IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

Enhanced Feature Extraction, Selection and Classification of MRI and Mammogram Texture Analysis

A. Sivaramakrishnan[1*], M. Asmi[2]

Assistant Professor, Department of Computer Science, DMI St. Eugene University, Lusaka, Zambia[1]

Lecturer, Department of Computer Science, DMI St. Eugene University, Lusaka, Zambia[2]

Abstract: The MRI and mammogram 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. Feature selection is focused on many areas, especially in artificial intelligence, medical image processing, Data Mining [Dom et al.] and pattern recognition. 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 labelled or unlabelled data.

Keywords: Magnetic Resonance Imaging (MRI), Surrounding Region Dependency Matrix, Spatial Gray Level Dependency Matrix, Feature Selection.

I. INTRODUCTION

The MRI and mammogram 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. Feature selection is focused on many areas, especially in artificial intelligence, medical image processing, Data Mining [Dom et al.] and pattern recognition. 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 labelled or unlabelled data.

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 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.

II. TEXTURE-ANALYSIS METHODS

This texture-analysis method such as Surrounding Region Dependency Matrix (SRDM), Spatial Gray Level Dependency Matrix (SGLDM), Gray Level Run-Length Matrix (GLRLM) and the Gray Level Difference Matrix (GLDM). The segmented mammogram and brain images are considered inputs for feature extraction methods.

III. SURROUNDING REGION DEPENDENCY MATRIX

The SRDM is based on a second-order histogram in two surrounding regions. The mammogram and brain image is transformed into a Surrounding Region-Dependency Matrix and the features are extracted for this matrix. Let us consider two rectangular windows centered on a current pixel (x, y). R1 and R2 are the outermost and outer surrounding region of size 7 7 and 5 5, respectively. The number of pixels greater than the selected threshold value (q) is counted in each region. Let us assume m and n to be the total number of pixels from the outermost region (R1) and the outer region (R2). The element in the corresponding surrounding region dependency matrix M (m, n) is incremented by 1. This procedure is repeated for all the image pixels and the matrix gets updated. Figure 1.1 shows the algorithm of SRDM.

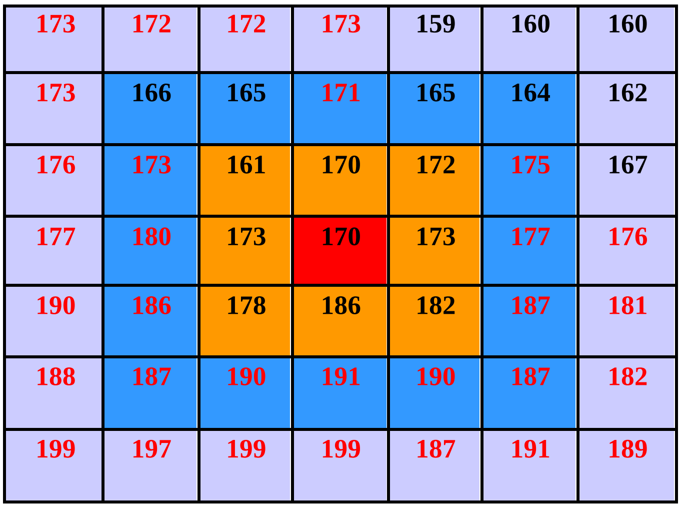
Step 1. Mij ⟵ Original Image
Step 2. P ← Select a pixel, whose intensity value is greater than the selected threshold value.
Step 3. R1, ← outer most regions of size 7 7 Step 4. R2 outer regions of size 5 5
Step 5. M(q) ← [(i, j)], $0 \le i \le m$, $0 \le j \le n$: where q is the threshold value and m, n are the number of pixels in the regions R1 and R2.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

Vol. 9, Issue 6, June 2020

Step 6. a. (I, j) ⟵ #{(x, y)|cR1 (x, y) = i and cR2 (x, y) = j, (x, y) Lx Ly} where cR1 ⟵ (x, y) # {(k, l) | (k, l) R1 and [S(x, y) – S(k, l)]> q } b. cR2 ⟵ (x, y) #{ (k, .l) | (k, l) R2 and [S(x, y) – S(k, l)] > q } where # denotes the number of elements in the set
c. S (x, y) is the intensity value of the current pixel (x, y)

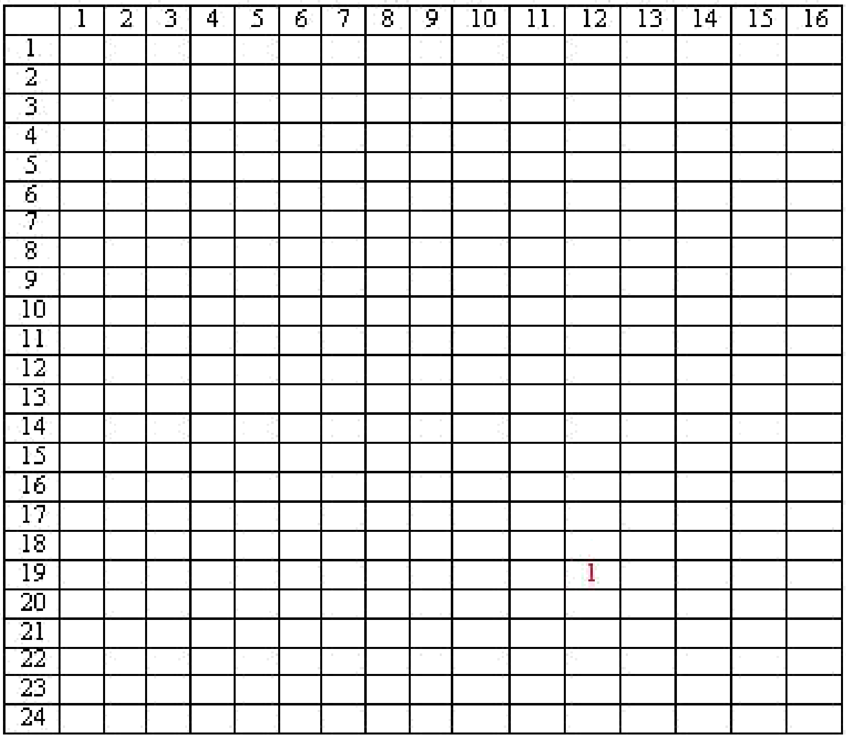
Figure 1.1 Algorithm Surrounding Region Dependency Matrix: Algorithm



