CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

In this chapter, the methods of image acquisition, pre-processing and analyses of structural properties of femur trabecular bone that include structure tensor, lacunarity, succolarity, fractal and wavelet analysis have been presented. Also, correlation of structural parameters with macrostatic parameter is discussed. Feature reduction using PCA and classification using ELM are discussed in detail.

3.2 PROCESS PIPE-LINE

A unique flow of process pipe-line is shown in Figure 3.1. The work includes pre-processing, spectral and spatial analyses, correlation, feature selection and classification procedures carried out for femur radiographic images.

3.3 RADIOGRAPHIC IMAGE ACQUISITION

The femur bone images (N=40) are recorded using a Siemens 500mA Polyscope clinical X-ray unit. The radiographic exposures were made with 160 mAs at 60 kVp and exposure time was 0.05s. These parameters remain constant for all the captured images. The standard anteroposterior view is used to image all specimens as more vertically oriented compressive

trabeculae and arc shaped tensile of the proximal femur are well demonstrated in this view

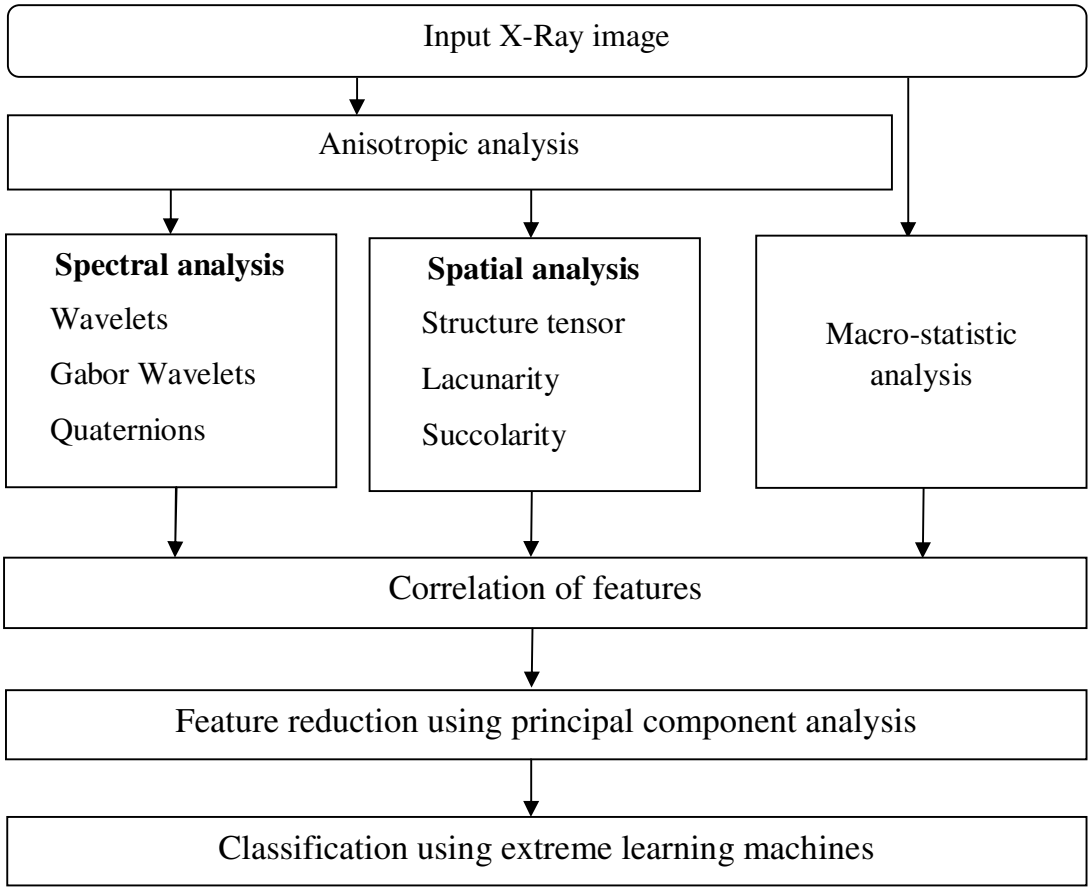

Figure 3.1 Flow chart of the process pipe-line

3.4 RADIOGRAPHIC IMAGE PRE-PROCESSING

The contrast of the acquired radiographic images is enhanced using Adaptive Histogram Equalization (AHE) by considering the neighbourhood pixels to avoid information loss (Haller 2011). This method involves mapping a sample of pixels to histogram equalization and interpolating the mapping between these sample locations. It is therefore suitable for improving the local contrast of an image and in bringing out more detail. The interpolated AHE is given by

$$m(i) = a\left[bm_{--}(i) + (1-b)m_{+-}(i)\right] + \left[1-a\right]\left[bm_{-+}(i) + (1-b)m_{++}(i)\right] \qquad (3.1)$$

where $a = (y - y_-)/(y_+ - y_-)$ and $b = (x - x_-)/(x_+ - x_-)$ and i is the intensity, x_+ and x_- represent the position of pixels to the right and left of current position x, y_+ and y_- represent the position of pixels to the top and bottom of current position y, intensity m_{+-} is the mapping at the grid pixel (x_+, y) to the upper right of (x, y) and similarly the subscripts for the mappings + +, - + , + - and $--$ represent the locations of the grid pixels to the lower right, lower left, upper left and upper right respectively. Pixels in the borders of the image outside the sample pixels are mapped using linear interpolation at the two closest points or, in the corner where there is only a single close sample pixel.

3.5 DELINEATION OF REGION OF INTEREST

The trabecular bone orientation and density in bones such as the proximal femur is heterogeneous, and thus the projected pattern shows considerable variations in different regions of the radiographic projection. In the conventional planar radiographic images, two strength regions namely, compressive and tensile regions of trabecular patterns are visible. These regions are identified manually by the position in terms of a coordinate system whose axes comprise the shortest line across the femoral neck and a line through the center of the femoral head (Singh et al 1970).

