This chapter clearly discusses the feature selection method, and its important role in River ice classification. This phase focuses on the dimensionality reduction problem to enhance best feature subset selection which in turn increase classification accuracy for River ice types. River ice classification is widely used for remote sensing application to monitor the conditions of the weather prediction and to guide the people staying around the area of River ice formation. The main objective of this step is to remove irrelevant and noise features to solve course dimensionality problem and to enhance the classification accuracy of the River ice image samples. Feature selection can considerably enhance the comprehensibility of the resultant classifier models and frequently construct a model that simplifies better unnoticed points.

Several parameters can affect the performance of River ice classification to analyse the characteristics of a River ice types. Among them, feature extraction and representation of patterns can be considered as most important. Reduction of pattern dimensionality via feature extraction and selection is one of the most fundamental steps in image processing. One of the most important and crucial tasks in any pattern recognition system is to overcome the curse of dimensionality problem, which forms an inspiration for developing an efficient feature selection method. An improper feature selection approach would considerably affect the classification accuracy of image samples. In order to solve these problems, several number of feature selection algorithms have been proposed in earlier works to minimize the dimensionality problem and to improve classification accuracy. Feature selection methods not only lessen the cost of running a classification algorithm by diminishing the feature space, they can also offer a better classification model because of the statically favored feature space that better fits River ice classification.
Feature selection methods are categorized into filter, Wrapper and hybrid approaches. Most of the feature selection algorithms select features based on the Wrapper approach, which comprises of meta-heuristic techniques like Genetic Algorithm, Simulated Annealing, Ant Colony Optimization, and hybrid of them. Numerous meta-heuristic based feature selection techniques have been proposed in earlier works (Castro and Tsuzuki, 2008).

Global-search-based feature selection algorithms, or Meta heuristic algorithms, start a search with full features rather than a partial feature space to find a high quality solution. Many evolutionary algorithms have been used for feature selection, which include Genetic Algorithms and Swarm Algorithms. Swarm Algorithms include, in turn, Ant Colony Optimization (ACO) (Zhang and Hu, 2005), Bat Algorithm (BAT) (Nakamura et al., 2012), and Artificial Bee Colony (Karaboga et al., 2012). Yang and Honavar, (1998), proposed a standard Genetic Algorithm (GA) is simply used for feature selection. In some hybrid approaches (Kabir et al., 2009), hybrid GAs is used for feature selection through different local search operations. Recent feature selection approach (Wang et al., 2007) uses a global search involving a Particle Swarm Optimization (PSO) algorithm and rough sets. Self-Adaptive Differential Evolution (SADE) (Ghosh et al., 2013) is used for feature subset generation. Generated subsets are assessed by means of a Wrapper model where Fuzzy K-nearest neighbour classifier is taken into account. SADE approach also employs a feature ranking method, Relief F algorithm, for eliminating duplicate attributes.

Among the entire feature selection methods available in the literature, two optimization methods, namely, Genetic Algorithm (GA) and Differential Evolution Feature Selection (DEFS) are chosen because of the following reasons:-

- Genetic algorithm does not have a mathematical requirements about the optimization problems, it can handle any type of objective function, in this work, Joint Conditional Entropy (JCE) is selected as fitness function,

- The periodicity of evolution operators enables GA very efficient at performing global search, and
• The time required to reach convergence for feature selection is less in DEFS.

The interest in River ice image characteristics and behaviour is derived from their roles in areas such as global climate monitoring and navigation. With the development of remote sensing techniques, a vast amount of River ice imagery is being provided by satellite platforms. As an important aspect of measurement, monitoring and understanding of River ice types during the winter and other seasons, the generation of ice type is a fundamental step in the interpretation of this research work.

This chapter clearly discusses the segmentation process using Gabor filter and Fuzzy C Means clustering by Zainab, M. Hussain (2013). Then, feature extraction process is carried out through First Order Statistical and Second Order Statistical such as GLCM, GLRLM and GLDM. Classification is performed for extracted features using PNN. Finally, feature selection process using Genetic Algorithm (GA) and Differential Evolution Feature Selection (DEFS) is carried out. This chapter mainly focuses on the feature selection aspect, as most of the existing River ice type classification methods fail to select important features before performing the classification tasks. The proposed feature selection methods select optimal features from the original feature set and it is further used for classification. The proposed feature selection and classification approach consists of the following steps:

• Segmentation using Gabor filter and Fuzzy C Means (FCM) Clustering,
• Feature extraction using First Order Statistical and Second Order Statistical such as, GLCM, GLRLM and GLDM,
• Classification using Probabilistic Neural Network (PNN) Classifier, and
• Proposed Feature selection approach using Genetic Algorithm (GA) and Differential Evolution Feature Selection (DEFS).

4.1. River Ice Image Segmentation using Gabor Filter and Fuzzy C Means (FCM) Clustering

Over the past few decades, image segmentation technique is gaining importance in pattern recognition. Image segmentation is an important and challenging
problem in image analysis. It is widely used in applications like high-level image interpretation, robot vision, object recognition and pattern recognition. The objective of image segmentation is to segment an image into a collection of disjointed areas with consistent and homogeneous attributes such as texture, intensity, colour, tone, etc.

An imperative job in image processing and machine vision is the process of segmenting regions of different texture in an image. Texture is a significant characteristic of surfaces which differentiates the nature of the surface. Segmentation of textured images is an elementary complication in image processing and pattern recognition, which has been observed for several applications with numerous techniques such as remote sensing, military surveillance, medical imaging, cartography, robot vision and assessment of textile products. Therefore, it is important to follow an effective segmentation technique for texture based image segmentation that is able to partition a textured real River ice sample images into multiple segments based on the textures, accurately and effectively.

In this research work, Gabor filter and cluster based method is considered for texture segmentation. Clustering helps in grouping the River ice textures, which are similar to one another. Many clustering strategies are available in the literature which includes hard clustering, soft clustering, fuzzy based clustering, etc., each of which has its own special characteristics. The traditional hard clustering process controls each point of the data set to completely just one cluster. Accordingly, with this scheme the segmentation results are typically very crisp, i.e., each pixel of the image belongs to accurately just one class. However, in many real scenarios, for images, issues such as limited spatial resolution, poor contrast, overlapping intensities, noise and intensity in homogeneities variation make this hard clustering (crisp) based segmentation a difficult task.

Fuzzy clustering (Krassimir and Atanassov, 2012), as a soft segmentation method, has been widely studied and successfully applied in many image segmentation applications (Nadernejad and Sharifzadeh, 2013). In particular, FCM has extensively
been employed to solve the segmentation problem and such an achievement, primarily attributes to introduction of fuzziness for the belongingness of each image pixel, which enables the clustering approaches to preserve more texture information from the original image than the hard or crisp segmentation (Choong et al., 2012). Moreover, FCM algorithm is “fuzzy relative” to the simple C-Means technique.

Figure 4.1: Gabor Filter and FCM Clustering based Texture Image Segmentation
The abundant texture information in River ice imagery is useful for segmentation of the pertinent ice types. The problem of texture segmentation involves subdividing an image into different texture regions. For this, Gabor filter and FCM clustering procedure is used. The results are notably distinct for different textured regions. Texture regions are extracted from an image by using Gabor filter method. The resultant image obtained from Texture segmentation has a variety of sub-regions of textures with different colours for each texture, and these colour regions are extracted as separate images by using color-based texture segmentation. Fuzzy C Means (FCM) clustering is implemented to segment different texture regions to form different clusters of a given image. The proposed Gabor filter and FCM based texture image segmentation for River ice images is diagrammatically presented in above Figure 4.1.

The color texture image segmentation of the proposed method consists of two main steps as follows:

1. Texture regions are extracted from an image by using Gabor filter method, and
2. Color-based texture segmentation is carried out using Fuzzy C Means (FCM) clustering.

- Decorrelation stretching, and
- FCM clustering for color texture segmentation.

4.1.1. Gabor Filter Method for Texture Region Extraction

As discussed earlier, Gabor filters have the capability to carry out multi-resolution decomposition because of its localization both in spatial and spatial frequency domain. Texture segmentation requires simultaneous measurements in both the spatial and the spatial-frequency domains for River ice images. Filters with lesser bandwidths in the spatial-frequency domain are more desirable for the reason that they facilitate finer differences among different textures. Alternatively, precise localization of texture boundaries necessitates filters that are localized in the spatial domain. On the other hand, usually, the effective width of a filter in the spatial domain and its bandwidth in
the spatial-frequency domain are inversely associated, in accordance with the uncertainty principle. Hence, Gabor filters are well suited for this kind of problem. Gabor method is used for detecting the discontinuity in the filters output and their statistical properties, which helps in segmentation.