173	172	172	173	159	160	160
173	166	165	171	165	164	162
176	173	161	170	172	175	167
177	180	173	170	173	177	176
190	186	178	186	182	187	181
188	187	190	191	190	187	182
199	197	199	199	187	191	189

Figure 1.2 7*7 image for Surrounding Region Dependency Matrix extraction

Figure 1.2 shows a sub image of a segmented image for constructing SRDM matrix. Figure 1.3 shows a SRDM matrix. Normally a set of five different thresholds is selected, such as 120, 130, 140 150 and160. A separate matrix is generated for each threshold value. The SRDM matrix has the dimension of 24 × 16, where 24 is the total number of pixels in the R1 region and 16 is the total number of pixels in the R2 region.
For example, if the threshold value is 170, R1 contains 19 pixels and region R2 contains 12 pixels having greater intensity values than the threshold value. So, the value of (19, 12)th element in the SRDM matrix is incremented by one as M(19,12)=M(19,12)+1.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19												1				
20																
21																
22																
23																
24																

Figure 1.3 16*24 SRDM matrix

DOI 10.17148/IJARCCE.2020.9636

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 6, June 2020

IV. SPATIAL GRAY LEVEL DEPENDENCY MATRIX

In this method, a co-occurrence matrix is generated to extract the texture features from the segmented mammogram and brain image. The co-occurrence matrix is a technique that allows for the extraction of statistical information from the image regarding the distribution of pairs of pixels. It is computed by defining a direction and a distance (d) and pairs of pixels separated by this distance, computed across the defined direction ( ), are analyzed. A count is then made of the number of pairs of pixels that possess a given distribution of gray level values. Each entry of the matrix thus corresponds to one such gray level distribution.

The mammogram image is an eight-bit image; for such an image, the allowed gray level values range from 0 to 255. The size of this matrix will then be 256 256. A set of 20 co-occurrence matrices are computed for five different distances in the horizontal, vertical and two diagonal directions: The distances are 1, 3, 5, 7 and 9 and the four angles 0o, 45o, 90o and 135o are defined for calculating the matrix for each of the five distances. Since the co- occurrence matrix analyzes the gray level distribution of pairs of pixels, it is also known as the second-order histogram.

The estimated joint conditional probability density functions are defined as follows:

P (i,j | d, 0) = # { (k, l), (m, n)) ε (Lx Ly) (Lx Ly); k=m, |l - n| = d, S(k, l) = i, S(m, n) = j } / T(d, 0)

P (i,j | d, 45) = # { (k, l), (m, n)) ε (Lx Ly) (Lx Ly);

k-m = d, l - n = -d, or k-m = -d, l-n = d
S(k, l) = i, S(m, n) = j } / T(d, 45)

P (i,j | d, 90) = # { (k, l), (m, n)) ε (Lx Ly) (Lx Ly);

k-m = d, l = n, S(k, l) = i, S(m, n) = j } / T(d, 90)

P (I, j | d, 135)= # { (k, l), (m, n)) ε (Lx Ly) (Lx Ly); k-m=d, l–n=d, or k-m =-d, l-n =-d S(k, l)=i, S(m, n)=j} / T(d, 135)

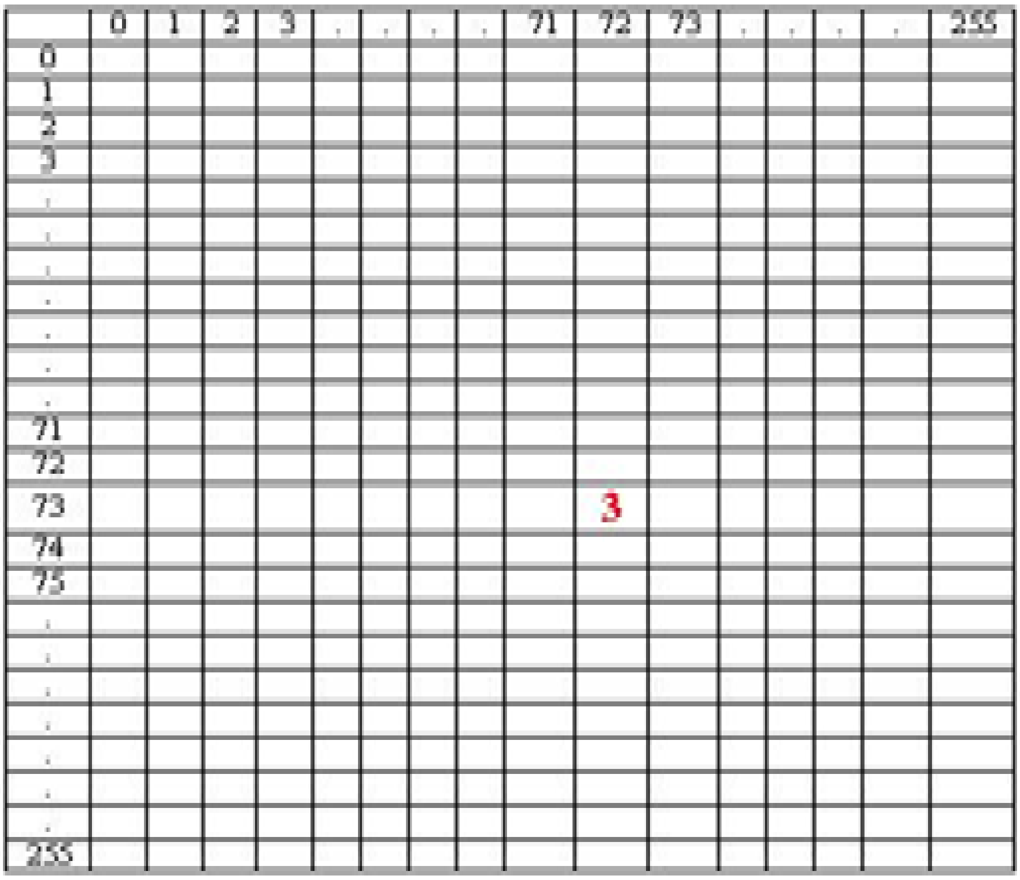
where # denotes the number of elements in the set, S(x, y) is the image intensity at the point (x, y), k, l, m and n are the spatial coordinates, Lx and Ly are the dimension for SGLD matrices and T stands for the total number of pixel pairs within the image which have the inter sample distance d and direction. The features are selected for various combinations of distance and theta values let us consider a 7×7sub image of a segmented image. Suppose d = 1 and the angle = 0^0 and 180^0, then count the number of similar pairs of pixels which are successive in horizontal direction and enter the count into the matrix as M (73, 72) = 3. Figure 5.4 shows a sub image for constructing the SGLDM. Figure 5.5 shows a typical SGLDM matrix.