The compressive strength region of size 140 x 80 and tensile strength region of size 250 x 80 are delineated using this method. These regions are heterogeneous in nature, having varying bone strength and change most markedly in appearance with bone loss (Singh et al 1970). The qualitative analysis is also performed on the delineated images to derive apparent porosity which is ratio of void area to bone area.

3.6 WAVELET TRANSFORM

The delineated strength regions are subjected to wavelet analysis. The wavelet analysis decomposes images into four subbands namely A^1, H^1, V^1 and D^1. The subband A^1 corresponds to coarse level or approximation coefficients and the subbands labelled H^1, V^1 and D^1 represent the finest level or detail coefficients. These coefficients are computed by convolving signals with the low-pass filter for approximation, and with the high-pass filter for detail. The convolved coefficients are down-sampled by keeping the even indexed elements. An image f(x, y) with a spatial resolution of 2^j decomposed at first level is represented as

$$f(x,y) = A^1\varphi_{2^j}(X - 2^{-j}n, Y - 2^{-j}m) + H^1\psi^1_{2^j}(X - 2^{-j}n, Y - 2^{-j}m)$$

$$+ V^1\psi^2_{2^j}(X - 2^{-j}n, Y - 2^{-j}m) + D^1\psi^3_{2^j}(X - 2^{-j}n, Y - 2^{-j}m) \qquad (3.2)$$

where integer j is a decomposition level, m, n are integers, $\phi(x)$ is a one-dimensional scaling function which provides low frequency information, and $\psi(x)$ is a one-dimensional wavelet function which provides high frequency information. The wavelet decomposition can thus be interpreted as image decomposition in a set of independent, spatially oriented frequency channels (Mallat 1989), so that $\phi(x)$ and $\psi(x)$ can be defined as:

Dilation equation $$\phi(t) = \sqrt{2}\sum_k c(k)\phi(2t - k) \qquad (3.3)$$

Wavelet equation $$\psi(t) = \sqrt{2}\sum_k d(k)\phi(2t - k) \qquad (3.4)$$

Haar wavelet has coefficients: $c(0)=c(1)=1/\sqrt{2}$, $d(0)= 1/\sqrt{2}$, and $d(1)= - 1/\sqrt{2}$. Thus its dilation equation and wavelet equation can be expressed as:

$$\phi(t) = \phi(2t) + \phi(2t-1) \tag{3.5}$$

$$\psi(t) = \phi(2t) - \phi(2t-1) \tag{3.6}$$

To obtain the next level of decomposition, the approximation coefficients A^1 is further decomposed into four subbands. The decomposition process can be recursively applied to the low-frequency channel A^1 to generate image details A^2 (low-frequency channel) and H^2, V^2 and D^2 (high-frequency channels), at the next level and so on. The approximation and detail coefficients can be computed with a pyramid algorithm based on convolutions with two one-dimensional parameter filters (Myint et al 2002). The algorithm for the application of the filters and down sampling procedure for computing the approximation and detail coefficients is illustrated in Figure 3.2.

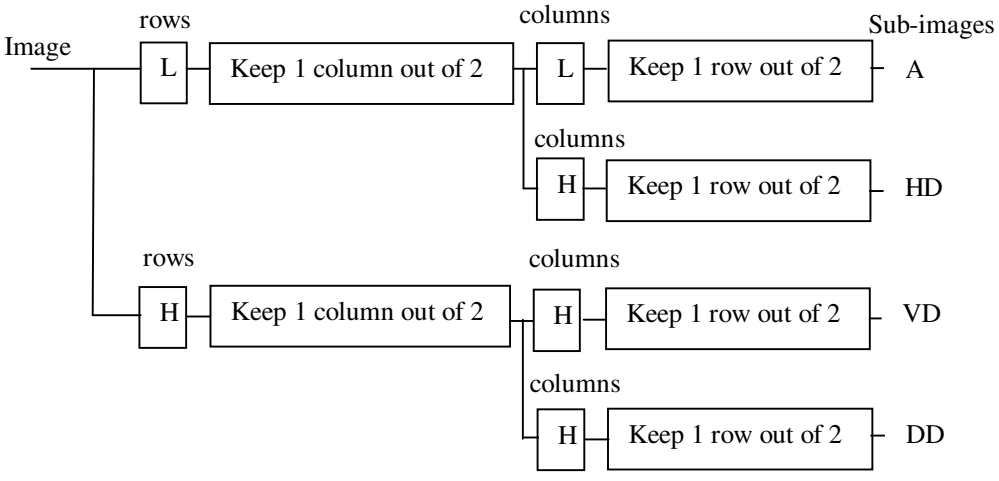

Figure 3.2 Decomposition procedure of an image by the multiresolution analysis

Initially, the rows of an image are convolved with a one-dimensional Low Pass Filter (LPF) L. Next, the filtered images are down sampled. In the first step, down sampling is performed by keeping one column out of two. Then the resulting images are convolved with another one-dimensional low pass filter, retaining every other row. To obtain a horizontal detail image, first the rows of the input image are convolved with a lowpass filter and the filtered image are down sampled by keeping one column out of two. In the next stage, the columns of the image are convolved with a High Pass Filter (HPF) and again every other row is retained. For the vertical details, the original images are convolved first with a highpass filter H and then with a lowpass filter L, following the above procedure. For the diagonal detail image, the same down sampling procedure is carried out, using two highpass filters consecutively (Myint et al 2002). The second level approximation and details coefficients of a two-dimensional image f(x, y) at resolution 2^j are represented using the following equations:

$$A_{2^j}^{d} f = ((f(x,y)^{*} \phi_{2^j}(-x)\phi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in Z^2} \tag{3.7}$$

$$D_{2^j}^{1} f = ((f(x,y)^{*} \phi_{2^j}(-x)\psi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in Z^2} \tag{3.8}$$

$$D_{2^j}^{2} f = ((f(x,y)^{*} \psi_{2^j}(-x)\phi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in Z^2} \tag{3.9}$$

$$D_{2^j}^{3} f = ((f(x,y)^{*} \psi_{2^j}(-x)\psi_{2^j}(-y))(2^{-j}n, 2^{-j}m))_{(n,m)\in Z^2} \tag{3.10}$$

