid classifier because of the following objectives:

In MFLDA-FFN classifier approaches consist of preprocessing and segmentation, it helps to predict the best results because preprocessing is a vital role in image processing and it removes the unwanted noises. In the case of segmentation, the mass and microcalcification part of the mammogram image is detected.

The features are extracted from the segmented part, whereas in hybrid classifier of MLFFB-ANN, features are extracted from the full mammogram image.
Basically, artificial neural network classifier is based on the number of hidden rules and learning process of the network is too long. So this condition will affect the hybrid classifier whereas fuzzy classifier is based on membership function and if-then rules, the processing speed of fuzzy classifier is very fast.

4.11 RESULTS AND DISCUSSION

In this section, overall performance of the proposed research work is carried out. The overall comparison is performed for precision, recall, accuracy and execution time of Bayesian Network, SVM, proposed hybrid classifier and proposed MFLDA-FNN classifier.

<table>
<thead>
<tr>
<th>Methods, Metrics</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>Execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian Network</td>
<td>67.73</td>
<td>70</td>
<td>72</td>
<td>4.1</td>
</tr>
<tr>
<td>SVM</td>
<td>74.45</td>
<td>79.82</td>
<td>82.25</td>
<td>3.5</td>
</tr>
<tr>
<td>Hybrid classifier</td>
<td>82.12</td>
<td>82.34</td>
<td>87.85</td>
<td>2.8</td>
</tr>
<tr>
<td>MFLDA-FNN classifier</td>
<td>90.01</td>
<td>91.12</td>
<td>96.23</td>
<td>2.7</td>
</tr>
</tbody>
</table>

Table 4.3 gives the comparison of overall performance metrics for Bayesian network, SVM, hybrid classifier and MFLDA-FNN classifier. The overall performance is also illustrated in Figure 4.16-4.18.

The precision comparison of four classifiers: The Precision percentage of SVM classifier is 6.72% increased compared to Bayesian network and Hybrid classifier is 7.67% increased compared to the SVM classifier. The Precision percentage of the FNN classifier is 7.89 % increased compared to Hybrid classifier.
The Recall comparison of four classifiers: The Recall percentage of SVM classifier is 9.82% increased compared to Bayesian network and Hybrid classifier is 2.52% increased compared to the SVM classifier. The Recall percentage the FNN classifier is 8.78% increased compared to Hybrid classifier.

The Accuracy comparison of four classifiers: The Accuracy percentage of SVM classifier is 10.25% increased compared to Bayesian network and Hybrid classifier is 5.60% increased compared to the SVM classifier. The Accuracy percentage of the FNN classifier is 8.38% increased compared to Hybrid classifier.

The Execution time comparison of four classifiers: The Execution time of SVM classifier is 0.6ms decreased compared to Bayesian network and Hybrid classifier is 0.7ms decreased compared to the SVM classifier. The Accuracy percentage of the FNN classifier is 0.1ms decreased compared to Hybrid classifier.

So from this analysis, proposed FNN classifier has obtained higher percentage in Precision, Recall and Accuracy compared to other approaches of proposed Hybrid classifier, SVM and Bayesian network. The proposed FNN classifier requires less Execution time compared to other approaches of proposed Hybrid classifier, SVM and Bayesian network.
Figure 4.16 Comparison of Recall and Precision between four classifiers

Figure 4.17 Comparison of Accuracy between four classifiers
4.12 SUMMARY

In this chapter, CBIR mammogram image is obtained through FNN classifier. This chapter provides the overall comparison of proposed work where FNN classifier provides better results because of the addition of preprocessing and segmentation algorithm along with MFLDA feature selection. The overall result of this research is discussed in this chapter.