73	72	65	63	59	60	60
69	66	65	71	65	64	62
64	73	72	75	76	75	67
77	80	73	70	72	73	76
90	86	78	86	82	87	81
88	87	90	91	90	87	82
99	97	99	99	87	91	89

Figure 1.4 7*7 image for Spatial Gray Level Dependency Matrix extraction

DOI 10.17148/IJARCCE.2020.9636

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940



	0	1	2	3	.	.	.	.	71	72	73	.	.	.	.	255
0																
1																
2																
3																
.																
.																
.																
.																
.																
.																
.																
71																
72																
73										3						
74																
75																
.																
.																
.																
.																
.																
.																
.																
255																

Figure 1.5 SGLDM matrix

V. GRAY LEVEL RUN-LENGTH MATRIX

The run-length matrix is a way of searching the image, always across a given direction, for runs of pixels having the same gray level value. Thus, given a direction, the run-length matrix measures for each allowed gray level value how many times there are runs of, for example, 2 consecutive pixels with the same value. Next it does the same for 3 consecutive pixels, then for 4, 5 and so on. Note that many different run-length matrices may be computed for a single image, one for each chosen direction (Galloway, M.M).

The GLRLM is based on computing the number of gray level runs of various lengths. A gray level run is a set of consecutive and collinear pixel points having the same gray level value. The length of the run is the number of pixel points in the run. The gray level run-length matrix is as follows.

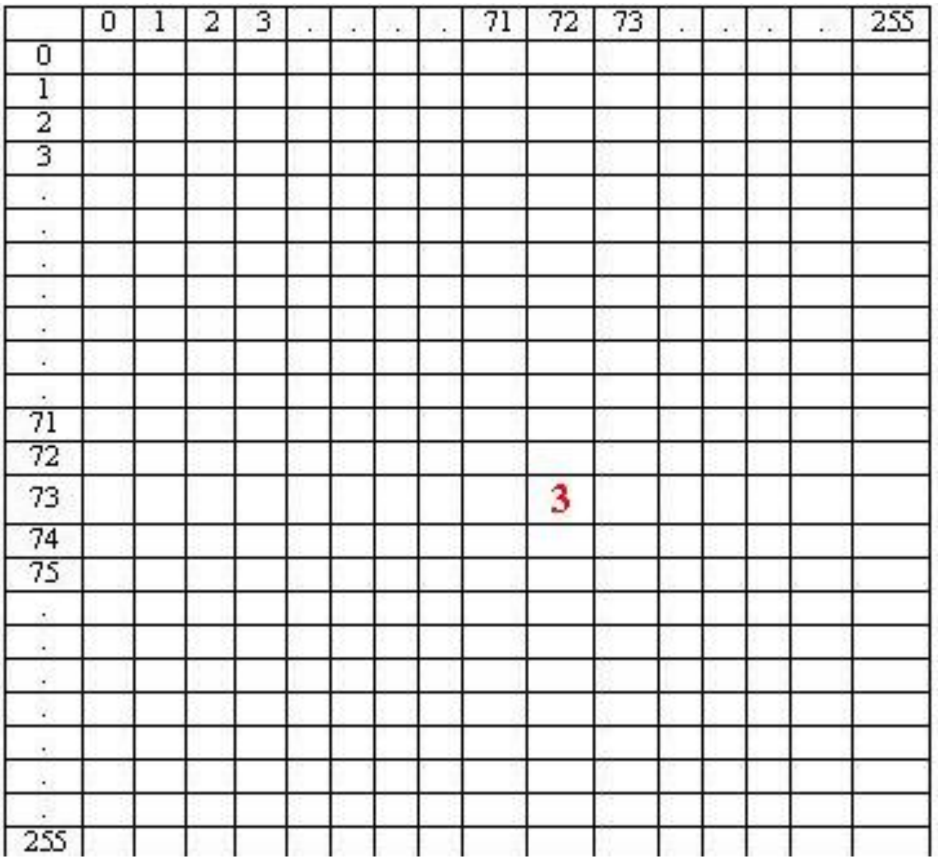
$R(\theta) = (g(i, j) \mid \theta), 0 \le i \le Ng , 0 \le j \le Rmax$

where Ng is the maximum gray level and Rmax is the maximum length.

For example, let us consider, angle = 0^0, M (80,1) = 2. The count is made for all the distances in the same direction and they are summed up.

Figure 1.6 shows the sub image of a segmented image for constructing the GLRLM. Figure 1.7 shows that the GLRLM of the sub image in Figure 1.6.

