CHAPTER 5

FEATURE EXTRACTION, FEATURE SELECTION AND CLASSIFICATION

5.1 Feature Extraction

The texture of mammogram images refers to the appearance, structure and arrangement of the parts of an object within the image. A feature value is a real number, which encodes some discriminatory information about a property of an object. It may not always be obvious what type of information or feature, is useful for a particular detection task. Additionally, there are potentially many ways to describe a particular object characteristic such as texture. It may not be obvious which method of computation extracts the most useful discriminatory information[92].

The performance of the classifiers, i.e. the ability to assign the unknown object to the correct class, is directly dependent on the features selected that represent the object of description. Texture is one of the important characteristics used in identifying an object. The texture coarseness or fineness of an image can be interpreted as the distribution of the elements in the matrix. This section presents SRDM based texture-analysis method. The segmented mammogram images from MRF-ABC approach are considered inputs for feature extraction methods[93].

5.1.1 Surrounding Region Dependency Matrix

The SRDM is based on a second-order histogram in two surrounding regions. Let us consider two rectangular windows centered on a current pixel (x, y). \( R_1 \) and \( R_2 \) are the inner surrounding region and the outer surrounding region, respectively. An image is transformed into a surrounding region-dependence matrix and the features are extracted for this matrix. In this method, two different regions of size 5x5 and 7x7 are selected for each pixel. And the number of pixels greater than the selected threshold value (q) are counted in each region. Let as assume \( m \) and \( n \) are the count from each region, the element in the surrounding region dependence matrix \( M(m,n) \) is incremented by 1. This
procedure is repeated for all the image pixels and the matrix gets updated. Normally a set of 5 different thresholds is selected such as 50, 60, 70, 80 and 100. A separate matrix is generated for each threshold value. Figure 5.1 Shows the Algorithm of SRDM The algorithm is as follows:

Algorithm: Surrounding Region Dependence Method

\[
\begin{align*}
M_{ij} & \leftarrow \text{Original Image} \\
Th & \leftarrow \text{peak threshold value from the histogram of the mammogram.} \\
P & \leftarrow \text{Select a pixel, whose intensity value is greater than the selected threshold value.} \\
R_1, R_2 & \leftarrow \text{sub regions of size 5\text{*}5 and 7\text{*}7} \\
M(q) & \leftarrow \left[ \alpha (i,j) \right], 0 < i < m, 0 < j < n,
\end{align*}
\]

where \( q \) is the threshold value and \( m, n \) are the number of pixels in the regions \( R_1 \) and \( R_2 \).

\[
\begin{align*}
\alpha(i,j) & \leftarrow \# \{ (x,y) \mid cR_1 (x,y) = i \text{ and } cR_2 (x,y) = j, (x,y) \in L_x \cdot L_y \},
\end{align*}
\]

where,

\[
\begin{align*}
cR_1 (x,y) & \leftarrow \# \{ (k,l) \mid (k,l) \in R_1 \text{ and } [ S(x,y) - S(k,l) ] > q \}, \\
cR_2 (x,y) & \leftarrow \# \{ (k,l) \mid (k,l) \in R_2 \text{ and } [ S(x,y) - S(k,l) ] > q \},
\end{align*}
\]

where \# denotes the number of elements in the set,

\( S(x,y) \) is the intensity value of the current pixel \( (x,y) \).

From this surrounding region dependence matrix the following features are extracted.

Horizontal Weighted Sum (HWS) \( \leftarrow \Sigma \Sigma \alpha^2 r(i,j) \)

Vertical Weighted Sum (VWS) \( \leftarrow 1/N \Sigma \Sigma \alpha^2 r(i,j) \)
Diagonal Weighted Sum (DWS) ← \( \frac{1}{N} \sum_{m+n} k^2 \left[ \sum_{ij} i^2 r(i,j) \right] \)

Grid Weighted Sum (GWS) ← \( \frac{1}{N} \sum_{ij} ij r(i,j) \),

N ← total sum of elements, and

\( r(i,j) ← \frac{1}{\alpha(i,j)} \).