A Gabor function in the spatial domain is a sinusoidal modulated Gaussian. For a 2-D Gaussian curve with a spread of $\sigma_x$ and $\sigma_y$ in the $x$ and $y$ directions, respectively, and a modulating frequency of $u_0$, the real impulse response of the filter is given by

$$h(x, y) = \frac{1}{2\pi \sigma_x \sigma_y} \exp\left\{-\frac{1}{2} \left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\} \cos(2\pi u_0 x) \quad (4.1)$$

In case of the spatial-frequency domain, the Gabor filter happens to be two shifted Gaussians at the place of the modulating frequency. The equation of the 2-D frequency response of the filter is as follows:-

$$h(u, v) = \exp\left\{-2\pi^2[\sigma_x^2(u - U) + \sigma_y^2 v^2]\right\} + \exp\left\{-2\pi^2[\sigma_x^2(u + U)
+ \sigma_y^2 v^2]\right\}$$

where $$[(u - U)', (v - V)'] = [(u - U) \cos \theta + (v - V) \sin \theta, -(u - U) \sin \theta + (v - V) \cos \theta] \quad (4.3)$$

Thus, from equation 4.2, Gabor Elementary function (GEF) frequency response has the shape of a Gaussian. The Gaussian's major and minor axis widths are determined by $\sigma_x \sigma_y$ and rotated by an angle $\theta$ with respect to the positive $u$-axis, and it is centred about the frequency $(U, V)$. Thus, the GEF acts as a band-pass filter. In most cases, letting $\sigma_x = \sigma_y = \sigma$ is a reasonable design choice and the filters focus on particular range of frequencies. When an input image has two different texture regions, the local frequency differences among the areas will identify the textures in one or more filter output sub-images. Each Gabor filter is specified by a GEF and these GEFs can carry out joint space decomposition. GEFs were first employed by Gabor and later extended to 2-D (Daugman et al., 2011). A GEF is given by
\[ h(x, y) = g(x', y') \exp[j2\pi(U_x + V_y)] \] (4.4)

where \( (x', y') = (x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta) \), indicate rotated spatial-domain rectilinear co-ordinates, if it assumed that \( \sigma_x = \sigma_y = \sigma \). Subsequently the parameter \( \theta \) is not essential and the equation of GEF (4.4) simplifies to

\[ h(x, y) = \frac{1}{2\pi \sigma^2} \exp \left\{ \frac{-x^2 + y^2}{2\sigma^2} \right\} \exp[j2\pi(U_x + V_y)] \] (4.5)

Gabor Filter \( O_h \) is defined by

\[ m(x, y) = O_h(i(x, y)) = |i(x, y) \otimes h(x, y)| \] (4.6)

where \( i \) is the input image and \( m \) is the output image.

### 4.1.2. Enhancing Color Texture Segmentation using FCM Clustering

Enhancing the color separation of River ice image samples for texture feature using decorrelation stretching is carried out and then the regions are grouped into a set of classes using Fuzzy C Means clustering algorithm. By means of this two step process, it is likely to lessen the computational cost circumventing feature calculation for each pixel in the image. Even though the color is not often employed for image segmentation, it provides a high discriminative power of regions exists in the image.

- **Decorrelation stretching**

Decorrelation stretching improves the color partition of an image with considerable band-to-band correlation. The exaggerated colors enhance visual interpretation and construct feature discrimination simpler. The number of spectral bands (NBANDS) in the color image is taken as three bands (RGB). The original color values of the image are plotted to a new collection of color values with an extensive range. The color intensities of each pixel are transformed into the color Eigen space of the NBANDS-by-NBANDS covariance or correlation matrix, stretched to equalize the
band variances, and then transformed back to the original color bands. In order to describe the band-wise information, the complete original image is employed with the subset option, or with any chosen subset of it. The basic aim is to segment colors in an automated manner by means of the $L*a*b*$ color space and Fuzzy c means clustering.

- **FCM Clustering for color texture segmentation**

The standard fuzzy c-means objective function for partitioning color space samples $\{c\{x_k\}_k\}^N_{k=1}$ into $c$ number of clusters is given by

$$J(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} F_{U_{ik}}^p ||c\{x_k\}_i - v_i||^2$$

where $\{c\{x_k\}_k\}^N_{k=1}$ the texture color space feature vector for every pixel are, $\{v_i\}^c_{i=1}$ the prototypes of the clusters and the array $[F_{U_{ik}}] = FU$ indicates a partition matrix specifically

$$\sum_{i=1}^{c} F_{U_{ik}} = 1 \quad 0 \leq F_{U_{ik}} \leq 1, \forall k = 1,2,3,...N$$

$$0 \leq \sum_{k=1}^{N} F_{U_{ik}} \leq 1$$

The parameter $p$ represents a weighting exponent on each fuzzy membership and decides the quantity of fuzziness of the resultant segmentation. The FCM objective function is reduced when the high membership values are allocated to pixels whose intensities are nearly the centroid of their specific class, and low membership values are allocated when the pixel data is distant from the centroid. The constrained optimization could be solved using one Lagrange multiplier

$$F_m = \sum_{i=1}^{c} \sum_{k=1}^{N} F_{U_{ik}}^p \parallel c\{x_k\}_i - v_i \parallel^2 + \lambda(1 - \sum_{i=1}^{c} F_{U_{ik}})$$

where $\lambda$ denotes a Lagrange multiplier. The derivative of $F_m$ with respect to $F u_{i k}$ is computed and the result is set to zero, for $p > 1$

$$\frac{\partial F_m}{\partial F u_{i k}} = p F u_{i k}^{p-1} \|c x_k - v_i\|^2 + \lambda$$  \hspace{1cm} (4.11)$$

$$F u_{i k} = \left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \frac{1}{\|c x_k - v_i\|^{\frac{2}{m-1}}}$$  \hspace{1cm} (4.12)$$

The identity constraint $\sum_{i=1}^{c} F u_{i k} = 1 \forall k$ was taken into consideration.

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \sum_{j=1}^{c} \frac{1}{\|c x_k - v_i\|^{\frac{2}{m-1}}} = 1$$  \hspace{1cm} (4.13)$$

This assists in determining the Lagrange multiplier $\lambda$.

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{j=1}^{c} \frac{1}{\|c x_k - v_i\|^{\frac{2}{m-1}}} = 1}$$  \hspace{1cm} (4.14)$$

The zero-gradient state for the membership estimator can be given as

$$F u_{i k} = \frac{1}{\sum_{j=1}^{c} \left(\|c x_k - v_i\|^{p-1}\right)^{\frac{2}{m-1}}}$$  \hspace{1cm} (4.15)$$

Seeing no constraints, the calculations of the prototypes were uncomplicated, the minimum of $J$ was calculated with regard to $v_i$, and the following equation is constructed as

$$\nabla_{v_i} J = 0$$  \hspace{1cm} (4.16)$$

The comprehensive solution based on the distance function. In the scenario of the Euclidean distance, this leads to the expression
The proposed color texture segmentation method includes the following major steps:

**Algorithm : Texture Segmentation**

**Step 1:** The original River ice image is taken as input.

**Step 2:** The input image is transformed into a 2-D matrix.

**Step 3:** Initialized $U$, $V$, $\sigma_x$, $\sigma_y$ Values.

**Step 4:** The value of $h$ is calculated using Eq. (4.5).

**Step 5:** The texture segmented output image $m(x,y)$ is obtained by the convolution of $i(x,y)$ with $h(x,y)$ using equation (4.6).

Step 6: repeat step 3 to step 5 until the textures are clearly discriminated

**Step 7:** Result from the texture segmentation is saved and then it is provided as the input to the color segmentation.

**Step 8:** The Decorrelation stretching technique is applied for colour separation of an image.

**Step 9:** The image has transformed from RGB Color Space to $L*a*b*$ Color Space.

**Step 10:** The Colors in $'a*b*' $Space is segmented using FCM clustering.

**Step 10.1:** The initial class prototype $\{v_i\}_{i=1}^{C}$ is selected

**Step 10.2:** All memberships $F_{U_{ik}v}$ is updated with Eq. (4.15).

**Step 10.3:** The prototype of clusters is obtained in the forms of weighted average with Eq. (4.18).
**Step 10.4:** Step 10.2-10.3 is repeated till termination. The termination criterion is $||W_{new} - V_{old}|| \leq \varepsilon$.

where $||.||$ is the Euclidean norm, $V$ is the vector of cluster centers, $\varepsilon$ is a small number that can be set by user (here $\varepsilon = 0.01$).

**Step 11:** Each pixel in the Image is labelled using the results from FCM cluster.

From the above mentioned steps, texture segmentation of River ice image samples are carried out using Gabor filtering and FCM Clustering algorithm. The results of the proposed FCM Clustering and Gabor filtering method is shown in Figure 4.2 (a) – (d) along with the separation of texture clusters into different images.

![Figure 4.2. (a)–(d): Texture Segmentation result from Original Images without Denoising](image)
4.2. Feature Extraction

A feature is described as an “interesting” component of an image, and is utilized as a starting point in key primitives for subsequent algorithms. The approaches employed in these systems are normally segmented into three tasks: extraction, selection and classification. For a suitable classification, there has to be a balanced nexus between the features, and it is the most significant assignment, because, the particular features made accessible for discrimination openly influence the efficacy of the classification task. The outcome of feature extraction is a collection of features, normally called a feature vector, which comprise a representation of an image.

Texture analysis aims at, judging a distinctive method of representing the fundamental features of textures and characterize them in some simpler but unique form, in order that they can be utilized for robust and precise classification. Despite the fact that texture plays a major part in image analysis and pattern recognition, only a few architectures apply on-board textural feature extraction.

Feature extraction engages simplifying the quantity of resources, necessary to characterize a large set of data accurately. When carrying out investigation of complex data, one of the major complications stems from the number of variables involved. Investigation with a huge number of variables normally necessitates a large quantity of memory, and computation power or a classification approach, which over fits the training sample and simplifies poorly to new samples. Feature extraction is a common term for methods of building combinations of the variables to obtain around these complications at the same time, still describing the data with adequate accuracy.

4.2.1. First Order Statistical and Second Order Statistical Feature Extraction Methods

In statistical texture examination, texture features are calculated from statistical distribution of observed combinations of intensities at particular positions comparative to each other in the image. Based on the number of intensity points (pixels) in every combination, statistics are categorized into first-order, second-order and higher-order statistics.
In this research, the segmented images are given as input for feature extraction. The purpose of feature extraction phase is to reduce the original image into a set of features, by measuring certain properties that distinguish one input pattern from another pattern between the River ice types. The feature extraction step functions to discover various features that best represent River ice images. In feature extraction, two categories of features are extracted. They are

- First Order Statistical, and
- Second Order Statistical Method.

Second order statistical feature extraction methods are, Gray Level Co-occurrence Matrix (GLCM) (Naseri et al., 2012), Gray Level Run Length Matrix (GLRLM) (Sohail ASM et al., 2011) and Gray Level Difference Matrix (GLDM). In most of the studies related to ice classification, only GLCM method is used. In this work, First Order Statistical and Second Order Statistical methods are considered. The list of various features extracted under each category is shown in Table 4.1.