DOI 10.17148/IJARCCE.2020.9636



	0	1	2	3	.	.	.	.	71	72	73	.	.	.	.	255
0																
1																
2																
3																
.																
.																
.																
.																
.																
.																
.																
71																
72																
73										3						
74																
75																
.																
.																
.																
.																
.																
.																
.																
255																

Figure 1.6 GLRLM matrix

73	72	65	63	59	60	60
69	66	65	71	65	64	62
64	73	72	75	76	75	67
77	80	73	80	73	72	76
90	86	78	86	82	87	81
80	87	80	91	90	87	82
99	97	99	99	87	91	89

Figure 1.7 7*7 image for Gray Level Run-Length Matrix extraction

VI. GRAY LEVEL DIFFERENCE MATRIX

The GLDM is based on the occurrence of two pixels which have a given absolute difference in gray level and which are separated by a specific displacement. For any given displacement vector

δ= (Δx, Δy) let S(x, y) = | S(x, y) - S(x+ Δx, y+ Δy) and

the estimated probability-density function is defined by

(i | δType equation here.) = Prob (So (x, y) = 1)$\leq$

Let us consider difference = 3, D = 1 and angle = 0^0; the element in the matrix is assigned the value of 5 as M (3) = 5. Figure 1.8 shows the sub image of a segmented image for constructing the GLDM. Figure 1.9 shows that the GLDM of the sub image in Figure 1.8

DOI 10.17148/IJARCCE.2020.9636

73	72	65	63	59	60	60
69	66	65	71	65	64	62
64	73	72	75	76	75	67
77	80	73	80	73	72	76
90	86	78	86	82	87	81
80	87	90	91	90	87	82
99	97	99	99	87	91	89

Figure 1.8 7*7 image for Gray Level Difference Matrix extraction

The Haralick features are extracted from all the texture analysis methods, from these features set; feature selection algorithms are used to select the reduced features. The Haralick features are discussed as follows:

VII. THE HARALICK FEATURES

The texture of mammogram and MRI brain 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 (Nadler, M. and Smith, E.P 1993) .

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

a. Feature that express visual texture characteristic: Angular Second Moment (ASM), Contrast(CON), Correlation (COR)
b. Features that are based on statistics: Variance (VAR), Inverse Difference Moment (IDM), Sum Average (SA), Sum Variance (SV) and Difference Variance (DV).
c. Features that are based on information theory: Entropy (ENT), Sum Entropy (SENT) & Difference Entropy (DENT)
d. Features that are based on information measures of correlation: Information Measures of Correlation (IMC1, IMC2) and Maximal Correlation Coefficient (MCC).

VIII. FEATURE SELECTION

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 (Kittler, J). 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.

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 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.

DOI 10.17148/IJARCCE.2020.9636

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. An overview of classical feature selection methods can be found in.

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.
- Reduce 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, seven different algorithms such as Decision Relative Discernibility based reduction, Heuristic approach, Hu’s algorithm, Quick Reduct (QR) and Variable Precision Rough Set, Genetic Algorithm and Ant Colony Optimization algorithm are proposed for feature selection. A comparative analysis has also been made with emphasis on images to select the best feature set.

Rough Set Based Feature Selection

In 1982, Pawlak introduced the theory of Rough Sets. This theory was initially developed for a finite universe of discourse in which the knowledge base is a partition, which is obtained by any equivalence relation defined on the universe of discourse. In the rough sets theory, the data is collected in a table called the decision table. Rows of the decision table correspond to objects and columns correspond to features (Lin, T.Y. and Cercone: Pawlak 1982, 1991)[145].
In the data set, a class label indicates the class to which each row belongs. The class label is called a decision feature and the rest of the features are the condition features. Consider that the data set (condition-features, decision-features) is stored in a relational table with the form Table. C is used to denote the condition features, D for decision features, where C D = Φ and tj denotes the j-th tuple of the data table. Rough sets theory defines three regions based on the equivalent classes induced by the feature values: lower approximation, upper approximation and boundary.

Lower approximation contains all the objects, which are classified surely based on the data collected and upper approximation contains all the objects, which can be classified probably, while the boundary is the difference between the upper approximation and the lower approximation. Hu et al. presented the formal definitions for rough sets theory
Let U be any finite universe of discourse. Let R be any equivalence relation defined on U. Here, (U, R) which is the collection of all equivalence classes is called the approximation space. Let W1, W2, W3 ,…, Wn be the elements of the approximation space (U, R). This collection is called knowledge base. Then for any subset A of U, the lower and upper approximations are defined as follows:

$$\underline{R}A = \cup\{Wi / Wi \leq A\}$$
$$RA = \cup\{Wi / Wi \cap A \neq \emptyset\}$$

The ordered ($\underline{R}$A, RA) pair is called a rough set. Once defined these approximations of A, the reference universe U is divided in three different regions: the positive region POSR(A), the negative region NEGR(A) and the boundary region BNDR(A), defined as follows:

$$POSR\ (A) = \underline{R}A$$
$$NEGR\ (A) = U - RA$$
$$BNDR\ (A) = RA - \underline{R}A$$

Hence, it is trivial that if BND (A) = θ, then A is exact. This approach provides a mathematical tool that can be used to find out all possible reduces. Two kinds of features are generally perceived as being unnecessary: features that are irrelevant to the target concept (like the row ID, customer ID) and features that are redundant, given other features. In actual applications, these two kinds of unnecessary features can exist at the same time but the latter redundant features are more difficult to eliminate because of the interactions between them. In order to reduce both kinds of unnecessary features to a minimum, feature selection is used.

Feature selection is a process to choose a subset of features from the original features. Feature selection has been studied intensively in the past decades. The purpose of the feature selection is to identify the significant features, eliminate the features that are irrelevant or dispensable and build a good learning model. The benefits of feature selection are twofold:

it considerably decreases the computation time of the induction algorithm and increases the accuracy of the resulting mode.