The normal and abnormal images are subjected to Haar, Daubechies5 and Coiflet5 and three levels of decomposition are performed to derive approximation and detail (horizontal, vertical and diagonal) coefficients. The statistical values of wavelet coefficients such as mean, standard deviation, energy, skewness and kurtosis are estimated for the

individual strength regions. The mean takes the average level of intensity of the image or texture being examined and is given by

$$Mean: \quad \mu = \sum_{i=0}^{G-1} ip(i) \tag{3.11}$$

where *p(i)* is the probability density of occurrence of the gray levels of the image. The mean is expected to be large if the gray levels G of the image are high. Energy feature measures the intensity level distribution of image. If energy level is high then the distribution is to a small number of intensity levels. Energy can be calculated

$$Energy: \quad E = \sum_{i=0}^{G-1} [p(i)]^2 \tag{3.12}$$

The skewness is a measure of the inequality of intensity levels distribution about mean. The value may be positive or negative of the skewness. The negative value represents that the tail of the block pixel values on the left side of the probability distribution is skewed or longer than the right side and the bulk of the values lie to the right of the mean. A positive value represents that the tail on the right side is longer than the left side and the bulk of the values lie to the left of the mean. A zero value of skewness means that the gray level values on both sides of the mean are relatively equally distributed. The skewness can be calculated as

$$Skewness: \quad \mu_3 = \sigma^{-3} \sum_{i=0}^{G-1} (i-\mu)^3 p(i) \tag{3.13}$$

where σ is the standard deviation of the gray levels of an image. The fourth moment kurtosis is a measure of the peakedness of the probability distribution of gray levels. For high value of kurtosis the distribution has a

sharper peak and longer, fatter tails, while for low value distribution gives more rounded peak and shorter thinner tails. Kurtosis can be defined as

$$Kurtosis: \quad \mu_4 = \sigma^{-4} \sum_{i=0}^{G-1} (i-\mu)^4 p(i) \tag{3.14}$$

3.7 GABOR WAVELET TRANSFORM

The input image is convolved with a bank of Gabor filters to produce a series of output images containing Gabor coefficients. The impulse response of an even symmetric Gabor filter is given as

$$h(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\} \cos(2\pi u_o x) \tag{3.15}$$

where u_0 is the frequency of a sinusoidal plane wave along x-axis, and σ_x and σ_y are the space constants of the Gaussian envelope along the x and y axes, respectively (Xiang et al 2007, Liu et al 2012). Both σ_x and σ_y take on values related to the center radial frequency u_0. Filters with arbitrary orientations (θ) can be obtained through a rigid rotation of x-y coordinate system.

$$x' = x\cos\theta + y\sin\theta \tag{3.16}$$

$$y' = -x\sin\theta + y\cos\theta \tag{3.17}$$

Thus Gabor wavelets or filters are generated which cover the two dimensional image plane. The Fourier transform representation of each Gabor filter is given by

$$H(u,v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} + \exp\left\{-\frac{1}{2}\left[\frac{(u+u_0)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \tag{3.18}$$

where,

$$\sigma_x = \frac{1}{2\pi\sigma_u}$$
$$\sigma_y = \frac{1}{2\pi\sigma_v}$$

The Gabor filter is composed of filters with center frequencies, u_0 and orientations θ_n, where m ranges over the sequence of filter centre frequencies (1~M) and n ranges over the number of orientations (1~N) (Xiang et al 2007). The design of Gabor filter involves determining the filter parameters u_0, σ_x and σ_y. From the geometric restraint, σ_x and σ_y are expressed as,

$$\sigma_x = \frac{3\sqrt{2\ln 2}}{2\pi u_0} \tag{3.19}$$

$$\sigma_y = \frac{1}{2\pi \tan\left(\frac{\theta}{2}\right)\sqrt{\frac{u_0^2}{2\ln 2} - \frac{1}{4\pi^2\sigma_x^2}}} \tag{3.20}$$

The center frequencies u_0's are chosen such that the frequency bands associated with them represent the actual frequency contents of a particular texture. The maximum value of the centre frequency u_{0max} is given by

$$u_{0\max} = \frac{3N_w}{8}(cycles / N_w) \tag{3.21}$$

where N_w denotes the image width in pixels. The highest possible frequency is given by

$$u_{0\max} + \Delta u = \frac{N_w}{2}(cycles / N_w) = \frac{1}{2}(cycles / pixel) \tag{3.22}$$

where Δu, the half-peak span of a Gaussian function, is given by

$$\Delta u = \sqrt{2\ln 2\sigma_u} \tag{3.23}$$

The successive lower frequency bands are determined by setting them one octave apart. Therefore, the sequence of center frequencies u_0's from the highest to the lowest frequency, assuming N_w as power of 2, can be given as:

$$\frac{3N_w}{8}, \frac{3N_w}{16}, ..., \frac{3}{1}, \frac{3}{2}(cycles / N_w) \tag{3.24}$$

The last few frequency bands represent frequencies that are low. N_w is chosen to be 240, five frequency bands 120, 60, 30, 15 and 6 cycles/N_w (Xiang et al 2007). The Gabor coefficients $g_{mn}(x, y)$ is obtained by convolving input image with the Gabor filters. Features such as texture energy, skewness and kurtosis are derived from these coefficients and are defined as

$$e_{mn}(x, y) = \frac{1}{W^2} \sum_{(x_1, y_1) \in W_{xy}} G(x, y, \sigma_x, \sigma_y) \left| g_{mn}(x_1, y_1) \right| \tag{3.25}$$

$$S_{mn}(x, y) = \frac{1}{W - 1} \sum_{(x_1, y_1) \in W_{xy}} \left(G(x, y, \sigma_x, \sigma_y) \left(\frac{\left(g_{mn}(x_1, y_1) - \mu_{mn}(x, y) \right)^3}{\sigma_{mn}^3(x, y)} \right) \right) \tag{3.26}$$

$$K_{mn}(x, y) = \frac{1}{W - 1} \sum_{(x_1, y_1) \in W_{xy}} \left(G(x, y, \sigma_x, \sigma_y) \left(\frac{\left(g_{mn}(x_1, y_1) - \mu_{mn}(x, y) \right)^4}{\sigma_{mn}^4(x, y)} \right) \right) \tag{3.27}$$

where σ_x , σ_y are the standard deviations, W_{xy} is a W×W window and G(x, y, σ_x, σ_y) is a Gaussian window centered at the pixel with coordinates (x, y). Therefore, if M total frequency bands and N total orientations are chosen, the