Figure 5.1: Algorithm of SRDM

5.1.2 The Textural Haralick Features

A smooth region in a texture image consists of pixels having more or less equal gray levels. Thus, peaks along the diagonal of the distribution matrices represent smooth regions, while off-diagonal peaks may correspond to regions having a specific texture. They also correspond to edge regions, provided that the edges are sharp enough.

The texture analysis matrix itself does not directly provide a single feature that may be used for texture discrimination. Instead, the matrix can be used as a representation scheme for the texture image and the features are computed.

The features based on the distribution matrices should therefore capture some characteristics of textures such as homogeneity, coarseness, periodicity and others [94; 95; 96]. Haralick et al. have suggested 14 texture features, which can be put into four groups:

a. Features that express visual texture characteristics: are Angular Second Moment (ASM), Contrast (CON), and Correlation (COR).

b. Features that are based on statistics: are Variance (VAR), Inverse Difference Moment (IDM), Sum Average (SA), Sum Variance (SV) and Difference Variance (DV).

c. Features that are based on information theory: are Entropy (ENT), Sum Entropy (SENT) and Difference Entropy (DENT).
d. Features that are based on information measures of correlation: are Information Measures of Correlation (IMC1, IMC2) and Maximal Correlation Coefficient (MCC).

5.1.3 Results

The following table shows the extracted feature values based on SRDM from the segmentation image. The experiments have been carried out for all the 322 images and the results for the first thirty nine images are listed in the following table. Table 5.1 Shows the SRDM based feature Extraction with MRF - ABC method.

Table 5.1 SRDM based feature extraction with MRF-ABC method

<table>
<thead>
<tr>
<th>Image Ref.</th>
<th>mdb030</th>
<th>mdb031</th>
<th>mdb032</th>
<th>mdb033</th>
<th>mdb034</th>
<th>mdb035</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASM</td>
<td>0.07491</td>
<td>0.07886</td>
<td>0.07958</td>
<td>0.07624</td>
<td>0.08626</td>
<td>0.07058</td>
</tr>
<tr>
<td>CON</td>
<td>0.05769</td>
<td>0.05449</td>
<td>0.05669</td>
<td>0.0553</td>
<td>0.05923</td>
<td>0.05509</td>
</tr>
<tr>
<td>COR</td>
<td>-0.0312</td>
<td>-0.0327</td>
<td>-0.0322</td>
<td>-0.0326</td>
<td>-0.0345</td>
<td>-0.0299</td>
</tr>
<tr>
<td>VAR</td>
<td>-0.0287</td>
<td>-0.0252</td>
<td>-0.0274</td>
<td>-0.0259</td>
<td>-0.0305</td>
<td>-0.0255</td>
</tr>
<tr>
<td>IDM</td>
<td>-0.1587</td>
<td>-0.1717</td>
<td>-0.1646</td>
<td>-0.1732</td>
<td>-0.1818</td>
<td>-0.1532</td>
</tr>
<tr>
<td>SA</td>
<td>0.06134</td>
<td>0.05776</td>
<td>0.06023</td>
<td>0.05867</td>
<td>0.06309</td>
<td>0.05842</td>
</tr>
<tr>
<td>SV</td>
<td>-0.0633</td>
<td>-0.0591</td>
<td>-0.062</td>
<td>-0.0602</td>
<td>-0.0653</td>
<td>-0.0599</td>
</tr>
<tr>
<td>DV</td>
<td>0.05158</td>
<td>0.04903</td>
<td>0.0508</td>
<td>0.04969</td>
<td>0.05281</td>
<td>0.0495</td>
</tr>
<tr>
<td>ENT</td>
<td>-0.1028</td>
<td>-0.1076</td>
<td>-0.1069</td>
<td>-0.1067</td>
<td>-0.1144</td>
<td>-0.0984</td>
</tr>
<tr>
<td>SENT</td>
<td>0.08906</td>
<td>0.09461</td>
<td>0.09428</td>
<td>0.09355</td>
<td>0.10393</td>
<td>0.08325</td>
</tr>
<tr>
<td>DENT</td>
<td>-0.0634</td>
<td>-0.0592</td>
<td>-0.062</td>
<td>-0.0602</td>
<td>-0.0654</td>
<td>-0.0599</td>
</tr>
<tr>
<td>IMC1</td>
<td>-0.2079</td>
<td>-0.2902</td>
<td>65535</td>
<td>65535</td>
<td>-0.1536</td>
<td>-0.2417</td>
</tr>
<tr>
<td>IMC2</td>
<td>-0.1101</td>
<td>-0.1152</td>
<td>-0.1138</td>
<td>-0.115</td>
<td>-0.1214</td>
<td>-0.1058</td>
</tr>
<tr>
<td>MCC</td>
<td>0.38907</td>
<td>0.3666</td>
<td>0.43204</td>
<td>0.30888</td>
<td>0.40653</td>
<td>0.37882</td>
</tr>
</tbody>
</table>
5.2 Feature Selection