**Table 4.1: Statistical Features Extracted from River Ice Images**

<table>
<thead>
<tr>
<th>Statistical Methods</th>
<th>Features Extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Order Statistical</td>
<td>Mean, Standard Deviation, Variance, Skewness, Kurtosis</td>
</tr>
</tbody>
</table>
| Second order statistical | GLCM  
Autocorrelation, Contrast, Correlation, Energy, Entropy,  
Homogeneity, Sum of Squares, Sum Average, Sum Variance,  
Sum Entropy, Difference entropy, Difference variance,  
Information measures of correlation (1), Information measures of correlation (2), Inverse difference, Inverse difference normalized, Inverse difference moment normalized |
| Second order statistical | GLRLM  
Short-run emphasis (SRE), Long-run emphasis (LRE), Gray level Non-uniformity (GLN), Run Percentage (RP), Run-Length Non-uniformity (RLN), Low gray-level emphasis (LGRE), High Gray-level emphasis (HGRE) |
| Second order statistical | GLDM  
Large Difference Emphasis (LDE), Sharpness (SHP), Second Moment of distribution of gray level differences (SMG),  
Second Moment of distribution of the average gray level difference (SMO), and Long Distance Emphasis for Large difference (LDEL) |
The four feature categories were first analyzed for their efficiency in ice type identification through ice class types from River ice images. The feature extraction results for River ice images are illustrated in Figure 4.3.

**Figure 4.3: First Order and Second Order Statistical Feature Extraction Methods**
4.2.1.1 First Order Statistical Method

First Order Statistical feature computes the probability of observing a gray value at an arbitrarily selected location in the image. First Order Statistical can be calculated from the histogram of pixel intensities in the image. These are based only on individual pixel values and not in accordance with the interaction or co-occurrence of neighbouring pixel values. The average intensity in an image is an example of the first order statistic (Tuceryan et al., 1998). The reason behind this is the reality that the spatial distribution of gray values is one of the significant aspects of texture. First Order Statistical texture measures calculated are mean, energy, variance, kurtosis and Skewness, which are directly calculated from the images.

Mean

Standard deviation or variance reveals the contrast of an image. Image with good contrast should have high variance. Standard Deviations (SD) also characterize the cluster.

\[
mean: \mu = \sum_{i=1}^{L} k_ip(k_i)
\]

Variance

Variance shows the variations of an image.

\[
\text{variance: } \sigma^2 = \sum_{i=1}^{L} (k_i - \mu)^2 p(k_i)
\]

Skewness

Skew measures is how irregularity (unbalance) the distribution of the gray level.

\[
\text{skewness: } \mu_3 = \sigma^{-3} \sum_{i=1}^{L} (k_i - \mu)^{-3} p(k_i)
\]

Kurtosis

It indicates the level of sharpness relatively curve of an image.
\begin{equation}
Kurtosis: \mu_4 = \sigma^{-4} \sum_{i=1}^{L} (k_i - \mu)^{-4} p(k_i) - 3
\end{equation}

Standard deviation

\begin{equation}
\text{standard deviation} = \sqrt{\text{variance}}
\end{equation}

where $k_i$ = gray value of the $i^{th}$ pixel, $L$ = number of distinctive gray levels, $p(k_i)$ normalized texture feature gray level value.

\begin{equation}
p(k_i) = \frac{\text{Number of pixels with gray level of } I}{\text{Total number of pixels in the region}}
\end{equation}

From the above mentioned steps, the First Order Statistical (FOS) features are extracted for the texture segmented River ice image samples. Initially, FOS feature extraction method extracts five features, namely, mean, standard deviation, variance, Kurtosis and Skewness. The statistical features extracted from FOS method are shown in Table 4.2. It is clearly shown that totally 5 features are extracted from FOS methods.

**Table 4.2: Statistical Features Extracted from First Order Statistical (FOS) Method without Denoising**

<table>
<thead>
<tr>
<th>Images</th>
<th>Mean</th>
<th>Variance</th>
<th>Standard deviation</th>
<th>Skewness</th>
<th>Kurtosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Img1</td>
<td>552.25</td>
<td>34439227</td>
<td>5868.4945</td>
<td>11.9778</td>
<td>149.8484</td>
</tr>
<tr>
<td>Img2</td>
<td>624.43</td>
<td>36349685</td>
<td>5718.9354</td>
<td>12.6101</td>
<td>169.9747</td>
</tr>
<tr>
<td>Img3</td>
<td>892.52</td>
<td>65055171</td>
<td>8065.6786</td>
<td>11.0506</td>
<td>125.0509</td>
</tr>
<tr>
<td>Img4</td>
<td>552.25</td>
<td>30702511</td>
<td>5540.9846</td>
<td>12.2418</td>
<td>158.4572</td>
</tr>
<tr>
<td>Img5</td>
<td>892.52</td>
<td>85315439</td>
<td>9236.6357</td>
<td>11.3987</td>
<td>133.0112</td>
</tr>
<tr>
<td>Img6</td>
<td>552.25</td>
<td>31720512</td>
<td>5632.0966</td>
<td>11.1445</td>
<td>125.4652</td>
</tr>
<tr>
<td>Img7</td>
<td>892.52</td>
<td>58869510</td>
<td>7672.6469</td>
<td>11.8615</td>
<td>150.3301</td>
</tr>
<tr>
<td>Img8</td>
<td>601.87</td>
<td>34135178</td>
<td>5649.7634</td>
<td>11.8700</td>
<td>150.8522</td>
</tr>
<tr>
<td>Img9</td>
<td>892.52</td>
<td>58869510</td>
<td>7672.6469</td>
<td>11.8615</td>
<td>150.3301</td>
</tr>
<tr>
<td>Img10</td>
<td>892.52</td>
<td>110857988</td>
<td>10528.9120</td>
<td>15.5171</td>
<td>245.3835</td>
</tr>
</tbody>
</table>
4.2.1.2 Second Order Statistical Method

The features generated from the FOS method offers information associated with the gray-level distribution of the image. On the other hand, they do not provide any information regarding the relative positions of the several gray levels inside the image. These characteristics will not be capable of measuring whether the entire low-value gray levels are positioned mutually, or they are interchanged with the high-value gray levels. As a result, Second Order Statistical feature extraction methods are used to consider the relative positions of the several gray levels within the images.

A. Gray Level Co-Occurrence Matrices (GLCM)

One of the recognized statistical methods for extracting texture feature information from images is by using GLCM method. GLCM computes second order texture features, which play a significant role in human vision and has been found to realize a similar level of classification performance. This approach is normally employed in texture analysis, since it provides for each sample a huge collection of features, and it can be presumed that at least one of these features reflects the minute variation of texture between classes. GLCM of an $N_x \times N_y$ image, includes pixels with gray levels $(0, 1, \ldots, G - 1)$ is a two dimensional matrix $P(i,j)$, where each element of the matrix indicates the probability of joint occasion of intensity levels $k$ and $l$ at a particular distance $d$ and an angle $\theta = 0$. The following features are calculated from the co-occurrence matrix for $d = 1$ and zero angle (0°). Before building the matrix, two parameters $\theta = 0^\circ$ (direction of the pixel pairs) and $d$ (distance between the pixel pairs) which is shown in Figure 4.4, required to be selected by the user. GLCM is obtained by specifying a matrix of relative frequencies $P_{ij}$, with two neighboring resolution cells separated by distance ‘$d$’ taking place on the image, one with gray tone $i$ and the other with gray tone $j$. In general, the matrices $P_{\theta i,j | d}$, where $\theta = 0^\circ$, $d$ ranges from 1 to $d_{max}$ and $i,j$ are taken over all gray levels (Chappard et al., 2003). In general, before exploiting a single displacement one uses a set of displacements to acquire the required property to which the textural feature corresponds. From the co-occurrence matrix 17 texture features are extracted. On the other hand, these 17 features are not free of each
other as few of them might point towards the same image texture properties. Unfortunately, it is still unclear which feature can be ignored.

Figure 4.4: The Direction ($\theta$) of the pixel pairs and the Distance ‘d’ between the pixel pairs used to construct the GLCM Feature Set

Illustration shows, direction ($\theta$) and distance ($d$) in the images are calculated by pixel pairs $(x_1, y_1)$ and $(x_2, y_2)$, with $\theta = 0^\circ$ and $d = 2$. GLCM features extracted are as shown below:

**Contrast (CON)**

Contrast is the chief diagonal close to the moment of inertia, which determines how the values of the matrix are allocated and number of images of local transformations reflecting the image clearness and texture of shadow deepness. Large Contrast represents deeper texture.

$$CON = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \right\}$$  \hspace{1cm} (4.25)

**Correlation (CORR)**

Correlation is a measure of gray level linear dependence among the pixels at the particular locations relative to each other.

$$CORR = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i,j)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$  \hspace{1cm} (4.26)
\[
\mu_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) \\
\mu_y = \sum_{j=1}^{N_g} \sum_{i=1}^{N_g} p(i,j) \\
\sigma_x = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu_x)^2 p(i,j) \\
\sigma_y = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (j - \mu_x)^2 p(i,j) \\
\]

where \(\mu_x\), \(\mu_y\) are the mean values and \(\sigma_x\), \(\sigma_y\) are the standard deviations of \(P_x\) and \(P_y\), respectively.