M×N features associated with each pixel will be as [e_{11} e_{22}...e_{1n}, e_{21} e_{22}...e_{2n}, …, e_{m1} e_{m2}...e_{mn}] (Xiang et al 2003). The anisotropy is a measure of the difference between the maximum and minimum energy and is represented as, (Xiang et al 2007).

$$M_{anisotropy} = \max\left(\left|\frac{\sum_{m=1}^{M} e_{mn}}{M}\right|_n\right) - \min\left(\left|\frac{\sum_{m=1}^{M} e_{mn}}{M}\right|_n\right) \tag{3.28}$$

Similarly the texture kurtosis and skewness are derived for different orientations and corresponding anisotropy values are calculated.

3.8 QUATERNION WAVELET TRANSFORM

The quaternion wavelet transform provides a quaternionic multiresolution analysis with 2D analytic wavelets. It consists of one magnitude and three phase components (Yin et al 2012). The first two QWT phases describe the shifts of the image features in the vertical and horizontal directions, and the third QWT phase can describe the texture information of the image. The analytic wavelets of 2D signal are constructed by setting its Hilbert transform in the imaginary part (Gai et al 2013). The analytic extension of mother wavelets ψ^D , ψ^V , ψ^H and scaling function φ are given as

$$\psi^D(x,y) = \psi_h(x)\psi_h(y) \rightarrow \psi^D + iH_{i1}\psi^D + jH_{i2}\psi^D + kH_i\psi^D \tag{3.29}$$

$$\psi^V(x,y) = \phi_h(x)\psi_h(y) \rightarrow \psi^v + iH_{i1}\psi^v + jH_{i2}\psi^v + kH_i\psi^v \tag{3.30}$$

$$\psi^H(x,y) = \psi_h(x)\phi_h(y) \rightarrow \psi^H + iH_{i1}\psi^H + jH_{i2}\psi^H + kH_i\psi^H \tag{3.31}$$

$$\phi(x,y) = \phi_h(x)\phi_h(y) \rightarrow \phi + iH_{i1}\phi + jH_{i2}\phi + kH_i\phi \tag{3.32}$$

where H_{i1}, H_{i2} are partial Hilbert transform along x and y axis respectively, H_i is combination of two partial HTs along both axes. 2D HT of separable functions is equivalent to 1D HT along rows and columns. The 1D Hilbert pair of wavelets and scaling functions are given by $(\psi_h, \psi_g = H\psi_h)(\phi_h, \phi_g = H\phi_h)$ The quaternion wavelets are represented in terms of separable products and are given by

$$\psi^D(x,y) = \psi_h(x)\psi_h(y) + i\psi_g(x)\psi_h(y) + j\psi_h(x)\psi_g(y) + k\psi_g(x)\psi_g(y) \quad (3.33)$$

$$\psi^v(x,y) = \phi_h(x)\psi_h(y) + i\phi_g(x)\psi_h(y) + j\phi_h(x)\psi_g(y) + k\phi_g(x)\psi_g(y) \quad (3.34)$$

$$\psi^H(x,y) = \psi_h(x)\phi_h(y) + i\psi_g(x)\phi_h(y) + j\psi_h(x)\phi_g(y) + k\psi_g(x)\phi_g(y) \quad (3.35)$$

$$\phi(x,y) = \phi_h(x)\phi_h(y) + i\phi_g(x)\phi_h(y) + j\phi_h(x)\phi_g(y) + k\phi_g(x)\phi_g(y) \quad (3.36)$$

Each quaternion wavelet basis contains four components in the vertical, horizontal and both directions, which are of 90 degree phase shift with each other.

3.9 QUATERNION HILBERT ANALYSIS

QHT is based on improved version of Hilbert transform. The combination of a 1D-signal and its Hilbert transform is called the analytic signal. The Hilbert transform f_{Hi} of a real 1D-signal f is given by:

$$f_{Hi}(x) = f(x) * \frac{1}{\pi x} \quad (3.37)$$

where * denotes convolution. The complex signal of f is given by

$$f_A(x) = f(x) + if_{Hi}(x) \quad (3.38)$$

The real part of f_A is identical to the input signal, while the imaginary part is a phase-shifted version of f. In the frequency domain, the Hilbert transform is defined by

$$F_{Hi}(u) = -\frac{iu}{|u|}F(u) \tag{3.39}$$

where F and F_{Hi} are the Fourier transforms of f and f_{Hi}, respectively.

The analytic signal can be written as

$$f_A(x) = |f_A(x)| \exp\left(i\phi(x)\right) \tag{3.40}$$

Here $|f_A(x)|$ is called the local amplitude and $\phi(x)$ is the local phase of f.