Feature selection is focused on many areas, especially in artificial intelligence, medical image processing, Data Mining and pattern recognition. The main issues in developing feature selection, also known as dimensionality reduction techniques, are choosing a small feature set in order to reduce the space and running time of a system, as well as achieving an acceptably high recognition rate.

This has led to the development of a variety of techniques for selecting an optimal subset of features from a larger set of possible features. These feature selection techniques fall into two main categories. In the first approach, problem specific strategies are developed based on the domain knowledge in order to reduce the number of features used to a manageable size. The second approach is used when the domain knowledge is unavailable or expensive to exploit. In this case, generic heuristics and greedy algorithms are applied to select a subset “d” of the available “m” features[97].

One approach would be to extract and use a large set of potentially useful features. Some features may contain irrelevant or redundant information, which may have detrimental effects on classifier performance. The phenomenon is known as the “curse of dimensionality”. Ideally, classification should be based on a small number of significant features that effectively characterize the input data. The goal of feature selection is to find an optimal subset of d features for a particular detection task given a full set of D features, where $d \leq D$. Thus, a method of evaluating the goodness of a set of features is required. The misclassification error rate of the classifier being utilized is a good evaluation criterion.

The only way to guarantee the selection of an optimal feature vector is an exhaustive search of all possible subsets of features. The problem can be formatted as a search of a directed graph. The size of the power set (the set of all subsets) of D features
is \(2^D\). As a result, a number of sub optimal search techniques are often utilized for feature selection.

There are many reasons for using feature selection technique to reduce the number of features.

§ Satisfying the general goal of maximizing the accuracy of the classifier while minimizing the associated measurement costs.

§ Improving accuracy by reducing irrelevant and possibly redundant features.

§ Reducing the complexity and the associated computational cost.

§ Reducing the amount of data needed for the training.

§ Improving the chances that a solution will be both understandable and practical.

Feature selection is meant here to refer to the problem of dimensionality reduction of data, which initially contain a high number of features. One hopes to choose optimal subsets of the original features that still contain the information essential for the classification task, while reducing the computational burden imposed by using many features.

In this chapter, three different algorithms such as Genetic Algorithm, Particle Swarm Algorithm and Artificial Bee Colony Optimization algorithm have been proposed for feature selection. A comparative analysis has also been made with emphasis on mammograms to select the best feature set.

5.2.1 Feature Selection using Genetic Algorithm

In this chapter, SRDM textural matrices are created for each mammogram image. For each defined distance and direction the Haralick features are extracted for all the 322-mammogram images. The features are grouped into four categories as discussed in
section 5.6. A single feature value for all the images is considered the initial population string for Genetic Algorithm[98].

An optimum value is computed for each individual feature set. In a group, the optimum value from each individual set is compared; the feature set, which selects the optimum among other features in the same group, is selected for classification. Like this, for every group an optimum feature is selected. Finally, the algorithm selects the four optimum features from the set of fourteen features. Only the selected features are used for classification.