DRD based Feature Reduction

Peter and Skowron introduced the representation of the decision table into the discernibility matrix to compute the reduct. Let T = (U, A, C, D) be a decision table. By a discernibility matrix of T, denoted by M (T), mean the n n matrix is defined as:

$$M_{ij}=\{a\ \ C: a(x_i) \neq a(x_i) \wedge d\ \ D: d(x_i) \neq (x_i)\},\ i, j=1, 2, \ldots, n$$

The discernibility function is given by taking the conjunction of the disjunctive expressions of the discernibility matrix.

IX. HU'S ALGORITHM

Hu et al. claimed that the reduct algorithms developed based on the database operations Projection and Count is more efficient than the algorithms developed based on the traditional rough set models The data table may contain inconsistent records. Two records are said to be inconsistent if they have the same values on the condition features, but are labeled as different classes (having different values on the decision features). Inconsistent records cannot be classified.
Thus, the inconsistent records should be eliminated from the data table before the classification process proceeds. It is assumed that the inconsistent records are noisy data; otherwise more features and values for records should be collected further to ensure that the data table is consistent. The core features are selected initially to eliminate the inconsistent features from the data table. These core features are further combined with the condition features to form a new set of features and the table of features is used to reduce the features. Figure 1.10 shows the algorithm for finding core features. Figure 1.11 shows the algorithm of computing a minimal feature subset.
Output: A Set of minimum feature subset (REDU)

Method:

Step 1. Run the algorithm in Figure 6.1 to get the core features of the table CO
Step 2. REDU = CO
Step 3. AR = C – REDU
Step 4. Compute the merit values for all features of AR Merit (Cj, C, D) =1-Card (π(C-Cj+D))/Card (π(C+D))
Step 5. Sort features in AR based on merit values in decreasing order
Step 6. Choose a feature Cj with the largest merit values (if there are several features with the same merit value, choose the feature which has the least number of combinations with those features in REDU)
Step 7. REDU = REDU $\cup$\{ Cj \}, AR = AR – \{ Cj \}
Step 8. If K (REDU, D) = 1, then terminate, otherwise go back to step (d).
Step 9. K(REDU, D)=Card (π(REDU + D)) / Care(C + D)

Figure 1.10 Algorithm of compute a minimal feature subset

This algorithm initially calls the core reduction algorithm to find all the core features and initializes the reduct with the complement of the core features set against the condition features set. Then the algorithm ranks the features based on the features merit and adopts the backward elimination approach to remove the redundant features. When two or more features have the same merit values, the feature with the least number of possible values is removed. This process is repeated until a reduct is generated.

Heuristic Algorithm for Feature Selection

Features are selected one by one from among the unselected features and added them to the features subset until a reduct approximation is obtained. Figure 1.12 shows the Algorithm of Heuristic approach.

Quick Reduct Algorithm

The reduction of features is achieved by comparing equivalence relations generated by sets of features. Features are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct is defined as a subset of minimal cardinality Rmin of the conditional feature set C such that gR(D) = gC(D).

$R = \{X : X \ \ \leq C;\ g_X(D) = g_C(D)\}$
$R_{min} = \{X : X \ \ \in R;\ \ Y \ \ \in R;\ |X| \ \ \leq |Y|\ \}$

```
Step 1.  Let R be a set of selected condition features, P be a set of unselected condition features, U a set of all instances
         and EXPECT an accuracy threshold. In the initial state, set R = CORE(C). P = C - CORE(C), K = 0.
Step 2.  Remove all consistent instances: U = U - POSR (D)
Step 3.  If K >= EXPECT,
where K = γR(D) = Card(POSR(D)) / Card(U), then stop
Step 4.  Else if POSR (D) = POSC (D) return ‘only K = Card (POSC (D) / Card (U) is available’ and stop.
Step 5.  Calculate γp = Card (POS R  ∪{p}(D))
Step 6.  mp = max-size(POS (R ∪ {p})(D))/(R  ∪{p}∪  D) for any p    ∈P
Step 7.  Choose the best feature p. i.e. that with the largest vp X mp and set R = R ∪{p}, P = P – {p}
Step 8.  Go back to step (c).
```

Figure 1.11 Algorithm of Heuristic approach

The intersection of all the sets in Rmin is called the core, the elements of which are those features that cannot be eliminated without introducing more contradictions to the dataset. In this method a subset with minimum cardinality is searched for. Figure 1.13 shows the Algorithm of Quick Reduct Method.
The Quick Reduct algorithm attempts to calculate a reduct without exhaustively generating all possible subsets. It starts off with an empty set and adds in turn, one at a time, those features that result in the greatest increase in the rough set dependency matrix, until this produces its maximum possible value for the dataset.

```
QUICK REDUCT (C, D)

C, the set of all conditional features;

D, the set of decision features.

Step 1.  R ←—— {}
Step 2.  Do
Step 3.  T ←—— R
Step 4.    ∀x € (C-R)
Step 5.  if gR ∪ {x}(D) > gT (D),
where gR(D)=Card(POSR(D)) / Card(U)
Step 6.  T  ←——    R ∪ {x}
Step 7.  R  ←——  T
Step 8.  until gR(D) = gC(D)
Step 9.  return R
```

Figure 1.12 Algorithm of QR method

According to the Quick Reduct algorithm, the dependency of each feature is calculated and the best candidate chosen. However, is not guaranteed to find a minimal subset as has been shown in. Using the dependency function to discriminate between candidates may lead the search down a non-minimal path. It is impossible to predict which combinations of features will lead to an optimal reduct based on changes in dependency with the addition or deletion of single features. It does result in a close to minimal subset though, which is still useful in greatly reducing dataset dimensionality. Chouchoulas et al. presents a potential solution to this problem wherein the Quick Reduct algorithm is altered, making it into an n-look-ahead approach . Even this cannot guarantee a reduct unless n is equal to the original number of features, but this reverts back to generate and test. It still suffers from the same problem as the original Quick Reduct, i.e. it is impossible to tell at any stage whether the current path will be the shortest to a reduct.