Quaternion analytic signal is an approach used to extend the analytic signal in 2D. A quaternion may be represented in hypercomplex form as

$$q = a + bi + cj + dk \tag{3.41}$$

where a, b, c and d are real, i, j and k are complex operators where, $i^2 = j^2 = k^2 = -1, ij = -ji = k, jk = i, kj = -i, ki = j, ik = -j$

A quaternion has a real part and an imaginary part. The real part is referred to as the scalar part of the quaternion S(q). The imaginary has three components and can be used as a vector quantity, often denoted by

$$V(q) = bi + cj + dk \tag{3.42}$$

A quaternion with a zero real or scalar part is called a pure quaternion. The modulus and conjugate of a quaternion follow the definitions for complex numbers as given by

$$|q| = \sqrt{a^2 + b^2 + c^2 + d^2} \tag{3.43}$$

$$q = a - bi - cj - dk \tag{3.44}$$

Euler's formula for the complex exponential generalizes to hypercomplex form

$$e^{u\varphi} = \cos\varphi + u\sin\varphi \tag{3.45}$$

where $u = \frac{V(q)}{|V(q)|}$ is a unit pure quaternion. Any quaternion may be represented in polar form as $q = |q|e^{u\varphi}$, where u and φ are referred to as the Eigen-axis and Eigen-angle of the quaternion respectively. The u identifies the direction of the vector part and may be regarded as a true generalization of the complex operator i, since $u_2 = -1$.

$$\varphi = \arctan\frac{|V(q)|}{S(q)} \tag{3.46}$$

where φ is analogous to the argument of a complex number, but is unique only in the range [0, π] because a value greater than the range can be reduced to this range by negating or reversing the Eigen-axis.

Let f be the real two-dimensional image and F^q be quaternionic Fourier transform. In the quaternionic frequency domain, the quaternionic analytic image of a real image is defined as

$$F_A^q(\mu) = \big(1 + \mathrm{sgn}(u)\big)\big(1 + \mathrm{sgn}(v)\big)F^q(\mu) \tag{3.47}$$

where μ = (u, v). This can be expressed in the spatial domain as follows

$$f_A^q(x) = f(x) + n^T f_{Hi}(x) \tag{3.48}$$

where $n^T = (i, j, k)^T$ and f_{Hi} is a vector which consists of the total and the partial Hilbert transforms of f and is given by (Xu 2012).

$$f_{Hi} = \begin{pmatrix} f_{Hi_1}(x) \\ f_{Hi_2}(x) \\ f_{Hi}(x) \end{pmatrix} \tag{3.49}$$

The local features amplitude, phase, orientation and instantaneous frequency of an image are extracted from the quaternion analytic image of the form,

$$f_A^q(x) = f(x) + n^T f_{Hi}(x) \tag{3.50}$$

The analytic image of quaternion Hilbert transform spectral analysis used in 2D (Qiao et al 2009) is represented as

$$f_a^q(x) = Ae^{u\varphi} \tag{3.51}$$

where A, amplitude, φ, phase and u, direction in xy space of the vector part are given by

$$A = \sqrt{f(x)^2 + f_{Hi_1}(x)^2 + f_{Hi_2}(x)^2 + f_{Hi}(x)^2} \tag{3.52}$$

$$\varphi = \arctan\left(\left|f_{Hi_1}(x)i + f_{Hi_2}(x)j + f_{Hi}(x)k\right| / f(x)\right) \tag{3.53}$$

$$u = \left((f_{Hi_1}(x)i + f_{Hi_2}(x)j + f_{Hi}(x)k) / \left|f_{Hi_1}(x)i + f_{Hi_2}(x)j + f_{Hi}(x)k\right|\right) \tag{3.54}$$

where f_{Hi} is a vector which consists of the total and the partial Hilbert transforms of f.

$$\text{diffx} = \varphi(x+1, y) - \varphi(x, y) \tag{3.55}$$

$$\text{diffy} = \varphi(x, y+1) - \varphi(x, y) \tag{3.56}$$

diffx and diffy are the instantaneous frequencies which represent the frequency in horizontal and vertical directions (Qiao et al 2009).

Statistical features which include mean, standard deviation, kurtosis, skewness and energy are extracted for wavelet approximation and detail coefficients, QHT local features and QWT approximation, horizontal, vertical and diagonal components.

3.10 STRUCTURE TENSOR ANALYSIS

The input image I(x, y) is subjected to structure tensor analysis. Structure tensor matrix J can be derived using two different methods namely, pixelwise and piecewise techniques. The structure tensor of an image I is a symmetric positive semi-definite 2× 2 tensor given by

$$J(\nabla I) = K_j * (\Delta I . \Delta I^T) \tag{3.57}$$

where ΔI is the image gradient and K_j, a Gaussian weighting function with sigma ρ. The smoothing is performed by convolution of the matrix components with a Gaussian kernel with standard deviation j. Local shifting of edge locations is minimized by applying a Gaussian smoothing filter pixel-by-pixel to the structure tensor (Nicolescu & Medioni 2003, Bigun et al 2004).

The Eigen-analysis of this structure tensor gives the orientation of the local image features:

$$J(\nabla I) = [v_1, v_2] \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \tag{3.58}$$

where v_1,v_2 are Eigen vectors which give the local image orientations and Eigen values μ_1, μ_2 describes the average contrast in those directions (Kroon & Slump 2009).

Anisotropic nonlinear diffusion is done to construct nonlinear structure tensors by using the Eigen vectors v_1 and v_2, Eigen value μ_1 and μ_2

and gradient ΔI. Here nonlinear diffusion creates a family of images {u(x, t) | t ≥ 0} of initial image I(x, y) by solving the partial differential equation (Kroon et al 2010, Weickert 1998)

$$\frac{\delta u}{\delta t} = \nabla \cdot \left(D_T \nabla u \right) \tag{3.59}$$

where the diffusion tensor D_T is given as

$$D_T = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \tag{3.60}$$

where

$$D_{11} = k_1 v_{11}^2 + k_2 v_{21}^2 \tag{3.61}$$

$$D_{22} = k_1 v_{12}^2 + k_2 v_{22}^2 \tag{3.62}$$

$$D_{12} = k_1 v_{11} v_{12} + k_2 v_{21} v_{22} \tag{3.63}$$

and

$$\begin{aligned} k_1 &:= \alpha, \\ k_2 &:= \begin{cases} \alpha & \mu_1 = \mu_2, \\ \alpha + (1-\alpha) \exp\left(\dfrac{-C}{\left(\mu_1 - \mu_2\right)^{2m}} \right) & else, \end{cases} \end{aligned} \tag{3.64}$$

where the exponential function was chosen to ensure the smoothness of D and the small positive α ε (0,1) keeps D ($J(\nabla u_\sigma)$) uniformly positive definite. The constants may be chosen as C=1, m=1 and α=0.001 (Weickert 1998).