From the population of the individual feature set, the fitness value is calculated for each feature using the fitness function (1/(1+Pi)), where Pi is the feature value. Then the probability of each feature value is calculated. And the cumulative probability is compared for each feature value. Then a random number between zero and one is generated for each feature value. If the cumulative probability value for a feature is higher than the random number, then the feature selection count is incremented by one.

This procedure is repeated for the number of times equal to the population size. Next, the population is reproduced with the feature values whose selection count is greater than zero. Each feature is copied into the reproduced population corresponding to the number of times it has been selected. For example, if a selection count for a feature is two, then that feature will be copied two times in the reproduced population.

After reproduction the single point crossover operation is performed on population strings depending upon the crossover probability (Pc). The Pc ranges from zero and one. In the single point crossover operation, initially the pair of population strings is randomly selected for matting. And a random bit position is selected for each pair.
The bits available after the random bit position are exchanged between the population strings in the pair. Thus the matting is performed to create another population set with different values. Next, the mutation operator is applied to the matted population strings depending upon the mutation probability ($P_m$), where $P_m$ is a small number ranging from zero and one. In mutation, a random bit position is selected from the population. If the bit value is one in that position it is flipped to zero; else it is changed to one. The population now contains a new set of strings for the next population. The next iteration is performed with the new population of strings. This procedure is repeated 30-200 times. Finally the maximum value from the recent population is returned as optimum value of the feature set. The features selected from this algorithm are ASM, VAR, ENT and IMC2. Figure 5.2 shows the algorithm of feature selection using GA.

Step 1. $P_i$ ← feature values.
Step 2. $F_i = 1/ (1 + P_i)$, { Fitness values}
Step 3. Calculate the probability and cumulative probability, $CP$
Step 4. Reproduction
   a. $r$ ← random()
   b. if ($CP_i > r$) count=count + 1 for $CP_i$
   c. Repeat the steps (a) and (b) for all the population strings.
   d. If count=0, then delete that $P_i$
   e. Reproduce the population by copying the selected strings with the corresponding number of times it has been selected.
Step 5. Crossover
   a. $r$ ← random()
   b. S select the pair of strings for matting randomly
   c. if ($r > P_c$) $k$← random bit position
d. interchange the bits after $k^{th}$ position in parent1 and parent2
e. repeat this step for all the pairs.

Step 6. Mutation

a. $r \leftarrow$ random()
b. if ($r >= P_m$), $k$ $\leftarrow$ random bit position
c. complement the value of the $k^{th}$ bit
d. repeat this steps for all the strings.

Step 7. $P_{\text{new}}$ $\leftarrow$ population after reproduction, crossover and mutation.
Step 8. $P_{t}$ $\leftarrow$ $P_{\text{new}}$

Figure 5.2: Algorithm of feature selection using GA

5.2.2 Feature Selection using Particle Swarm Optimization (PSO)

The optimum feature is selected from each group and only those selected features are further used in the classification. As a result, the ASM, IDM, ENT and IMC1 are the selected features from this algorithm.

PSO is initialized with a population of random solutions, called ‘particles’. Each particle is treated as a point in an S-dimensional space. The $i^{th}$ particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iS})$. The best previous position (pbest, the position giving the best fitness value) of any particle is recorded and represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iS})$. The index of the best particle among all the particles in the population is represented by the symbol ‘gbest’. The rate of the position change (velocity) for particle $i$ is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iS})$. 
The particles are manipulated according to the following equation:

\[ v_{id} = w^*v_{id} + c_1*r\text{rand}(\_)*\left(p_{id} - x_{id}\right) + c_2*r\text{rand}(\_)*\left(g_{id} - x_{id}\right) - 1 \]

\[ x_{id} = x_{id} + v_{id} \]

Where \( d = 1,2,..., S \), \( w \) is the inertia weight, it is a positive linear function of time changing according to the generation iteration. Suitable selection of the inertia weight provides a balance between global and local exploration, and results in fewer iterations on an average to find a sufficiently optimal solution. The acceleration constants \( c_1 \) and \( c_2 \) in equation (1) represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past, target regions. \( \text{rand}() \) and \( \text{Rand}() \) are two random functions in the range.