**Energy (ENER)**

This value is also known as Uniformity or Angular second moment. It computes the textural consistency, that is, pixel pair recurrences. It identifies disorders in textures. Energy reaches a maximum value equal to one. High energy values take place when the gray level distribution has a steady or periodic form. Energy has a normalized range. The GLCM of less homogeneous image will have huge number of small entries.

\[
ENER = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j)^2
\]

**Entropy (ENT)**

Entropy is a tricky term to describe. The conception comes from thermodynamics, and indicates to the quantity of energy that is enduringly lost to heat every time a reaction or a physical transformation takes place. Entropy cannot be recovered to do constructive work. As a consequence of this, the term can be understood as quantity of irremediable chaos or disorder.
\[ \text{ENT} = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} [P(i,j) \log P(i,j)] \]  

**Inverse Difference Moment (IDM)**

IDM is typically called homogeneity that computes the local homogeneity of an image. IDM feature acquires the measures of the proximity of the distribution of the GLCM elements to the GLCM diagonal. IDM weight value is the contrary of the contrast weight, with weights declining exponentially away from the diagonal.

\[ IDM = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \left| \frac{1}{1 + (i-j)^2} P(i,j) \right| \]  

**Sum of Squares (SOS)**

Sum of squared deviates of the images is indicated as the estimated variance of the source population.

\[ SOS = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu)^2 p(i,j) \]  

**Sum Average (SA)**

Sum average is the average of normalized gray tone image in the spatial domain, respectively

\[ SA = \sum_{i=2}^{2N_g} \frac{1}{P_{x+y}(i)} \]  

**Sum Variance (SV)**

This characteristic puts comparatively high weights on the elements that differ from the average value of \( P(i,j) \).

\[ SV = \sum_{i=2}^{2N_g} [(i - SA)^2 P_{x+y}(i)] \]
Sum Entropy (SE)

The sum entropy is a measure of randomness within an image and it is given as

\[
SE = - \sum_{i=2}^{2N_g} \left[ P_{x+y}(i) \log P_{x+y}(i) \right]
\]

(4.37)

Difference Variance (DV)

The difference variance is an image variation in a normalized co-occurrence matrix.

\[
DV = - \sum_{i=0}^{N_g-1} [(i - f') P_{x-y}(i)] \quad \text{where} \quad f' = \sum_{i=0}^{N_g-1} [i P_{x-y}(i)]
\]

(4.38)

Difference Entropy (DEn)

The difference entropy is also an indication of the amount of randomness in an image.

\[
DEn = - \sum_{i=0}^{N_g-1} [P_{x-y}(i)] \log [P_{x-y}(i)]
\]

(4.39)

Autocorrelation (AUTO_CORR)

The autocorrelation task, which is calculated along the horizontal and vertical axes of the analysis window \( I \) of an image in accordance with the following equation:

\[
R_{(x,y)}^{I(\alpha,\beta)} = \sum_{x} \sum_{y} I(x, y) I(x + \alpha, y + \beta)
\]

(4.40)

where \( I(x + \alpha, y + \beta) \) represents the translation of the analysis window of an image \( I(x, y) \) by \( \alpha \) and \( \beta \) pixels along the horizontal and vertical axes correspondingly, defined on the plane \( \Omega \).

Homogeneity (HOMOG)

The homogeneity measure (H) estimates the homogeneity accuracy in terms of matching areas among the ground truth and result regions, i.e., spatial overlaps of the label association between the ground truth and the result regions.
\[ HOMOG = \sum_{i,j} \frac{p(i,j)}{1 + |i-j|} \]  
\hspace{1cm} (4.41)

**Information Measures of Correlation (1) (IMC_1)**

Extract Information Measure of Correlation 1 property of the input GLCM.

\[ IMC_1(1) = \frac{H(X,Y) - HXY_1}{\max \{HX, HY\}} \]  
\hspace{1cm} (4.42)

**Information Measures of Correlation (2) (IMC_2)**

Extract Information Measure of Correlation 2 property of the input GLCM

\[ IMC_2 = \sqrt{(1 - \exp[-2.0(HXY^2 - HXY)])} \]  
\hspace{1cm} (4.43)

**Inverse Difference (Inv_Diff)**

It computes number of local transformations in image texture. Its value in large is demonstrated that image texture among the different areas of the lack of transformation and partial very evenly. Here \( p(x,y) \) is the gray-level value at the coordinate \((x,y)\).

\[ Inv_Diff = \sum_{x} \sum_{y} \frac{1}{1 + (x-y)^2} p(x,y) \]  
\hspace{1cm} (4.44)

**Inverse Difference Normalized (In_Df_Nrm)**

\[ In_Df_Nrm = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{P(i,j)}{1 + |i-j|^2} \]  
\hspace{1cm} (4.45)

Second order statistical feature extraction methods are consider to extract the relative positions of the several gray levels within the images. In Second Order Statistical (SOS), GLCM method is used to extract texture feature information by calculation of co-occurrence matrix. Based on the above process, 17 different texture features are extracted for selected images which are shown in Tables 4.3.
Table 4.3: Statistical Features Extracted from GLCM Method without Denoising

<table>
<thead>
<tr>
<th>Image</th>
<th>Auto _Cor</th>
<th>Cont</th>
<th>Corr</th>
<th>Ener</th>
<th>Ent</th>
<th>Homog</th>
<th>Sum _Sq</th>
<th>Sum _Av</th>
<th>Sum _En</th>
<th>Dif _Vr</th>
<th>Dif _Ent</th>
<th>IMC_1</th>
<th>IMC_2</th>
<th>Inv _Diff</th>
<th>In_Df_ Nrm</th>
<th>In_Df_ Mom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Img1</td>
<td>10.83</td>
<td>0.07</td>
<td>0.99</td>
<td>0.49</td>
<td>0.85</td>
<td>0.99</td>
<td>10.78</td>
<td>5.28</td>
<td>35.34</td>
<td>0.83</td>
<td>0.07</td>
<td>0.13</td>
<td>-0.89</td>
<td>0.86</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Img2</td>
<td>7.57</td>
<td>0.40</td>
<td>0.94</td>
<td>0.48</td>
<td>1.30</td>
<td>0.92</td>
<td>7.70</td>
<td>4.22</td>
<td>22.23</td>
<td>1.16</td>
<td>0.40</td>
<td>0.52</td>
<td>-0.60</td>
<td>0.82</td>
<td>0.92</td>
<td>0.98</td>
</tr>
<tr>
<td>Img3</td>
<td>10.93</td>
<td>0.07</td>
<td>0.99</td>
<td>0.49</td>
<td>0.87</td>
<td>0.99</td>
<td>10.88</td>
<td>5.31</td>
<td>35.53</td>
<td>0.85</td>
<td>0.07</td>
<td>0.14</td>
<td>-0.88</td>
<td>0.87</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Img4</td>
<td>11.82</td>
<td>0.12</td>
<td>0.98</td>
<td>0.46</td>
<td>0.99</td>
<td>0.98</td>
<td>11.79</td>
<td>5.61</td>
<td>37.70</td>
<td>0.94</td>
<td>0.12</td>
<td>0.22</td>
<td>-0.83</td>
<td>0.87</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>Img5</td>
<td>4.96</td>
<td>0.24</td>
<td>0.95</td>
<td>0.70</td>
<td>0.82</td>
<td>0.96</td>
<td>5.03</td>
<td>3.24</td>
<td>15.86</td>
<td>0.74</td>
<td>0.24</td>
<td>0.34</td>
<td>-0.65</td>
<td>0.74</td>
<td>0.96</td>
<td>0.99</td>
</tr>
<tr>
<td>Img6</td>
<td>11.62</td>
<td>0.18</td>
<td>0.98</td>
<td>0.43</td>
<td>1.16</td>
<td>0.96</td>
<td>11.62</td>
<td>5.53</td>
<td>35.83</td>
<td>1.08</td>
<td>0.18</td>
<td>0.30</td>
<td>-0.76</td>
<td>0.87</td>
<td>0.96</td>
<td>0.99</td>
</tr>
<tr>
<td>Img7</td>
<td>11.36</td>
<td>0.08</td>
<td>0.99</td>
<td>0.48</td>
<td>0.90</td>
<td>0.98</td>
<td>11.32</td>
<td>5.46</td>
<td>36.78</td>
<td>0.87</td>
<td>0.08</td>
<td>0.16</td>
<td>-0.87</td>
<td>0.87</td>
<td>0.98</td>
<td>1.00</td>
</tr>
<tr>
<td>Img8</td>
<td>4.94</td>
<td>0.06</td>
<td>0.99</td>
<td>0.77</td>
<td>0.53</td>
<td>0.99</td>
<td>4.92</td>
<td>3.16</td>
<td>16.90</td>
<td>0.50</td>
<td>0.06</td>
<td>0.13</td>
<td>-0.84</td>
<td>0.73</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Img9</td>
<td>17.42</td>
<td>0.12</td>
<td>0.99</td>
<td>0.46</td>
<td>0.96</td>
<td>0.98</td>
<td>17.37</td>
<td>6.73</td>
<td>58.35</td>
<td>0.91</td>
<td>0.12</td>
<td>0.19</td>
<td>-0.85</td>
<td>0.87</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>Img10</td>
<td>6.92</td>
<td>0.27</td>
<td>0.96</td>
<td>0.62</td>
<td>0.98</td>
<td>0.95</td>
<td>7.00</td>
<td>3.81</td>
<td>22.05</td>
<td>0.88</td>
<td>0.27</td>
<td>0.37</td>
<td>-0.67</td>
<td>0.79</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>Img11</td>
<td>6.30</td>
<td>0.13</td>
<td>0.98</td>
<td>0.70</td>
<td>0.67</td>
<td>0.98</td>
<td>6.31</td>
<td>3.55</td>
<td>21.26</td>
<td>0.63</td>
<td>0.13</td>
<td>0.18</td>
<td>-0.80</td>
<td>0.77</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>Img12</td>
<td>9.98</td>
<td>0.27</td>
<td>0.97</td>
<td>0.50</td>
<td>1.18</td>
<td>0.95</td>
<td>10.04</td>
<td>4.70</td>
<td>31.28</td>
<td>1.07</td>
<td>0.27</td>
<td>0.40</td>
<td>-0.69</td>
<td>0.84</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>Img13</td>
<td>5.09</td>
<td>0.21</td>
<td>0.96</td>
<td>0.73</td>
<td>0.70</td>
<td>0.97</td>
<td>5.15</td>
<td>3.24</td>
<td>16.85</td>
<td>0.64</td>
<td>0.21</td>
<td>0.27</td>
<td>-0.70</td>
<td>0.73</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td>Img14</td>
<td>4.96</td>
<td>0.07</td>
<td>0.99</td>
<td>0.77</td>
<td>0.52</td>
<td>0.99</td>
<td>4.94</td>
<td>3.16</td>
<td>17.01</td>
<td>0.50</td>
<td>0.07</td>
<td>0.13</td>
<td>-0.84</td>
<td>0.73</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Img15</td>
<td>8.03</td>
<td>0.24</td>
<td>0.96</td>
<td>0.49</td>
<td>1.13</td>
<td>0.95</td>
<td>8.08</td>
<td>4.40</td>
<td>24.35</td>
<td>1.03</td>
<td>0.24</td>
<td>0.37</td>
<td>-0.70</td>
<td>0.84</td>
<td>0.95</td>
<td>0.99</td>
</tr>
</tbody>
</table>
B. Gray Level Run Length Matrices Method