Thus diffusion matrix D from structure tensor J is calculated. The largest to smallest Eigenvalues λ_1, λ_2, λ_3, mean and trace are derived

following diagonalization of the full diffusion tensor. Energy is determined from the Eigen values and is given by

$$E = Trace(J) = \lambda_1 + \lambda_2 + \lambda_3 \tag{3.65}$$

For the piecewise analysis, the local neighborhood which is defined as a discrete square region centered at x, y is created for scale factor of S=1, 2, 3……10 following the same procedure as pixelwise method. The tensor matrix is created for the region and anisotropic indices are derived based on the corresponding Eigen values.

3.11 ANISOTROPY FEATURE ESTIMATION

The Anisotropy Index (AI) is estimated from wavelet transforms, Gabor wavelets, Quaternions and structure tensor. The largest to smallest Eigenvalues, which are given by Eigenvalue I (λ_1), Eigenvalue II (λ_2), Eigenvalue III (λ_3) and λ, average of these Eigen values, are calculated for all the spectral and spatial derived images. Further from these Eigen values, Relative Anisotropy (RA) and fractional anisotropy are derived as given by

$$RA = \sqrt{\frac{1}{3}} \frac{\sqrt{(\lambda_1 - (\lambda))^2 + (\lambda_2 - (\lambda))^2 + (\lambda_3 - (\lambda))^2}}{(\lambda)} \tag{3.66}$$

$$FA = \sqrt{\frac{3}{2}} \frac{\sqrt{(\lambda_1 - (\lambda))^2 + (\lambda_2 - (\lambda))^2 + (\lambda_3 - (\lambda))^2}}{\sqrt{\lambda_1^{\,2} + \lambda_2^{\,2} + \lambda_3^{\,2}}} \tag{3.67}$$

FA represents anisotropic diffusion weighted against total diffusion (Budde et al 2012).

3.12 LACUNARITY ANALYSIS

The gliding box and differential box counting algorithms are used to estimate lacunarity. The Region Of Interest (ROI) is subjected to five different methods of lacunarity namely, Lacunarity Binary (LB), Lacunarity Gray (LG), Lacunarity Range (LR), Lacunarity Third Moment (LTM) and Lacunarity Differential Box Counting (LDBC). Lacunarity values are calculated for different box sizes r = 2, 3, 4.....10.

In the LB method, the ROI is converted into binary image using auto threshold binarization algorithm. A unit box of size r is chosen and the number of set points p within it is counted. This procedure is then repeated as the box is centred about each point within the set, creating a distribution of box masses B (p,

'r). This distribution is converted into a probability distribution given by

$$Q(p,r) = \frac{B(p,r)}{B(r)} \tag{3.68}$$

where B(r) is the total number of boxes of size r. The first and second moments of the box mass probability distribution are expressed as

$$Z^{(1)}(r) = \sum_p pQ(p,r) \tag{3.69}$$

$$Z^{(2)}(r) = \sum_p p^2Q(p,r) \tag{3.70}$$

The gliding box lacunarity is then defined as

$$LB = \frac{Z^{(2)}r}{Z^{(1)}(r)^2} \tag{3.71}$$

The moments can be accumulated as the box is slided through the data set, which provides an efficient method to perform the algorithm (Tolle et al 2003). The first and second moments, gliding box lacunarity are modified as

$$Z^{(1)}(r) = \frac{1}{B(r)} \sum_p pB(p,r) \tag{3.72}$$

$$Z^{(2)}(r) = \frac{1}{B(r)} \sum_p p^2 B(p,r) \tag{3.73}$$

$$LB = \frac{B(r) \sum_p pB(p,r)}{\left[\sum_p pB(p,r)\right]^2} \tag{3.74}$$

The number of boxes containing exactly p data elements in the i^{th} box of radius r is defined as:

$$T(i,p,r) = \begin{cases} 1, for\ p(i,r) = p \\ 0, for\ p(i,r) \neq p \end{cases} \tag{3.75}$$

The distribution of box masses and second moment can be expressed as

$$B(p,r) = \sum_{i=1}^{B(r)} T(i,p,r) \tag{3.76}$$

$$Z^{(2)}(r) = \frac{1}{B(r)} \sum_{i=1}^{B(r)} \sum_p p^2 T(i,p,r) \tag{3.77}$$

Since T (i, p, r) is zero everywhere except at p = p (i, r), the sum $\Sigma_p p^2 T$ (i, p, r) reduces to P (i, r) ^{2}. Therefore, the equation for the second moment reduces to

$$Z^{(2)}(r) = \frac{1}{B(r)} \sum_{i=1}^{B(r)} p(i,r)^2 \tag{3.78}$$

Similarly the first moment can be written as

$$Z^{(1)}(r) = \frac{1}{B(r)} \sum_{i=1}^{B(r)} p(i,r) \tag{3.79}$$

Using the reduced expressions, the gliding box lacunarity can be expressed as

$$LB = \frac{B(r) \sum_{i=1}^{B(r)} p(i,r)^2}{\left[\sum_{i=1}^{B(r)} p(i,r)\right]^2} \tag{3.80}$$

The process is repeated by moving the boxes along the column of the image (Tolle et al 2003).

In LG method, the box of different size r = 2, 3… 10 slide on the entire image and mass M inside each box is calculated. Initially, box of size 2× 2 is initially placed over the upper left corner of the image. The box mass M is calculated as the summation of pixel intensities inside the placed box. Then the box is moved one column to the right, and the box mass M is counted again for all the boxes and process is repeated for the entire image. Average of all box masses is computed as expected mass E(M) (Dong 2000).