Particles’ velocities on each dimension are limited to a maximum velocity, \( V_{\text{max}} \). It determines how large steps through the solution space each particle is allowed to take. If \( V_{\text{max}} \) is too small, particles may not explore sufficiently beyond locally good regions. They could become trapped in local optima. On the other hand, if \( V_{\text{max}} \) is too high particles might fly past good solutions. The first part of equation (1) provides the “flying particles” with a degree of memory capability allowing the exploration of new search space areas. The second part is the “cognition” part, which represents the private thinking of the particle itself. The third part is the “social” part, which represents the collaboration among the particles. Equation (1) is used to calculate the particle’s new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group’s best experience. Then the particle flies toward a new position according to equation (2). The performance of each particle is measured according to a pre-defined fitness function.
Algorithm PSO

Input:

m: the swarm size; c1, c2: positive acceleration constants; w: inertia weight

MaxV: maximum velocity of particles

MaxGen: maximum generation

MaxFit: maximum fitness value

Output:

Pgbest: Global best position

Begin

Swarms \{xid, vid\} = Generate(m); /* Initialize a population of particles with random positions and velocities on S dimensions*/

\[ Pbest(i) = 0; i = 1, \ldots, m, d = 1, \ldots, S \]

Gbest = 0;

Iter = 0;

While(Iter < MaxGen and Gbest < MaxFit) {

For(every particle i) {

Fitness(i) = Evaluate(i);

IF(Fitness(i) > Pbest(i))

\{ Pbest(i) = Fitness(i); p = x id; d id = 1, \ldots, S \}

IF(Fitness(i) > Gbest)

\{ Gbest = Fitness(i); gbest = i; \}

}

For(every particle i) {

For(every d ){

}

}
\[ v_{id} = w \cdot v_{id} + c_1 \cdot \text{rand}() \cdot (p_{id} - x_{id}) + c_2 \cdot \text{Rand}() \cdot (p_{gd} - x_{id}) \]

IF( \( v_{id} > \text{MaxV} \) ) \{ \( v_{id} = \text{MaxV} \) \}

IF( \( v_{id} < -\text{MaxV} \) ) \{ \( v_{id} = -\text{MaxV} \) \}

\[ x_{id} = x_{id} + v_{id} \]

\( \text{Iter} = \text{Iter} + 1; \)

/* \text{rand}() and \text{Rand}() are two random functions in the range [0,1]*/

Return \( P_{\text{gbest}} \)

End

---

**Figure 5.3: Feature Selection Using Particle Swarm Optimization**

In this section, the idea of PSO for the optimal feature selection problem is considered. A large feature space full of feature subsets can be considered. Each feature subset can be seen as a point or position in such a space. If there are \( N \) total features, then there will be \( 2^N \) kinds of subset, different from each other in the length and features contained in each subset. The optimal position is the subset with least length and highest classification quality. Now we put a particle swarm into this feature space, each particle takes one position. The particles fly in this space, their goal is to fly to the best position. Over time, they change their position, communicate with each other, and search around the local best and global best position. Eventually, they should converge on good, possibly optimal, positions. It is this exploration ability of particle swarms that should better equip it to perform feature selection and discover optimal subsets. Figure 5.3 Shows the Feature Selection Using Particle Swarm Optimization.
As a result the ASM, IDM, ENT and IMC2 are the selected features from this algorithm.

5.2.3 Feature Selection using Artificial Bee Colony

The proposed feature selection system is derived from ABC algorithms. This method has been developed using principles of the ABC and mutual information (MI). The ABC algorithm is an optimization algorithm that uses the behavior of the bees searching for food. A bee colony is an organized team work system where each bee contributes significant information to the system. There are three types of worker bees which involve in collecting nectar viz. employed bees, onlooker bees and scout bees.