A Gray Level Run Length Matrix (GLRLM) technique is a way of obtaining second order statistical texture measures. A collection of consecutive pixels with the similar gray level, collinear in a specified direction, comprise the gray level run. The run length is the amount of pixels in the run and the run length value is the amount of times such a run takes place in an image. The GLRLM is a two dimensional matrix in which each element $p(i, j | \theta)$ provides the entire number of occasions of runs of length “j” at gray level “i” in a specified direction $\theta$. Features extracted from GLRLM can be used to estimate the size distribution of the sub patterns. Two kinds of methods are used for processing the grey level pixel-run length. In the initial one, a vector considering pixel-runs are generated from the function $q(L, \theta, T)$, in which $L$ represents the length of the pixel-run while $\theta$ represents direction of the pixel run and $T$, the threshold. Direction $\theta$ of pixel-run is defined similar to that in the GLCM method. Threshold value $T$ for pixels to be merged into the pixel-run is given manually by the user. The procedure of constructing the pixel-runs is as follows: each pixel row of image at direction $\theta$ is scanned and the first pixel of the row is set to be the first pixel-run with length $L$ and same grey value $I$ as the first pixel; then the next pixel in the row is scanned; if $|I - I_n| \leq T$ ($I_n$ represents the grey value of the next pixel), the subsequent pixel is merged into the pixel-run, if not, a new pixel-run is generated and the pointer is moved to the next pixel. This procedure is carried out until the scanning of the entire row is completed, and a new row is started (Renzetti et al., 2011).

In the GLRLM approach, the gray level runs are exemplified by the gray tone of the run and the length of the run and the direction of the run. Let $P(i, j)$ represent the run length matrix array. The matrix array includes elements with the gray tone “i” has a run length “j”. Textural features are computed from the array constituents that are employed to investigate the nature of image textures. In accordance with the original run length matrix $P(i, j)$, several numerical texture measures can be computed. The five original features of run length statistics derived by (Mohd Yakop et al., 2006) are as follows:-
Short Run Emphasis (SRE)

Short Runs Emphasis (SRE) fragments each run length value by the length of the run squared. This tends to emphasize short runs. The denominator is the overall number of runs in the image and provides as a normalizing factor.

\[
SRE = \frac{1}{n_r} \sum_{j=1}^{N} \frac{P_r(j)}{j^2}
\]  

(4.46)

Long Run Emphasis (LRE)

Long Runs Emphasis (LRE) multiplies every run length value by the length of the run squared. This should emphasize long runs. The denominator is a normalizing factor, as mentioned above.

\[
LRE = \frac{1}{n_r} \sum_{j=1}^{N} P_r(j).j^2
\]

(4.47)

Gray-Level Non-uniformity (GLN)

It computes the similarity of gray level values all the way through the image. The GLN is expected small if the run lengths are alike throughout the image.

\[
GLN = \frac{1}{n_r} \sum_{i=1}^{M} P_g(i)^2
\]

(4.48)

Run Percentage (RP)

It computes the homogeneity and the distribution of runs of an image in a particular direction. RP is the largest when the length of runs is 1 for all gray levels in specific direction.

\[
RP = \frac{n_r}{n_p}
\]

(4.49)
Run Length Non-uniformity (RLN)

It computes the similarity of length of runs all the way through the image. The RLN is expected small if the run lengths are alike throughout the image.

\[
RLN = \frac{1}{n_r} \sum_{j=1}^{N} P_r(i)^2
\]  

(4.50)

Low Gray-Level Run Emphasis (LGRE)

LGRE measures the joint distribution of long runs and low gray level values. It is expected large for the image with many long runs and low gray level values

\[
LGRE = \frac{1}{n_r} \sum_{i=1}^{N} P_g(i) / i^2
\]  

(4.51)

High Gray-Level Run Emphasis (HGRE)

HGRE measures the joint distribution of short runs and high gray level values. It is expected large for the image with many short runs and high gray level values.

\[
HGRE = \frac{1}{n_r} \sum_{i=1}^{N} P_g(i) \cdot i^2
\]  

(4.52)

In the above equations, \(n_r\) represents the total number of runs and \(n_p\) represents the number of pixels in the image. Depending on the observation that most features are simply functions of \(P_r(j)\), without taking the gray level information contained in \(P_g(i)\) into account.

Second order statistical feature extraction methods are exploited to consider the relative positions of the several gray levels within the images. GLRLM is used to obtain the second order statistical texture by measuring similar gray level values. The similarity values of seven different features extracted are shown in Table 4.4.
Table 4.4: Statistical Features Extracted from GLRLM Method without Denoising

<table>
<thead>
<tr>
<th>Images</th>
<th>SRE</th>
<th>LRE</th>
<th>GLNU</th>
<th>RUN</th>
<th>RLNU</th>
<th>LGLRE</th>
<th>HGLRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Img1</td>
<td>0.58</td>
<td>725.87</td>
<td>195.91</td>
<td>0.07</td>
<td>387.58</td>
<td>0.19</td>
<td>139.55</td>
</tr>
<tr>
<td>Img2</td>
<td>0.74</td>
<td>117.21</td>
<td>597.36</td>
<td>0.33</td>
<td>2734.27</td>
<td>0.23</td>
<td>104.38</td>
</tr>
<tr>
<td>Img3</td>
<td>0.58</td>
<td>900.64</td>
<td>186.07</td>
<td>0.08</td>
<td>402.93</td>
<td>0.20</td>
<td>131.88</td>
</tr>
<tr>
<td>Img4</td>
<td>0.63</td>
<td>1075.96</td>
<td>159.26</td>
<td>0.08</td>
<td>468.74</td>
<td>0.27</td>
<td>103.14</td>
</tr>
<tr>
<td>Img5</td>
<td>0.70</td>
<td>135.76</td>
<td>415.45</td>
<td>0.23</td>
<td>1722.18</td>
<td>0.26</td>
<td>86.46</td>
</tr>
<tr>
<td>Img6</td>
<td>0.70</td>
<td>371.04</td>
<td>282.79</td>
<td>0.16</td>
<td>1191.37</td>
<td>0.21</td>
<td>112.87</td>
</tr>
<tr>
<td>Img7</td>
<td>0.58</td>
<td>639.76</td>
<td>188.42</td>
<td>0.09</td>
<td>454.39</td>
<td>0.21</td>
<td>119.92</td>
</tr>
<tr>
<td>Img8</td>
<td>0.64</td>
<td>1130.49</td>
<td>158.36</td>
<td>0.07</td>
<td>473.86</td>
<td>0.20</td>
<td>128.05</td>
</tr>
<tr>
<td>Img9</td>
<td>0.58</td>
<td>1592.00</td>
<td>130.34</td>
<td>0.06</td>
<td>309.39</td>
<td>0.26</td>
<td>107.21</td>
</tr>
<tr>
<td>Img10</td>
<td>0.72</td>
<td>204.66</td>
<td>349.23</td>
<td>0.19</td>
<td>1551.23</td>
<td>0.25</td>
<td>105.27</td>
</tr>
</tbody>
</table>

C. Gray Level Difference Matrix Method

The run difference method is a generalized form of the GLDM, which is based on the estimation of the probability density function of gray level differences in an image. GLDM looks to obtain texture features that describe the size and prominence of textural elements in an image. Consider \( I(x,y) \) be the image intensity function. For any given displacement \( \delta = (\Delta X, \Delta Y) \) let \( I\delta(x,y) = |I(x,y) - I(X + \Delta X, Y + \Delta Y)| \), and \( f(i|\delta) \) be the probability density of \( I\delta(x,y) \). The value of \( f(i|\delta) \) is got from the number of occasions \( I\delta(x,y) \) takes place for a given, i.e. \( f(i|\delta) = P(I\delta(x,y) = i) \). When a texture is directional, the degree of stretch of the values in \( f(i|\delta) \) must diverge with the direction of \( d \), specified that its magnitude is in the suitable range. As a result, texture directionality can be analyzed by evaluating spread measures of \( f(i|\delta) \) for various directions of \( d \). In the current work, four possible forms of the vector \( d \) were considered: \((0,d), (d, 0), (-d, d) \) and \((-d, -d)\), with \( d \) being the inter pixel distance, each of which corresponds to a displacement in \( (0^\circ) \) direction.
\[ I_{rgdif} = \sum_{\theta \in \Theta} I^\theta_{rgdif} \]  

(4.53)

From which statistical measures are obtained from the distribution of gray level differences. More willingly than obtaining textural features directly from the matrix \( I \), three characteristic vectors are computed to define texture descriptors. The distribution of gray level differences (DGD) vector is given as below:

\[ DGD_j = \sum_{r=1}^{g} I_{grdif} \]  

(4.54)

The distribution of the average gray level difference specified \( r \) is indicated by the DOD vector as given below:-

\[ DOD_r = \sum_{gdif=1}^{g-1} g_{dif} I_{grdif} \]  

(4.55)

and the distribution of the average distance given \( g_{dif} \) is indicated by the DAD vector as given below:-

\[ DAD_j = \sum_{r=1}^{g} r I_{grdif} \]  

(4.56)