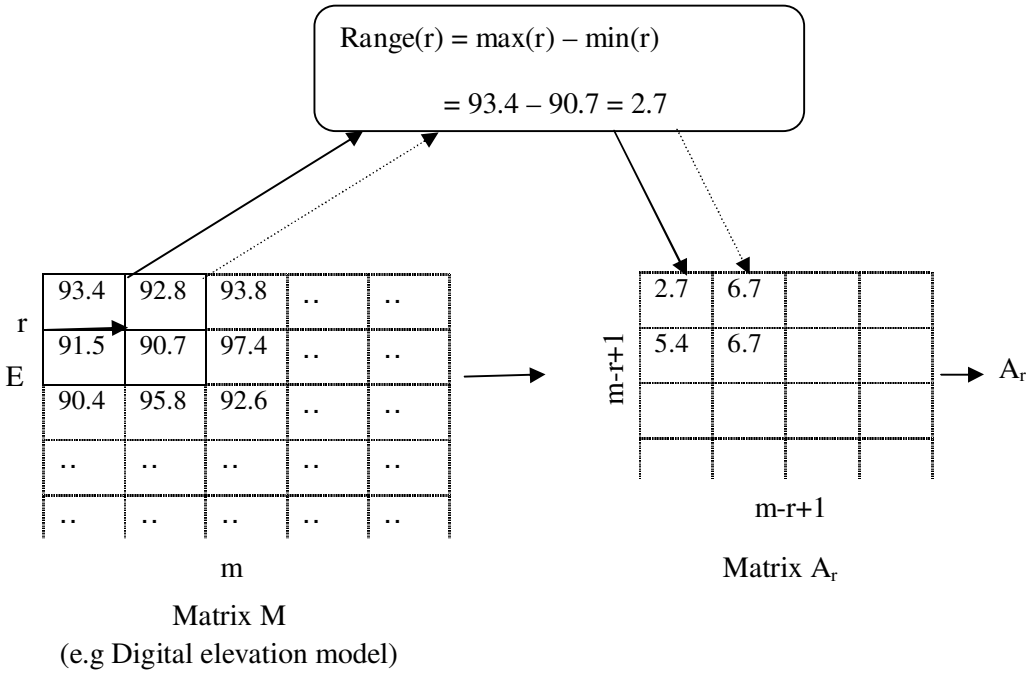

Figure 3.3 Illustrative example of Lacunarity Range method

The values of lacunarity (LG) are calculated using

$$LG = E\left\{\left(\frac{M_r}{E(M_r)} - 1\right)^2\right\} \tag{3.81}$$

where M is the mass and E $\{\cdot\}$ denotes the expected value of the entire box mass M. The process is repeated for all other box sizes r and lacunarity is calculated (Hoechsetetter et al 2011).

The gradient-based LR method is implemented using the following procedure. For a given image of size m × m pixels, a window referred to as box of size r × r is placed upon one corner of the image. The range of the values, also known as box mass, is obtained by finding the difference between maximum and minimum pixel intensity values within the box of size r × r. The box is shifted by one pixel, so that its new position overlaps with the previous one. The contained range of values in a box is recorded in a cell of resulting matrix A_r. The entire ROI is processed until all possible r × r-neighborhoods are accounted for and A_r reaches an extent of (m - r + 1) × (m - r + 1) pixels as illustrated in Figure 3.3

The lacunarity of the matrix A_r for box size r is finally calculated according to the following formula:

$$LR = \frac{S^2(A_r)}{\overline{S}^2(A_r)} + 1 \tag{3.82}$$

where S is arithmetic mean and S^2 is the variance. Thus the lacunarity value is obtained for each box size r.

Lacunarity based on LTM was calculated using gliding method. It has been estimated by a new box counting algorithm called as True box

counting method and is implemented using the formula (Esgiar & Chakraborty 2005).

$$LTM = \frac{M^3_{box}(L)}{(M_{box}(L))^3} - 3 * \frac{M^2_{box}(L)}{(M_{box}(L))^2} + 2 \tag{3.83}$$

where L is the box size and $M^q_{box}(L)$ represent various mass moments. Thus higher order lacunarity is calculated and plotted for various box sizes for further analysis.

In LDBC method, a cube of size r × r × r, where r =2, 3, 4, 5, 6, 7, ...10, is placed over the upper left corner of an image window of size W × W. W is considered as an odd number and r < w. Depending on the pixel values in the r × r gliding-box, a column with more than one cube may be needed to cover the image intensity surface. Typical cube with moving window size of 9×9 and gliding best size of 3 is represented in Figure 3.4. Numbers are assigned to the cubic boxes from bottom to top. The relative height of the column is defined as

$$n_{r(i,j)} = v - u - 1 \tag{3.84}$$

where i and j are image coordinates, u and v are the minimum and maximum

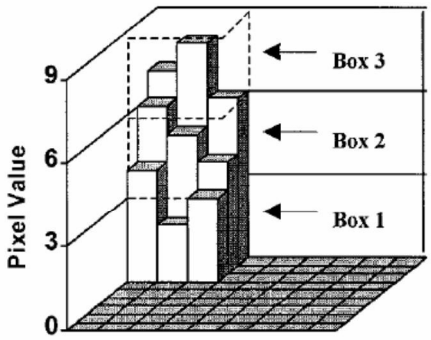

Figure 3.4 Differential box counting method

pixel values of each gliding-box respectively. When the r x r gliding-box moves throughout the W × W image window, the mass is computed as

$$M_r = \sum_{i,j} n_r(i,j) \tag{3.85}$$

The lacunarity value is determined using this mass using Equation 3.82 (Dong 2000). The mean lacunarity is then calculated by taking average of lacunarity values for different box sizes.