The ABC algorithm considers the position of food source as the possible solution of the optimization problem and the food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. The initial population of N solutions is randomly generated. Each solution is a D dimensional vector where D is the number of parameters to be optimized. They are relevant and redundant in this case. The population of solutions is subject to repeated search processes by the employed bees, onlooker bees and scout bees. A solution is randomly chosen and compared with the current solution. The objective function used here will be the mRMR function. The fitness function of each solution is given by

\[ f_{t_i} = \left(1 + \frac{1}{f_{t_i}}\right) \]

where \( f(i) \) is the objective function of the \( ith \) solution. If the fitness function of the new chosen solution is greater than the existing one, then the new solution is memorized and the old one is discarded. The employed bees share the information i.e, fitness value of the solutions in their memory with the onlooker bees. The probability of each solution based on its fitness, is calculated by
where \( \text{fitness}_i \) is the fitness value of the solution \( i \) and \( N \) is the number of solutions in the population. Candidate solutions are produced using the formula

\[
\nu_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})
\]

where \( k \in \{1, 2, \ldots, N\} \) and \( j \in \{1, 2, \ldots, D\} \) and \( \phi_{ij} \) is a random number ranging between -1 and 1. This ensures that values generated are different from those already existing. And also the newly generated solutions lie within the defined boundary. The parameter exceeding its limit is set to its limit value.

The performance of each candidate solution is compared with that of the existing solution. If the new solution has equal or better fitness value than the old solution, the old one is discarded with the new one occupying its position. Else, the old one is retained. In other words, a greedy selection mechanism is used for the selection process. If population is within the predefined number of cycles i.e. limit then, that population is abandoned and replaced with a new population. ABC algorithm is used here to optimize the redundancy and relevance parameters of mRMR function. The mRMR method proposed to uses the principle of mutual Information[99]. The mutual information between two variables \( A \) and \( B \) can be defined as

\[
I(A/B) = \log \left( \frac{P(A/B)}{P(A)} \right)
\]

Maximum Relevance orders features based on the mutual information between individual features \( x_i \) and target class \( h \) is such that the feature with the highest mutual information is the most relevant feature. The relationship is expressed as follows:

\[
\max V_p V_l=1/|s| \sum_{ij} I(h,l)
\]
Max Relevance often shows a high inter-dependence among the features. When two features are highly dependent on one another, the class-discriminative power of these two features would not change much if either one of them is to be removed and if not removed they become redundant as they convey the same characteristics. The minimal redundancy condition can be added to select mutually exclusive features of the dataset. The following relationship helps establish the minimum redundancy measure.

\[
\text{minWp WI}=\frac{1}{|\mathcal{I}|} \text{I}(h,l)
\]

The criterion combining the above two parameters is called “minimal-redundancy-maximal-relevance”. It is seen that the two measures could be used together to form two combinations for the purpose of improving the feature selection process. The two combinations considered are as follows:

\[
\begin{align*}
\max (v_l-w_l)
\end{align*}
\]

\[
\begin{align*}
\max (v_l/w_l)
\end{align*}
\]

Here Eq. \(\max (v_l-w_l)\), forms MID: Mutual Information Difference criterion and Eq. \(\max (v_l/w_l)\) forms MIQ: Mutual Information Quotient criterion. It has been observed that MID gives a better performance when compared to MIQ. This is found to be the case in the present study, as well. Redundancy is often a matter of concern when dealing with large datasets. It has been noticed that redundancy caused a negative effect on the accuracy of the classifying system. But it cannot be presumed that the relevance factor only facilitated the increase in accuracy. The conditions are seen to be purely situational. That is, depending on the dataset under study, either of the two, relevance or redundancy may drastically affect the percentage of accuracy.