Five features that illustrate the distribution of gray level differences are determined from these characteristic vectors: Large Difference Emphasis (LDE), which computes the predominance of huge gray level differences,

\[ LDE = \sum_{j=0}^{n_g} DGD(j) \cdot \ln \left( \frac{K}{j} \right) \]  

(4.57)

where \( K \) represents a constant Sharpness (SHP), which computes the contrast and definition in an image as given in the equation,
\begin{equation}
SHP = \sum_{j=0}^{n_g} DGD(j) \cdot j^2
\end{equation}

SMG (Second Moment of DGD), which computes the difference of gray level differences

\begin{equation}
SMG = \sum_{j=0}^{n_g} DGD(j)^2
\end{equation}

SMO (Second Moment of DOD), which computes the difference of average gray level difference

\begin{equation}
SMO = \sum_{r=0}^{f_{max}} DOD(r)^2
\end{equation}

LDEL (Long Distance Emphasis for Large difference), which computes the importance of large differences a long distance from each other.

\begin{equation}
LDEL = \sum_{j=0}^{n_g} DAD(j) \cdot j^2
\end{equation}

Second Order Statistical feature extraction methods are considered to extract the relative positions of the several gray levels within the images. In Second Order Statistical (SOS), GLDM is based on the estimation of the probability density function of gray level differences in an image. The procedure to calculate the gray level differences values of five different features are mentioned in the above equation (4.53) to equation (4.61) is extracted. The experimented results are shown in Tables 4.5.
Table 4.5: Statistical Features Extracted from GLDM Method without Denoising

<table>
<thead>
<tr>
<th>Image</th>
<th>Gray Level differences</th>
<th>Sharpness</th>
<th>Second Moment of DGD</th>
<th>Second Moment of DOD</th>
<th>Long Distance Emphasis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Img1</td>
<td>-31.49336641</td>
<td>435142.8</td>
<td>5414.661</td>
<td>2985.226</td>
<td>748082.1</td>
</tr>
<tr>
<td>Img2</td>
<td>-116.0773634</td>
<td>1236061</td>
<td>676.8345</td>
<td>24068.11</td>
<td>1930651</td>
</tr>
<tr>
<td>Img3</td>
<td>-54.11084794</td>
<td>704451.9</td>
<td>3925.61</td>
<td>7405.979</td>
<td>1231665</td>
</tr>
<tr>
<td>Img4</td>
<td>-66.93226993</td>
<td>901390.9</td>
<td>3400.703</td>
<td>10992.66</td>
<td>1570312</td>
</tr>
<tr>
<td>Img5</td>
<td>-72.26345974</td>
<td>841087.9</td>
<td>1802.78</td>
<td>11280.24</td>
<td>1348998</td>
</tr>
<tr>
<td>Img6</td>
<td>-72.23907529</td>
<td>990791.3</td>
<td>2771.4</td>
<td>11975.5</td>
<td>1609727</td>
</tr>
<tr>
<td>Img7</td>
<td>-35.3108802</td>
<td>478937.3</td>
<td>4827.456</td>
<td>3398.978</td>
<td>761353.5</td>
</tr>
<tr>
<td>Img8</td>
<td>-54.96031432</td>
<td>817872.3</td>
<td>4301.761</td>
<td>8058.408</td>
<td>1384655</td>
</tr>
<tr>
<td>Img9</td>
<td>-58.6028691</td>
<td>819045.6</td>
<td>3968.399</td>
<td>9327.615</td>
<td>1467883</td>
</tr>
<tr>
<td>Img10</td>
<td>-57.95919481</td>
<td>744874.1</td>
<td>2728.781</td>
<td>7789.582</td>
<td>1149388</td>
</tr>
</tbody>
</table>

4.3 Probabilistic Neural Network (PNN) Classifier

Determining the physical characteristics of River ice types is important for public safety and navigation, as well as for the prediction and mitigation of River ice disasters. Depending on the hydro-climatic conditions, occurring at freeze up, various types of ice are formed, each possessing very different mechanical properties that require specific management strategies (Iliescu, 2007).

During recent decades, Probabilistic Neural Network (PNN) has become an effective tool for solving diversified classification problems in the field of science and engineering. PNN depends on the theory of Bayesian classification and the assessment of Probability Density Function (PDF). In effect, by substituting the sigmoid activation function frequently employed in neural networks with an exponential function, PNN that can calculate nonlinear decision boundaries which approach the Bayes optimal is generated. Hence, in this work, PNN has been used as the classifier to classify the River ice types into seven classes. The main task is to classify unknown feature vectors into predefined classes. With the selected features, PNN is then trained to serve as a classifier for discriminating seven different types of River ice classes.
PNN is closely related to Parzen window PDF estimator. PNN consists of several sub-networks, each of which is a Parzen window PDF estimator for each of the River ice type classes. The growth of the PNN relies on Parzen windows classifiers. The Parzen windows technique is a non-parametric process that synthesizes an approximate of a PDF by superposition of a quantity of windows, replicas of a function (frequently the Gaussian). The Parzen window classifier takes a classification decision after calculating the probability density function of River ice classes using the selected feature vector as training examples. The multi category classifier decision is expressed as follows:

\[ p_k f_k > p_j f_j \]  \hspace{1cm} (4.62)

where \( p_k \) is the prior probability of occurrence of River ice class features from class \( k \) and \( f_k \) is the estimated PDF of class \( k \). It is regarded as a “neural network” as a consequence of its natural mapping onto a two-layer feed forward network.

- **Architecture**

The PNN structural design includes four layers: input layer, pattern layer, summation layer and decision layer.

![Figure 4.5: PNN Classifier Architecture](image)
PNN classifier architecture is shown in Figure 4.5 which classifies River ice types from image samples. The first layer shows the input pattern with n features for River ice types selected from feature selection algorithm such as GA and DEFS. The number of nodes in the pattern layer is equal to the number of selected features from River ice image samples. The number of nodes in the summation layer is equal to the number of River ice types (seven classes) in the feature selected training samples of River ice images samples. The pattern layer is semi-connected to the summation layer. Each group of feature selected River ice image training instances corresponding to each one of the River ice types as classes which is just connected to one node in the summation class.

PNN works by creating a set of multivariate probability densities that are derived from the selected training feature vector from River ice images presented to the network. The input selected feature vector instance with unidentified category is propagated to the pattern layer. Once each node in the pattern layer receives the input, the output of the node will be computed

$$\pi_i = \frac{1}{(2\pi)^{n/2}\sigma^n} \exp \left[ -\frac{(x - x_{ij})^T(x - x_{ij})}{2\sigma^2} \right]$$ (4.63)

where \(d\) is the number of selected River ice image feature vector for River ice image samples \(x\), \(\sigma\) is the smoothing parameter, and \(x_{ij}\) is a feature vector sample of River ice images under the River ice types category \(c\) by summarizing and averaging the output of all neurons that belong to the same River ice type class

$$p_i(x) = \frac{1}{(2\pi)^{n/2}\sigma^nN_t} \sum_{i=1}^{N_t} \exp \left[ -\frac{(x - x_{ij})^T(x - x_{ij})}{2\sigma^2} \right]$$ (4.64)

where \(N_t\) denotes the total number of feature vectors samples in River ice type class \(c\). If the a priori probabilities for each River ice type class are the same, and the losses associated with making an incorrect decision for each River ice type class are the same, the decision layer unit classifies the feature vector of River ice image samples \(x\) in accordance with the Bayes’s decision rule based on the output of all the summation layer neurons.
\[ C(x) = \arg \max \{ p_i(x) \}, i = 1, \ldots, c \]  \hspace{1cm} (4.65)\]

where \( C(x) \) denotes the estimated River ice type class of the selected feature vector River ice samples \( x \) and \( m \) is total number of River ice type classes such as seven classes for River ice images. If the a priori probabilities for each class are not the same, and the losses associated with making an incorrect decision for each River ice type class are different, the output of all the summation layer neurons will be,

\[ C(x) = \arg \max \{ p_i(x) \cdot \text{cost}_i(x) \cdot \text{aprior}_i(x) \}, i = 1, \ldots, c \]  \hspace{1cm} (4.66)\]

where \( \text{cost}_i(x) \) is the cost associated with misclassifying the River ice samples feature vector and \( \text{aprior}_i(x) \) is the prior probability of occurrence of patterns in River ice type class \( c \).

From the FOS and SOS methods, 34 features are extracted which are given as input to the PNN classifier for classification. Thus, the extracted features are classified without pre-processing and without feature selection methods. The performance of the classification process without denoising and without feature selection is evaluated with accuracy as the performance metrics. It is observed that, the accuracy of the PNN classification is 71.54\%, which is clearly shown in Table 4.6.

**Table 4.6: Accuracy of PNN classifier without Pre-Processing and without Feature Selection**

<table>
<thead>
<tr>
<th>Performance evaluation metric</th>
<th>Original feature set without Denoising and without Feature selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (in %)</td>
<td>71.54</td>
</tr>
</tbody>
</table>

Thus, the PNN classifier classifies the unknown feature vectors extracted from FOS and SOS feature extraction methods into 7 predefined classes. This classification process determines the physical characteristics of a River ice types under seven categories. But, the classification accuracy without pre-processing and without feature selection approach is very poor. The main reason for this degraded performance
is the presence of irrelevant and redundant features. **Thus, there is a necessity to include the process of pre-processing and feature selection to improve the classification accuracy.** The following section clearly describes the importance of the feature selection approach to improve classification accuracy for River ice types.

### 4.4 Feature Selection Approach for River Ice Images

This section focuses on the dimensionality reduction problem to select the best features and to enhance the classification accuracy for River ice image samples. Feature subset selection method normally incorporates a search approach for exploring the space of feature subsets. In recent years, there has been a growing interest in evolutionary algorithms for diversified applications in the fields of Science and Engineering. Genetic Algorithm (GA) and Differential Evolution Feature Selection (DEFS) algorithms are relatively novel optimization techniques to solve numerical optimization problems. In this research work, the original feature set consists of 34 features, which are given as input to GA and DEFS for feature selection, to reduce the curse of dimensionality problem for River ice image samples. Two major existing feature selection algorithms considered are,

- Genetic Algorithm (GA), and
- Differential Evolution Feature Selection (DEFS).