3.13 SUCCOLARITY ANALYSIS

The ROI is subjected to succolarity analysis using percolation algorithm. Four different orientations of binary image which include top to bottom (orientation (I)) and bottom to top (orientation (II)) right to left (orientation (III)) and left to right (orientation (IV)) are extracted using flood filling technique. In the binarized input image, zero value pixels and pixels with value one are considered as open and blocked sites respectively. In percolation algorithm, path for open sites is checked from the first pixel of an image. If an open site is found, then neighboring pixels are checked for four-connectivity. Depth first search technique checks all the pixels in the binary image to ensure that open sites are filled in Boolean matrix. The final Boolean matrix is called the flooded image. Each flooded image is analyzed using a box counting approach. A grid of size D=150 with box size l of 1 X 1 is placed on the flooded image. The number of flooding pixels NP(l) within each box is calculated. Increasing weights of w = 1, 2, 3…D are assigned vertically for orientations I and II flooded images.

A pressure matrix (*PR*) is created by calculating the centroid (pc) of the box which depends upon the box size (l) and the assigned weights. Succolarity for a given direction (dir) is calculated using the following equation (Melo & Conci 2011).

$$\sigma(dir) = \frac{\sum_{k=1}^{n} NP(l).PR(pc,l)}{\sum_{k=1}^{n} PR(pc,l)} \tag{3.86}$$

The process is repeated for various grid sizes D=150, 50, 10, 6 and 3 and with corresponding box size of l = 1, 3, 15, 25 and 50.

3.14 CORRELATION ANALYSIS

Correlation measures the association between two variables and quantitates the strength of their relationship. The features derived from femur radiographic images using wavelets, Gabor, quaternions, structure tensor, lacunarity and succolarity are correlated with macrostatistic feature apparent porosity and are analysed using their correlation coefficients.

3.15 FEATURE SELECTION USING PRINCIPAL COMPONENT ANALYSIS

Principal component analysis is performed using an Eigen analysis of the correlation matrix. It is based on the statistical representation of a random variable. For a random vector population $x = (x_1,...,x_n)^T$, the mean (μ_x) and covariance matrix (C_x) are given by

$$\mu_x = E(x) \tag{3.87}$$

$$C_x = E\left\{ (x-\mu_x)(x-\mu_x)^T \right\} \tag{3.88}$$

The components of C_x, denoted by C_{ij}, represent the covariance between the random variable components and x_i. The component C_{ii} is the variance of the x_i component. The variance of a component indicates the

spread of the component values around its mean value. From the covariance matrix, the Eigenvectors e_i and the corresponding Eigen values λ_i are computed from the solutions of the following equation

$$C_x e_i = \lambda_i e_i, \ i = 1, ..., n \tag{3.89}$$

Eigenvectors are ordered in descending Eigen values with the first Eigenvector having the direction of largest variance of the data to find directions in which the data set has the most significant amounts of energy. For a matrix *(A),* consisting of Eigenvectors of the covariance matrix as the row vectors, a data vector *(x)* is transformed into point in the orthogonal coordinate system defined by the Eigenvectors and is represented as

$$y = A(x - \mu_x) \tag{3.90}$$

The original data vector *x* can be reconstructed from *y* using

$$x = A^T y + \mu_x \tag{3.91}$$

The original vector *x* was projected on the coordinate axes defined by the orthogonal basis. The original vector was then reconstructed by a linear combination of the orthogonal basis vectors. Representing only a few basis vectors of the orthogonal basis instead of using all the eigenvectors of the covariance matrix which are the principal components, the matrix having the first *K* eigenvectors as rows is denoted by A_k and a similar transformation can be created for which

$$y' = A_k (x - \mu_x) \tag{3.92}$$

$$x' = A_k{}^T y' + \mu_x \tag{3.93}$$

The original data vector is projected on the coordinate axes having the dimension *K* and transforming the vector back by a linear combination of the basis vectors. This minimizes the mean-square error between the data and this representation with given number of Eigenvectors (Burstyn 2004, Aguado et al 2008).

The original feature space consists of each 219 parameters obtained from normal and abnormal compressive and tensile subjects. PCA is employed in transforming this feature space into a new space and the components that account for most of the variability are retained whereas the remaining components are ignored. The principal components obtained from PCA are analyzed for ranking the most significant features (Malhi & Gao 2004). The percentage variances between the various parameters are estimated for the normal and abnormal subjects. The principal components that explain the maximum percentage variance are chosen and the corresponding component magnitudes are analyzed. The parameters with highest magnitudes in the loadings of the principal components are chosen for further classification using ELM.

3.16 CLASSIFICATION USING EXTREME LEARNING MACHINE

The structure of ELM contains an input layer, hidden layer and an output layer. This single hidden layer feed forward network with K hidden neurons are trained with activation function g(x) to learn N distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}\ldots\ldots, x_{in}]^T \in R^n$ represent the significant features, $t_i = [t_{i1}, t_{i2}\ldots\ldots, t_{in}]^T \in R^n$ represent the targets. The main goal of the training process is to determine the network weights $w_j = [w_{j1}, w_{j2},\ldots, w_{jn}]^T$ and hidden layer biases $ß_i = [ß_{i1}, ß_{i2}, \ldots, ß_{in}]^T$. The SLFN aims to minimize the difference between network output and corresponding test target. The network model is mathematically modelled as

$$\sum_{i=1}^{\hat{N}} \beta_i g(w_i . x_j + b_j) = 0_j, j = 1,..., N \quad (3.94)$$

where g(x) is the activation function. The equation can be written as $H\beta = T$ form which the output weights are estimated as $\tilde{\beta} = H^{\dagger}T$, where $H^{\dagger}$ is the pseudo-inverse of H. Different basis which include sigmoid and sine functions of ELM are considered (Huang 2006).

3.17 PERFORMANCE ESTIMATION

The performance of the classifier is tested using accuracy. It is defined as

$$\text{Accuracy} = (TP+TN) / (TP+FP+TN+FN) \quad (3.95)$$

where True Positive (TP) and True Negative (TN) correspond to classification of abnormal as abnormal and normal as normal respectively, False Positive (FP) and False Negative (FN) is classification of a normal data as abnormal and classification of abnormal data as normal (Zhang et al 2008).