The following expression defines the proposed optimization criterion \(\max (aV_l-bW_l)\) given to relevance and redundancy for selecting the optimal feature set, \(V_l\) is
relevance and WI is redundancy. The value of the constants a and b are arrived at from the ABC algorithm. This algorithm is noticed to be applicable only for discrete datasets. And so, when continuous datasets are to be analyzed, discretization has to be done. Discretization comes with the disadvantage of loss of data which will further reduce the accuracy[100]. This can be alternated by scaling the data and employing logarithmic functions. Thus, the relationship is modified for continuous datasets as follows:

$$\max(ax\log(Vl) - bx\log(Wl))$$

The output of the proposed feature selection mechanism is the set of features in the decreasing order of importance. As in the ABC algorithm, the optimum feature is selected from each group and only those selected features are further used in the classification. As a result the ASM, IDM, ENT and IMC2 are the selected features from this algorithm.

### 5.2.4 Experiments and Results

Table 6.1 shows the list of features selected by the feature selection algorithms such as Genetic Algorithm, Particle Swarm Optimization and Artificial Bee Colony. The combined features such as ASM, IDM, ENT, IMC2 are given as input to the BPN classifier.

### 5.3 Classification

Classification of objects is an important area of research and of practical applications in a variety of fields, including pattern recognition, artificial intelligence and vision analysis. Classifier design can be performed with labeled or unlabeled data.

Neural Networks (NN) can learn various tasks from training examples:such as classifying phenomena and modelling nonlinear relationships. However, the primary features that are of concern in the design of the networks are problem specific. Despite
the availability of some guidelines, it would be helpful to have a computational procedure in this aspect, especially for the optimum design of an NN. The gradient descent algorithms have encountered difficulties in learning the topology of the networks whose weights they optimize[101].

Artificial Neural Networks (ANN) is the networks of interconnected simple units that are based on a simplified model of the brain. ANN is a useful learning tool because they enable one to compute results quickly interpolating data well. There are two main types of ANN, feed forward networks and recurrent networks. Perceptron is a special case of feed forward neural networks with only input and output nodes.

Three main perceptron learning algorithms are covered: mistake bound perceptron algorithm, perceptron training rule and the Delta rule. The Delta rule uses gradient descent, which makes it easy to compute what changes are needed to optimize the network. The Back Propagation learning algorithm is widely used for multi-layer feed forward network. Bayesian learning is based on statistics and knowledge of prior statistics to classify or predict. The Bayes theorem is central to Bayesian learning[102].

5.3.1 Neural Networks based classification

The classifier employed in this thesis is a three layer Back Propagation Neural network. The Back Propagation Neural network optimizes the net for correct responses to the training input data set. More than one hidden layer may be beneficial for some applications, but one hidden layer is sufficient if enough hidden neurons are used[103 ; 104 ; 105].
5.3.2 BPN hybrid with Ant Colony Optimization Algorithm (BPN-ACO)

Initially the selected features from the feature selection algorithms are normalized between [0,1]. That is each value in the feature set is divided by the maximum value from the set. These normalized values are assigned to the input neurons. The number of hidden neurons is equal to the number of input neurons[106, 107]. And only one output neuron. Initial weights are extracted using the ACO algorithm as follows:

In weight extraction, N random numbers are generated with d number of digits. Where, N is the total number of neurons in the BPN. The weights are extracted from the population of random numbers to determine the fitness values. The actual weight \( w_k \) is given by:

\[
    w_k = \frac{c^n [ x_{kd+2} 10^{d-2} + x_{kd+3} 10^{d-3} + \ldots + x_{(k+1)d} ]}{10^{d-2}}
\]

where \( c=1 \), if \( 5 \leq x_{kd+1} \leq 9 \), else \( c=-1 \), and \( k \) represents the population. The weights are extracted for each string in the population. The fitness values is calculated as defined below: \( F = 1 / E \),

Where, \( E = \sqrt{[ E_1 + E_2 + \ldots + E_m ] / m} \), Where, \( m \) – is the total number of training patterns, and \( E_1, E_2 \ldots, E_m \) are the errors for each pattern, i.e., \( E_i = (T_i - O_i)^2 \), where \( T_i \) is the desired output, and \( O_i \) is the actual result of the output layer.