The feature subset selected from the existing feature selection methods are illustrated in Figure 4.6.
Figure 4.6: Proposed Feature Selection Methods
4.4.1 Genetic Algorithm

- **Basics of Genetic algorithm**

  Genetic Algorithms are one of the best ways to solve numerical optimization problems. GA is one among the optimization algorithm and so will perform well in any search space. Genetic Algorithm will be capable of creating a high quality solution. Genetic Algorithms employ the principles of selection, crossover and mutation to construct numerous solutions to a specified problem.

- **Genetic Algorithm operations**

  In Genetic Algorithm, initially, population is created randomly with a group of individuals. The individuals in the population are then evaluated. The evaluation function is provided by the developer and individual scores are obtained based on the performance. Then, individuals are chosen depending on their higher fitness. These individuals subsequently “reproduce” to generate one or more offspring, after which the offspring are mutated arbitrarily. This persists until an appropriate solution has been found or a certain number of generations have passed.

**Selection**

  In this work, the most common type - roulette wheel selection is used for the selection process. In roulette wheel selection, individuals are given a probability of being selected that is directly proportionate to their fitness. Two individuals are subsequently chosen arbitrarily depending on these probabilities and produce offspring.

**Crossover**

  Crossover is a genetic operator employed to differ the programming of a chromosome or chromosomes from one generation to the next. It is equivalent to reproduction and biological crossover, upon which Genetic Algorithms are dependent. Cross over is a process of taking more than one parent solutions and producing a child solution from them. The example of the crossover operation between two chromosomes is shown below:
Optimized Feature Selection Methods for Classification of River Ice Types from Aerial Images

Crossover

Parent 1

| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

Child 1

| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

Parent 2

| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Child 2

| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

Mutation

Mutation preserves genetic diversity from one generation of a population of Genetic Algorithm chromosomes to the next. It is analogous to biological mutation. Mutation changes one or more gene values in a chromosome from its preliminary condition.

| Before | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| After  | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

Based on the above mentioned operations, Genetic Algorithm produces good optimal feature subset results, and redundant features are removed from images. So, Genetic Algorithm has been considered for feature selection from River ice image samples in this research work.
4.4.2 Genetic Algorithm for Feature Selection

The goal of the Genetic Algorithm (GA) is to find the optimal input variables. In GA, the chromosomes are composed of all the input feature extracted variables. The input vector is spread as an initial individual in the initial population with the intention of accelerating the convergence of the algorithm. Next, the remaining individuals of the initial population are randomly generated with uniform distribution and standard deviation ($\delta$) around the seeded individual. GA applied in this work operates on the decimal basis. In order to select individual in the input vector population, the calculation of the fitness function becomes major important. In this approach, the Joint Conditional Entropy (JCE) is directly used as the fitness function of the GA and the objective of the optimization problem is to obtain the input feature extracted vectors $X_1, \ldots, X_n$.

Discrete Joint Conditional Entropy (JCE)

JCE, which enables the evaluation of the amount of information, lacks to determine the target output variable (Cover, 1999). This work considers discrete variables. Hence, every continuous variable is discretized by sub-dividing its selection into a finite set of interval. A mean value for each interval is considered respectively. The conditional entropy of the scalar discrete random output variable $Y$, assuming the event $X = x_k$, is given by

$$H(Y| x_k) = \sum_{i=1}^{N} -log(P(y_i|x_k) \cdot P(y_i|x_k))$$

(4.67)

where $P(y_i|x_k)$ is the probability that $Y = y_i$ assuming event $X = x_k$. The conditional entropy of $Y$, considering the knowledge of $X$, is given by

$$H(Y| X) = \sum_{k=1}^{N} H(Y| x_k) \cdot P(x_k)$$

(4.68)

which describes the ambiguity regarding $Y$ when all the trials of $X$ are recognized with the following equation below:
\[ H(Y|X) = H(X,Y) - H(X) \]  \hspace{1cm} (4.69)

where \( H(X,Y) \) is the joint entropy:

\[ H(X,Y) = \sum_{i=1}^{N} \sum_{k=1}^{N} -\log(p(x_i,y_k)p(x_i,y_k)) \]  \hspace{1cm} (4.70)

The concept of joint entropy Equation (4.67) can be extended to the following general formulation Equation (4.68), where \( n \) input variables \( X_1, \ldots, X_n \), and \( N \) discrete possible values (bins) for each variable are assumed:

\[ H(Y,X_1,\ldots,X_n) = \sum_{i=1}^{N} \sum_{i_1=1}^{N_1} \sum_{i_n=1}^{N_n} -\log \left( P(y_{i_1,\ldots,i_n} \mid x_{i_1,\ldots,i_n}) \cdot P(x_{i_1,\ldots,i_n}) \right) \]  \hspace{1cm} (4.71)

where \( x_{i_1,i_j} \) is event \( i_j \) of variable \( X_j \), for \( i_j = 1, \ldots, N_j \), and \( j = 1, \ldots, n \).

This work defines

\[ H(Y|X_1,\ldots,X_n) = H(X_1,\ldots,X_n,Y) - H(X_1,\ldots,X_n) \]  \hspace{1cm} (4.72)

as the JCE, which defines the amount of expected information that lacks to determine a target variable \( Y \) with the input variables \( X_1,\ldots,X_n \).

- **Cross Entropy Function**

The cross correlation function is based on a linear adjustment among the variables. For this reason, the method encounters some problems when applied to non-linear systems. The problem of cross correlation function is overcome by Cross Entropy Function (XEF), which is a suitable analysis that is more appropriate for non-linear dynamic relationships. Normalized mutual information, \( R \), between two signals, \( X \) and \( Y \) is given below:

\[ R = \frac{H(Y) - H(Y|X)}{H(Y)} \]  \hspace{1cm} (4.73)
Parameter $R$ is limited to $[0, 1]$ interval and indicates the amount of information regarding the target variable that is revealed by the input variable. When $X$ contains all the necessary information to predict $Y$, then $H(Y|X) = 0$ and $R = 1$. When $X$ does not contain information regarding $Y$, then $H(Y|X) = H(Y)$ and $R = 0$.

In this approach, the JCE Equation (4.67) is directly used as the fitness function of the GA and the objective of the optimization problem is to obtain the input signals $X_1, \ldots, X_n$ that achieve the minimum JCE, with the equation below:

$$
\min_{x_1,\ldots,x_N} H(Y|(X_1, \ldots, X_n))
$$

(4.74)

Following fitness evaluation, the algorithm organizes the individuals by their fitness ranking the indexes of the individual vectors from most excellent to most horrible on their performance. After that, the crossover operator is executed with the purpose of creating a new generation. The chosen crossover genetic operator was the $BLX-\alpha$, defined by

$$
g^n_c(k + 1) = \text{round}(\alpha \cdot g^n_1(k) + (1 - \alpha) \cdot g^n_2(k))
$$

(4.75)

where $k$ is the generation; $\alpha \in [0,1]$ with uniform distribution; $g^n_c$ is the gene $n$ of the child chromosome $c$, and $g^n_1$ and $g^n_2$ are genes corresponding to position $n$ of parent chromosome vectors.

Each individual has a probability of reproduction that is given by its fitness value. More adapted individuals (i.e. the individuals in superior positions of the ranking) have more prospect of participating in the crossover operation. The selection process depending on a consistent random variable is not sufficient; consequently, random variable is employed to choose individuals to participate in the crossover. The random variable employed to choose individuals to take part in the crossover must have a probability density that is better for small values than for large values with the intention of favouring the more adapted individuals. To carry out this, random variable $\sigma'$ is used:

$$
\sigma' = \frac{e^{a \sigma} - 1}{e^a - 1}
$$

(4.76)
where \( a \) is a positive constant, and \( \sigma \in [0,1] \) is a random variable with uniform distribution. This procedure makes it possible to increase or decrease the selective search. Search space is increased for faster convergence. The algorithm may constrain the population to a local minimum for larger search space. According to discrete random variable individuals are selected for crossover by the rank \( i \),

\[
i = \text{round}(T \sigma')
\]

where \( T \), total number of individuals

From the above mentioned Genetic Algorithm based feature selection method, the selected feature subsets are determined without irrelevant features from the River ice images samples are removed. Table 4.7 shows the experimentation results of Genetic Algorithm (GA) based feature selection method without denoising from the original feature set. The desired numbers of features selected are 5, 7, 9, 10, 12, 15, 17 and 20. For demonstration, only 5, 10 and 15 features are shown. From the table, it is observed that, 5, 10 and 15 features are selected by the GA based feature selection process.

**TABLE 4.7: Features Selected by GA Method from Original Feature Set without Denoising**

<table>
<thead>
<tr>
<th>Best features</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Features</td>
<td>Mean, Correlation, Sum of Squares, LRE, LDEL</td>
</tr>
<tr>
<td>10 Features</td>
<td>Mean, Correlation, Sum of Squares, Sum Variance, Difference entropy, Information measures of correlation (1), Information measures of correlation (2), LRE, HGRE, LDEL</td>
</tr>
<tr>
<td>15 Features</td>
<td>Mean, Standard Deviation, Contrast, Correlation, Energy, Sum of Squares, Sum Variance, Information measures of correlation (1), Information measures of correlation (2), Inverse difference moment normalized, SRE, LRE, HGRE, SHP, LDEL</td>
</tr>
</tbody>
</table>
Thus, the selected features are given as input to the PNN classifier for classification. The performance of the classifier, after the feature selection process, is evaluated. Table 4.8 shows the classification accuracy of the PNN classifier with GA based feature selection approach for different features set like 5, 10 and 15. It is observed that, classification accuracy after GA based feature selection shows slight improvement. But the performance improvement is not significant.