Genetic Algorithm

In this analysis, textural matrices such as SRDM, SGLDM, GLRLM and GLDM are created for each mammogram and brain image. For each defined distance and direction the Haralick features are extracted for all the300 MRI and 322-mammogram images. A single feature value for all the images is considered the initial population string for Genetic Algorithm (Goldberg, D.E).

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.

DOI 10.17148/IJARCCE.2020.9636

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 (Pm), where Pm 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 1.14 shows the algorithm of feature selection using GA.

```
Step 1. Pi          feature values.
Step 2. Fi = 1/ (1+Pi ), { Fitness values}
Step 3. Calculate the probability and cumulative probability, CP
Step 4. Reproduction
a.        r       random()   ←
b.        if (CPi > r)  count=count + 1 for CPi
c.        Repeat the steps (a) and (b) for all the population strings.
d.        If count=0, then delete that Pi
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 > Pc)  k          random bit position  ←
d.        interchange the bits after kth position in parent1 and parent2
e.        repeat this step for all the pairs.

Step 6. Mutation
a.        r      random()←
b.        if (r >= Pm), k   random bit position
c.        complement the value of the kth bit
d.        repeat this steps for all the strings.

Step 7.   Pnew  ←  population after reproduction, crossover and mutation.
Step 8.   Pi ←  Pnew
Step 9.   Goto Step 2.
```

Figure 1.13 Algorithm of feature selection using GA

X. ANT COLONY OPTIMIZATION

In the proposed algorithm, the individual feature set is considered as the solution space for the ACO search. Each feature value is labeled with a number corresponding to its fitness value, calculated using the fitness function (1/1+Pi), where Pi is the feature value. A solution matrix is created with the feature values, fitness values and their corresponding labels (Dorigo, M. and Di Caro, G).

Initially, the number of ants (NA) start their search from a randomly selected feature value, with an initial pheromone of To. A random number q is generated and is compared with qo, if q ≤qo. Then the corresponding feature value is assigned the maximum label from the label set. Otherwise, a random label is assigned for that feature value. This step is repeated for each ant and for each feature value in the feature set. Once all the ants are created their solution, the pheromones of the ants are locally updated.

Then the fitness values of all the ants are locally improved by replacing with the maximum fitness value. The maximum fitness value is searched from the set of fitness values of the features having optimum label. The optimum fitness value is selected from the set of fitness values from the set of solutions created by all the ants. This value is known as the local minimum *(Lmin*). If this value is greater than the global minimum *(Gmin*), then *Gmin* is assigned to the *Lmin.* The ant that generates the Gmin is globally updated. At the final iteration, the *Gmin* has the label of the optimum feature.
The Figure 1.15 shows the algorithm of ACO for Feature Selection.

```
Step 1.   Pi $\longleftarrow$ feature values from the feature set for a particular distance and direction.
Step 2.   Fi = 1/(1+Pi), Fitness values
Step 3.   Li $\longleftarrow$ Label Matrix for fitness values
Step 4. Initialize all the ants with To, the initial pheromone value Step
5. Repeat for NI times
a. For each ant a = 1..A
b.        For each feature value
              i.      q $\leftarrow$ random()
              ii.     if (q  qo); Li    max(L)
              iii.    else Li $\leftarrow$ random() End End
c.        For each ant
d.        For each feature value
              i.      If (Li = max(L))
              ii.     Tnew = (1 – $\rho$) * Told + $\rho$* T0
e.        For each ant locally improve the fitness value
f.        Lmin $\leftarrow$ min(F2i);
g.        If (Lmin < Gmin ) then Gmin = Lmin
h.        Update the pheromone of the ants globally;

Tnew = (1 – $\alpha$) * Told + $\alpha$* $\Delta$Told, where 0 <$\alpha$ < 1 and $\Delta$ $\longleftarrow$ ( 1 / Gmin) for the ant which generates the Gmin, for
the remaining ants $\Delta$        0;
Step 6.  End
```

Figure 1.14 Algorithm of ACO for feature selection

This entire procedure can be repeated a number of times for obtaining the further enhanced value. As in the ACO 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.

CLASSIFICATION

Neural Networks (NN) can learn various tasks from training examples: classify phenomena and model 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.

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. Table 5.1 shows the components of an ANN. Perceptron is a special case of feed forward neural networks with only input and output nodes.

DOI 10.17148/IJARCCE.2020.9636

Table 1.1 Artificial Neural Networks components

SI.NO	Key Terms	Description
1	**Back Propagation**	A learning algorithm for multi-layered feed forward networks that uses the sigmoid function.
2	**Hidden layer**	The set of nodes that are not input or output units.
3	**Learning rate**	A value greater than zero but less than one, this is used so that the weights on the links do not change to quickly or the ANN might never converge onto the optimal solution.
4	**Linearly separable function**	A function where if plotted in a n-dimensional plane, the negative and positive examples of the function can be totally separated using a straight plane across the space.
5	**Multi-layer Feed Forward Networks**	A network with at least one unit that is not output or input, where the direction of data flow is in only one direction.
6	**Perceptron**	A network with no units that are output or input, where the direction of data flows is in only one direction.
7	**Supervised learning**	All learning algorithms where the known targets are used to adjust the network.
8	**Target**	The expected output of the input. This is used to calculate the error.
9	**Threshold function**	The function to decide whether a unit should fire or not. Typically, one for exceeding the threshold and zero otherwise.
10	**Units/Nodes**	Simple elements of an ANN, they take in input from other nodes or training data, sum up the data and applies a threshold function to decide what output to send.
11	**Weighted links**	Connects units together, conceptually shows the strength of the bond between two units.

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.