Thus the fitness value is calculated for a single population. Like this \( M \) populations are generated and their fitness values are calculated. The optimum fitness value is selected using ACO algorithm. All the minimum fitness values are replaced with the maximum fitness values. Now the weights are updated with the new fitness value and the training is performed again. This procedure is repeated until the error from the backpropagation network is less than the tolerance value. The algorithm is as follows:
Algorithm:

1. Repeat for M times
   a. Generate N random numbers with d number of digits
   b. Extract the weights from the random numbers as:
      c. \( w_k = \left\{ c \times \left[ x_{kd+2} \times 10^{d-2} + x_{kd+3} \times 10^{d-3} + \ldots + x_{(k+1)d} \right] \right\} / 10^{d-2} \)
      d. where \( c = 1 \), if \( 5 \leq x_{kd+1} \leq 9 \), else \( c = -1 \)
   c. Calculate the output at the hidden layer, \( S_1 = 1/(1+e^{-\lambda x}) \), where \( \lambda = 1 \), and \( x = \sum_i w_{ih} k_i \)
   d. Calculate the output at the output layer, \( S_2 = 1 / (1 + e^{-\lambda x}) \), where \( \lambda = 1 \), and \( x = \sum_i w_{ho} S_i \)
   e. Find the error, \( E_i = (T_i - O_i)^2 \)
   f. Repeat the steps (e) – (g) for all the training patterns.
   g. Calculate the Mean Square error, \( E = \sqrt{\frac{E_1 + E_2 + \ldots + E_m}{m}} \)
   h. Calculate the fitness values for each weight \( F_k = 1 / E \)

2. \( P_i \leftarrow F_k \), initial population for ACO algorithm
3. The optimum fitness value \( F_{\text{max}} \) is found out using ACO algorithm
4. The minimum fitness values are replaced with \( F_{\text{max}} \).
5. Repeat the procedure of weight extraction and fitness value calculation for the updated weights.
6. Repeat the steps (1)-(6) till the error becomes less than the tolerance value.

5.3.3. Experiments and Results

The output from the each hidden neuron is calculated using the sigmoid function, 
\( S_1 = 1/(1+e^{-\lambda x}) \), where \( \lambda = 1 \), and \( x = \sum_i w_{ih} k_i \), where \( w_{ih} \) is the weight assigned between input and hidden layer, and \( k \) is the input value. The output from the output layer is
calculated using the sigmoid function, \( S_2 = \frac{1}{1 + e^{-\lambda x}} \), where \( \lambda = 1 \), and \( x = \sum_i w_{ho} S_i \), where \( w_{ho} \) is the weight assigned between hidden and output layer, and \( S_i \) is the output value from hidden neurons. \( S_2 \) is subtracted from the desired output. The network is trained to produce a 0.9 output value for positive ROI (malignant) and 0.1 output value for a negative ROI (benign). The classification performance was studied by the jack-knife method. And the results were analyzed by using ROC analysis.

5.4 Summary

The mammogram textural features are extracted from the segmented image. The textural analysis method Surrounding Region Dependency Matrix is used to extract the fourteen Haralick features from the segmented image.

Textural features are extracted for classification of microcalcifications. The feature set may contain irrelevant or redundant information. These features are eliminated to improve the accuracy and to reduce the time complexity of the classifier. In this chapter, rough set based reduction algorithms and metaheuristic algorithms are used to select the features from the feature set. The reduced feature sets from each selection algorithms are combined to form the reduced feature set, which is used for classification.

A three-layer Back Propagation Neural network is used for classification. The values of the features available in the reduced feature set, constructed from the feature selection algorithms are normalized and given as input to the classifier. For each testing image, the output is calculated using sigmoid function. The error is calculated between the actual output and the target output. Based on this error value the weights are propagated to reduce the error value.

Thus the classifier is trained to produce the output value 0.9 for malignant images, 0.5 output values for benign images and 0.1 for normal images. The performances of all possible combinations between segmentation and feature extraction methods have been evaluated using this ROC analysis discussed in chapter 6.