**Table 4.8: Performance Evaluation Metrics of GA based Feature Subset Selection through PNN Classifier**

<table>
<thead>
<tr>
<th>Performance evaluation metrics</th>
<th>Feature subset selected from GA method without Denoising</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>5</td>
</tr>
<tr>
<td>Accuracy (in %)</td>
<td>66.39</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.71</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.64</td>
</tr>
</tbody>
</table>

As a result, Genetic Algorithms are not constantly the best choice because in certain circumstances, the convergence time taken by the GA is observed to be higher.

- Certain optimisation problems cannot be solved by means of Genetic Algorithm. This takes place because of poorly known fitness functions, which produce bad chromosome blocks despite the fact that only good chromosome blocks crossover, and
- Furthermore, there is no absolute guarantee that a Genetic Algorithm will discover a global optimum. It takes place very frequently when the populations have a lot of subjects.

In order to overcome these problems and to improve the feature selection accuracy for River ice images, Differential Evolution Feature Selection (DEFS) method is used for feature selection.
4.4.3 Differential Evolution Feature Selection (DEFS)

DEFS is an uncomplicated optimization technique that has parallel, direct search, straightforward, better convergence and quick implementation properties. The initial step in the DEFS optimization technique is to create a population of NP members each of D-dimensional real-valued parameters, in which NP is the population size, and $D$ indicates the number of parameters to be optimized. The fundamental idea behind DE is a new proposal for generating trial parameter vectors by appending the weighted difference vector among two population members $x_{r1}$ and $x_{r2}$ to a third member $x_{r0}$. The following equation shows how to merge three different randomly selected vectors to create a mutant vector, $v_{i,g}$ from the current generation $g$:

$$V_{ijg} = x_{j,y_{0}g} + F \times (x_{j,y_{1}g} - x_{j,y_{2}g})$$  \hspace{1cm} (4.78)

where $F \in (0,1)$ is a scale factor that control the rate at which the population evolve. The index $g$ indicates the generation to which a vector belongs. In adding up, each vector is assigned a population index, $i$, which runs from 0 to $NP - 1$. Parameters in vectors are indexed with $j$, which operates from 0 to $D - 1$. In addition, DE employs uniform crossover, also known as discrete recombination, in order to build testing vectors out of parameter values that have been copied from two different vectors. DE crosses each vector with a mutant vector, as specified in Equation (4.79):

$$u_{ijg} = \begin{cases} v_{j,i,g} \text{ if } rand(0,1) \leq C_r \\ x_{j,i,g} \text{ otherwise} \end{cases}$$  \hspace{1cm} (4.79)

where $v_{j,i,g}$ is the $j^{th}$ dimension from the $i^{th}$ trial vector along the current population $g$. The crossover probability $C_r \in [0,1]$, is a user specified value that manages the fraction of constraint values that are copied from the mutant (Ahmed Al-Ani et al., (2013)).

Like all population-based optimizers, DEFS attacks the starting point problem by sampling the objective function at multiple, randomly chosen initial points as original population. Thus an original population matrix of size $(NP \times DNF)$
containing NP randomly chosen initial vectors, \( x_i, i = \{1, \ldots, NP - 1\} \) is created, where DNF is the desired number of features to be selected. In the DEFS method, the search space is limited between 1 and the total number of features (NF). The subsequent step in the method is to produce a set of new vectors from the original population, as mutant population, in which each vector is indexed with a number from 0 to \( NP - 1 \) (Khushaba et al., 2011).

For each position in the original population matrix, a mutant vector is formed by adding the scaled difference between two randomly selected population members to a third vector, according to Equation (4.78). Unlike the original DE that uses a constant scale factor, the DEFS allows the scale factor to change dynamically as follows:

\[
F = \frac{C_1 \times \text{rand}}{\max \{x_{j_1,r_1,g}, x_{j_2,r_2,g}\}}
\]

(4.80)

where \( C_1 \) is a constant smaller than 1. The effect of this is to allow the population members to oscillate within bounds without crossing the optimal solutions and thereby aiding them to find improved points in the optimal region. Additionally, a system constant with stipulation is implemented as

\[
x_{j,i,g} = \begin{cases} 
NF & \text{if } x_{j,i,g} > NF \\
1 & \text{if } x_{j,i,g} < 1
\end{cases}
\]

(4.81)

In the selection stage, the trial vector competes against the population vector of the same index, \( x_0 \). The corresponding position in the population matrix will contain either the trial vector, \( u_0 \) (or its corrected version), or the original vector, \( x_0 \), depending on which one of them achieved a better fitness (i.e., lower classification error rate in this case). The procedure continues until each of the \( NP \) population vectors have competed against an arbitrarily generated trial vector. Once the last experiment vector has been tested, the survivors of the \( NP \) pair-wise competition become parents for the next production in the evolutionary cycle (Khushaba et al., 2011).

Due to the fact that a real number optimizer is being used, it facilitates the two dimension values from settling at the same feature indexes. As an example, if the
resultant vector is [3.7353, 20.1000, 13.0000, 4.0000, 13.1471, 10.8478, 20.0000, 21.9286, 15.0000, 8.0789], after rounding the values, the resultant vector values will be [4, 20, 13, 4, 13, 11, 20, 22, 15, 9].

The feature index 4 occurs twice which is completely unacceptable in feature selection method. In order to overcome such a problem, a roulette wheel weighting scheme is utilized (Haupt R. L, 2004). In this scheme, a cost weighting method is implemented in which the probabilities of each feature are calculated from the distribution factors associated with it. The distribution factor of feature $f_i$ within the current generation $g$, is referred as $FD_{j,g}$ which is calculated using below equation:

$$FD_{j,g} = a_1 \times \left( \frac{PD_j}{PD_j + ND_j} \right) + \frac{NF - DNF}{NF} \times \left( 1 - \frac{(PD_j + ND_j)}{\max(PD_j + ND_j)} \right)$$

(4.82)

where $PD_j$ the number of times that feature is $f_j$ has been used in the good subsets. $ND_j$ is the number of times that feature $f_j$ has been used in the less competitive subsets. $NF$, is the total number of features, $a_1$ is a suitably chosen positive constant that reflects the importance of features in $PD$, and $DNF$ is the desired number of features to be selected. Divide the estimated distribution factors for the current and the next iterations by the maximum value, i.e,

$$FD_g = \frac{FD_g}{\max(FD_g)}$$

(4.83)

and

$$FD_{g+1} = \frac{FD_{g+1}}{\max(FD_{g+1})}$$

(4.84)

Compute the relative difference according to the following equation:

$$T = \left( FD_{g+1} - FD_g \right) \times FD_{g+1} + FD_g$$

(4.85)

The above equation provides higher weights to features that make obvious improvement in the current iteration in comparison to the previous one. Then, some sort of randomness is added in this process to avoid selecting the same features every time, to emphasize the importance of unseen features.
\[ T = T - 0.5 \times \text{rand}(1,NF) \times (1 - T) \] (4.86)

For the remaining iterations, the distribution factors are updated in each and every iterations as \( FD_g = FD_{g+1} \), \( FD_{g+1} \) holds recently computed values in each iteration.

From the above discussed, DEFS feature selection method, the best features are selected whereas the irrelevant features from the River image samples are removed. Table 4.9 shows the features selected by DEFS method without denoising. It is clearly observed from the table that, the features selected by the GA method are different from the DEFS method. *This shows the significance of the feature selection approach.*

**Table 4.9: Features Selected by DEFS Method from Original Feature Set without Denoising**

<table>
<thead>
<tr>
<th>Features</th>
<th>DEFS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5 Features</strong></td>
<td>Skewness, Kurtosis, Sum of Squares, Information measures of correlation (1), HGRE</td>
</tr>
<tr>
<td><strong>10 Features</strong></td>
<td>Contrast, Correlation, Sum Average, Inverse difference moment normalized, SRE, LRE, RP, RLN, HGRE, SHP</td>
</tr>
<tr>
<td><strong>15 Features</strong></td>
<td>Skewness, Energy, Homogeneity, Sum of Squares, Sum Average, Sum Entropy, Difference entropy, Information measures of correlation (1), Inverse difference, Inverse difference normalized, Inverse difference moment normalized, SRE, LRE, RP, SHP</td>
</tr>
</tbody>
</table>

Then, the selected features are given as input to the PNN for classification. The performance of the DEFS approach is evaluated based on the classification Accuracy, Sensitivity and Specificity which are shown in Table 4.10. *It is observed that, classification accuracy after DEFS shows significant improvement.* The observed accuracy is higher than the GA based feature selection approach.
<table>
<thead>
<tr>
<th>Performance evaluation metrics</th>
<th>Feature subset selected from DEFS method without Denoising</th>
</tr>
</thead>
<tbody>
<tr>
<td>Features</td>
<td>5</td>
</tr>
<tr>
<td>Accuracy (in %)</td>
<td>73.73</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>0.77</td>
</tr>
<tr>
<td>Specificity</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Thus, the PNN classifier classifies the unknown feature vectors from GA and DEFS into 7 predefined classes. This classification process determines the physical characteristics of a River ice sample types under seven categories.

4.5. Summary

This chapter focuses on developing an efficient feature selection method to solve the curse of dimensionality problem and to select best features from the original dataset, which improves the result of classification accuracy from River ice types. Since the features extracted from River ice images consist of irrelevant and noise features, it reduces the classification accuracy in pattern recognition. So, efficient feature selection method is introduced in this research work to improve the classification performance of Probabilistic Neural Network (PNN). To improve feature selection results, initially, image segmentation is done by using Gabor filtering and Fuzzy C Means (FCM) segmentation, and then feature extraction is done by using First Order and Second Order Statistical features by GLCM, GLRM and GLDM. Population based feature selection methods such as GA and DEFS are introduced. The results obtained by these methods indicate the potential advantages of using feature selection techniques to improve the classification accuracy with less number of feature subset. From the result, one can conclude that the performance of DEFS is superior to GA method for PNN classification. The design of the feature selection after denoising is presented in the next chapter, Modified Feature Selection after Denoising (Chapter 5).