Artificial Neural Networks

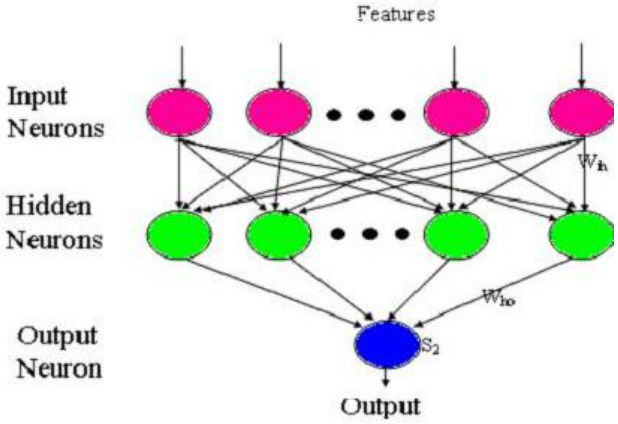


Figure 1.15 Back Propagation Neural network

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. Figure 5.15 shows the architecture of the BPN network and these networks allow for learning using highly parallel series of simple units and are suited for data that is noisy and vector based.

XI. PAGE STYLE

Back Propagation Algorithm

The Back Propagation algorithm can be implemented in two different modes: online mode and batch mode. In the online mode the error function is calculated after the presentation of each input image features and the error signal is propagated back through the network, modifying the weights before the presentation of the next image features. This error function is usually the Mean Square Error (MSE) of the difference between the desired and the actual responses of the network over all the output units (Renco).

Then the new weights remain fixed and a new image features are presented to the network and this process continues until all the image features have been presented to the network. The presentation of all the image features is usually called one epoch or a single iteration. In practice many epochs are needed before the error becomes acceptably small.
In the batch mode the error signal is calculated for each input image features and the weights are modified every time the input image features is been presented. Then the error function is calculated as the sum of the individual MSE for each image features and the weights are accordingly modified (all in a single step for all the images) before the next iteration. The single layer perceptron provides a powerful solution to the problems, which are linearly separable. Multi-layer perceptrons are considered to be difficult for training. Error Back Propagation algorithm is an effective solution to train multi-layer perceptrons based on error correction learning.

In the forward pass outputs are computed and in the backward pass weights are updated or corrected based on the errors. The development of the Back Propagation algorithm is a landmark in neural networks in that it provides a computationally efficient method for the training of multi-layer perceptrons.

Classifier

Initially the reduced feature set selected from the feature selection algorithms are normalized between zero and one. 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(herz etal).

The number of hidden neurons is greater than or equal to the number of input neurons. And there is only one output neuron. Initial weights are assigned randomly (-0.5 to 0.5). The output from the each hidden neuron is calculated using the sigmoid function

S1 = 1 / (1 + $e^{- x}$), where   = 1 and x =   i wih ki

where wih 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

S2 = 1 / (1 + $e^{- x}$), where =1, and x =   i who Si

where who is the weight assigned between hidden and output layer and Si is the output value from hidden neurons. S2 is subtracted from the desired output. Using this error (e) value, the updation of weight is performed as:

delta = e * S2 * (1 – S2)

Figure 5.16 shows the Algorithm of Back Propagation classifier for classification of massess. The diagrammatic representation of the classifier is given in Figure.

Values from the optimum feature set are considered as input.

These feature values are normalized between 0 and 1 and assigned to input neurons.

Initial weights are assigned randomly between -0.5 to 0.5

DOI 10.17148/IJARCCE.2020.9636

The output from the each hidden neuron is calculated using the sigmoid function, S1 = 1 / (1 + $e^{- x}$), where Type equation here.=1 and x = i wih ki where wih is the weight assigned between input and hidden layer and ki is the input value.

The output from the output layer is calculated using the sigmoid function, S2 = 1 / (1 + $e^{- x}$), where =1, and x = who Si where who is the weight assigned i between hidden and output layer and Si is the output value from hidden neurons.

S2 is subtracts from the desired output. Using this error (e) value, the weight change is calculated as: delta=e*S2*(1–S2)

Update the weights using this delta value.

Who = Who + (n * delta * S1);

Wih = Wih + (n * delta * ki) where n is the learning rate, k is the input values.

Perform steps (5) to (10) with the updated weights, till the target output is equal to the desired output.

Figure 1.16 Algorithm of BPN classifier for image classification

The weights assigned between input and hidden layer and hidden and output layer are updated as:

Who = Who + (n * delta * S1)

Wih = Wih + (n * delta * ki)

where n is the learning rate, k is the input values. Again the output is calculated from hidden and output neurons. Then the error (e) value is checked and the weights are updated. This procedure is repeated till the target output is equal to the desired output. The network is trained to produce the output value 0.9 for abnormal images and 0.1 for normal images.

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.

The following table 1.2 shows the extracted feature values based on SRDM, SGLDM, GLRLM and GLDM from the MRI and Mammogram segmented image. The experiments were carried out for all the 300 MRI and 322 mammogram images and the results for the first twenty images are listed in the following tables.

Table 1.2 SRDM based feature extraction for mammogram

ASM	**CON**	**COR**	**VAR**	**IDM**	**SA**	**SV**
0.073673	0.053362	-0.03105	-0.024	-0.16687	0.056492	-0.05769
0.080277	0.056555	-0.0336	-0.02727	-0.18319	0.060078	-0.0618
0.074915	0.055423	-0.03273	-0.02602	-0.17739	0.058804	-0.06033
0.084864	0.061163	-0.03484	-0.03354	-0.19342	0.065266	-0.06794
0.06789	0.054821	-0.02946	-0.02521	-0.15114	0.058115	-0.05954
0.089137	0.063203	-0.03611	-0.03732	-0.19475	0.067575	-0.07075

Table 1.2 SRDM based feature extraction for mammogram (cont)

DV	ENT	SENT	DENT	IMC1	IMC2	MCC
0.048116	-0.1019	0.086991	-0.05771	65535	-0.10968	0.334905
0.050691	-0.11024	0.09733	-0.06184	65535	-0.1183	0.351254
0.049782	-0.10651	0.092291	-0.06036	65535	-0.11529	0.30076
0.05433	-0.1147	0.10363	-0.06805	65535	-0.12248	0.339962
0.049286	-0.09677	0.080996	-0.05959	65535	-0.10433	0.349275
0.055919	-0.1193	0.109478	-0.07091	-0.37161	-0.12677	0.346227

Table 1.3 SRDM based feature extraction for MRI brain image

ASM	CON	COR	VAR	IDM	SA	SV
0.074915	0.055423	-0.03273	-0.02602	-0.17739	0.058804	-0.06033
0.084988	0.061175	-0.03493	-0.03356	-0.19492	0.065281	-0.06796
0.06789	0.054821	-0.02946	-0.02521	-0.15114	0.058115	-0.05954
0.089137	0.063203	-0.03611	-0.03732	-0.19475	0.067575	-0.07075
0.066942	0.05196	-0.02943	-0.02263	-0.15633	0.054925	-0.05592

Table 1.2 SRDM based feature extraction for MRI brain image (cont)

DV	ENT	SENT	DENT	IMC1	IMC2	MCC
0.050814	-0.11336	0.101145	-0.06203	65535	-0.12175	0.333784
0.049782	-0.10651	0.092291	-0.06036	65535	-0.11529	0.30076
0.054341	-0.11485	0.103826	-0.06807	65535	-0.12276	0.331694
0.049286	-0.09677	0.080996	-0.05959	65535	-0.10433	0.349275
0.055919	-0.1193	0.109478	-0.07091	-0.37161	-0.12677	0.346227
0.046974	-0.0963	0.079869	-0.05594	65535	-0.1042	0.314589

The image textural features are extracted from the segmented image. The textural analysis method such as Spatial Gray Level Dependency Matrix, Surrounding Region Dependency Matrix, Gray Level Run-Length Matrix and Spatial Gray Level Difference Matrix are used to extract the fourteen Haralick features from the segmented image.

Table 1.4 shows the list of features selected by the feature selection algorithms such as Decision Relative Discernibility based reduction, Heuristic approach, Hu's algorithm, Quick Reduct and Variable Precision Rough Set, Genetic Algorithm and Ant Colony Optimization. The combined features such as ASM, VAR CON, COR, IDM, ENT, IMC2 and MCC are given as input to the BPN classifier.

Table 1.4 Selected features from feature selection algorithms

Algorithms	**Selected Features**
DRR based Algorithm	ASM, VAR
Hu's Algorithm	ASM, CON, COR, VAR, IDM
Heuristic Algorithm	ASM, MCC
Quick Reduct	ASM, COR, VAR
EABCO	ASM, COR, VAR, IDM
GA	ASM, VAR, ENT, IMC2
ACO	ASM, IDM, ENT, IMC2

XII. CONCLUSION

Textural features are extracted for classification of normal and abnormal image. 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 analysis, 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.

REFERENCES

[1]. Dr. K Revathy, Applying EM Algorithm for Segmentation of Textured Images, Proceedings of the World Congress on Engineering 2007 Vol I WCE 2007, July 2 - 4, 2007, London, U.K.

[2]. John Babu, Sridevi Rangu, Pradyusha Manogna, A Survey on Different Feature Extraction and Classification Techniques Used in Image Steganalysis, Journal of Information Security, July 2017.

[3]. Goljan, M., Fridrich, J. and Cogranne, Rich Model for Steganalysis of Color Images, IEEE International Workshop on Information Forensics and Security (WIFS), 185-190 2014.

[4]. Sachnev, V., Ramasamy, S., Sundaram, S., Kim, H.J. and Hwang, H.J., A Cognitive Ensemble of Extreme Learning Machines for Steganalysis Based on Risk-Sensitive Hinge Loss Function. Cognitive Computation, 7, 103-110, 2015.

[5]. M.N.Sudha, S.Selvarajan, Feature Selection Based on Enhanced Cuckoo Search for Breast Cancer Classification in Mammogram Image, Apr 2016.

[6]. Michael D. Noseworthy, Texture feature based automated seeded region growing in abdominal MRI segmentation Jie Wu.

[7]. Bahía Blanca, Texture Filters and Fractal Dimension on Image Segmentation, Texture Filters and Fractal Dimension on Image Segmentation, Journal of Signal and Information Processing, 2018, 9, 229-238.

[8]. Reham G. Garout, Howayda M. Ahmed, Saddig D. Jastaniah*, Ibrahim A. Awad, Magnetic Resonance Imaging for Screening of Woman at High-Risk of Breast Cancer, Advances in Breast Cancer Research, 2014, 3, 59-67 Published Online July 2014 in SciRes.

[9]. Hamayun A. Khan, DM-L Based Feature Extraction and Classifier Ensemble for Object Recognition, Journal of Signal and Information Processing Vol.9 No.2，May 31, 2018

[10]. R. Nithya and B. Santhi, Application of texture analysis method for mammogram density classification, Journal of Instrumentation, Volume 12, July 2017

[11]. Karnan M, Thangavel K, Sivakuar R, Geetha K. Ant colony optimization for feature selection and classification of microcalcifications in digital mammograms. Advanced Computing and Communications. ADCOM: Surathkal, Heidelberg; 2006

[12]. Huang H, Xie HB, Guo JY, Chen HJ. Ant colony optimization-based feature selection method for surface electromyography signals classification. Comput Biol Med. 2012;42:30–38.

[13]. Benyamin Ghojogh, Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review

[14]. E. Alba, J. Garcia-Nieto, L. Jourdan, and E.-G. Talbi, “Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms,” in Evolutionary Computation, 2007. CEC 2007. IEEE Congress on. IEEE, 2007, pp. 284–290

[15]. M. Ciesielczyk, “Using mutual information for feature selection in programmatic advertising,” in INnovations in Intelligent SysTems and Applications (INISTA), 2017 IEEE International Conference on.IEEE, 2017, pp. 290–295.