CHAPTER 3

AN EFFICIENT ELM BASED APPROACH FOR SCREENING OF CHRONIC OBSTRUCTIVE PULMONARY DISEASE (COPD) FROM CHEST CT SCANS WITH LAPLACIAN GAUSSIAN FILTERING

3.1 INTRODUCTION

Medical imaging is one of the most useful diagnostic tools available in modern medicine. Medical diagnostic and imaging system are ubiquitous in modern health care facilities. The advantages of early detection of potential lesions and suspicious masses within the bodily tissue have been well established. It can detect and assess many different types of injuries, diseases, and conditions with the aid of the medical imaging that allows medical personnel to look into living cells non-intrusively (Keyvanfard et al 2011).

To acquire the medical images of the organs and internal structures of the body, X-rays, gamma rays, ultrasound, infrared thermograph and magnetic fields are used. In interventional radiology, imaging procedures are combined with other techniques to treat certain diseases and abnormalities to arrive at a conclusion without any doubt. Image processing is a series of operations that are applied to the images to enhance, alter, or select regions of interest.
Diagnoses of the disease are usually based on visual recognition of abnormal cells and tissues and the results can help the plan in development of methods to optimize patient treatment in the right direction. Doctors and technicians can more easily and exactly make a diagnosis, decide on a treatment, prescribe medication, and perform surgery or any other treatment. The main aim of the authors is to detect and extract the tumour from CT images. The CT images that reveal a suspicion for cancer are found out for more detailed examination by the attending physicians. There are several image processing methods implemented for the detection of various diseases in CT image (Amandeep Singh et al 2012).

Digital CT is a technique for recording images in computer code instead of on X-ray film. The images are displayed on a computer monitor and can be enhanced (lightened or darkened) before they are printed on film. Images can also be manipulated, the radiologist can magnify or zoom in on an area. This screening will generate a large number of CT images to be determined by a small number of radiologists resulting in misdiagnosis due to human errors caused by visual fatigue. The sensitivity of the human eye decreases with increasing number of images. Hence, it may be helpful for a radiologist, if a computer-aided system is used for the detection of various diseases in CT images. CAD involves the use of computers to bring suspicious areas on a CT to the radiologist’s attention. It is used after the radiologist has done the initial review of the CT. There are several image processing methods implemented to extract several diseases from the CT image for better view of the area and shape of the diseases. In some cases, the primary objective is to enhance the CT image.

This technique is previously proposed for X-ray lung cancer images using SVM classification technique with LGXP growing technique. In this chapter, the approach is explained. Here, first the input image is
pre-processed, the lung region is segmented from that image, cavity region in that lung region is segmented, some features are extracted for training the classifier and the SVM classifier is used to identify the tuberculosis affected lung (Rui 2010). The pre-processing is done by using the Laplacian Gaussian Filter to avoid the noise in the input image and to increase the image quality. The lung segmentation is done by comparing the region growing technique and the Local Gabor XOR Pattern (LGXP) based region growing technique. The cavity segmentation is done by evaluating the pixel range in the segmented lung region and setting a threshold value from that evaluated pixels and comparing every pixel with that threshold value. After the lung and cavity segmentation, some parameters are chosen to train the classifier to identify whether an X-ray image is normal or tuberculosis affected. The classifier used in the proposed technique is the ELM classifier. The ELM Classifier is then trained using the parameters chosen from the sample chest CT scan images to identify the normal lung and tuberculosis affected lung.

3.2 ANALYSIS OF LOCAL GABOR XOR PATTERN (LGXP)

3.2.1 Gabor

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representation of the Gabor filter are similar to those of the human visual system, and it has been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave (Sivalingamaiah & Venakramana Reddy 2012).

In this section, Gabor wavelet representation, LBP and LGXP respectively are described and compared between the local patterns.
A. Gabor Wavelet Representation

It was originally introduced by Dennis Gabor for 1-D signals and Daugman extended the Gabor filter for 2-D. The Gabor filters are band pass used for feature extraction. Gabor filters similar to STFT or windowed Fourier transform, are both frequency-selective and orientation-selective and has optimal joint resolution in spatial and frequency domain.

The 2-D Gabor filter is a sinusoidal plane of a particular frequency and orientation, modulated by a Gaussian envelope. Typical Gabor features, such as Gabor feature space consists of a response calculated by Gabor filters at several different orientation scales (frequencies) and a filter bank. A Bank of filters is used with different orientations so as to extract frequency information and thereby the features at different orientations, since all the facial features are not present at the same orientation. Scaling is done at each orientation so as to get maximum frequency information at each orientation i.e., orientation and scaling helps in extracting maximum frequency information (Santosh & Anil 2012).

A 2-D Gabor kernel is defined in Equation (3.1),

\[
\mu_{\mu,\nu}(z) = \frac{||k_{\mu,\nu}||^2}{\sigma^2} e^{-\left(\frac{||k_{\mu,\nu}||^2|||A||^2}{2\sigma^2}\right)} \left[ e^{k_{\mu,\nu}^2} - e^{-\frac{\sigma^2}{2}} \right] \tag{3.1}
\]

where \( \mu \) and \( \nu \) defines the orientation and scale of Gabor kernels, \( z = (x, y) \), \( |\cdot| \) denotes the norm operator and wave vector \( k_{\mu,\nu} \) is defined as in Equation (3.2),

\[
k_{\mu,\nu} = k_{\nu} e^{i \phi_{\mu}} \tag{3.2}
\]
where \( k_v = k_{max}/f^v \) and \( \phi_\mu = \pi \mu / 8 \), \( k_{max} \) is the maximum frequency and \( f \) is the spacing factor between the kernels in the frequency domain.

A filter bank consisting of several filters need to be used because relationship between the responses provide the basis for the distinguishing objects. The selection of discrete rotation angles is such that the orientation must be spaced uniformly as defined in Equation (3.3),

\[
\phi_\mu = \frac{2\pi \mu}{n}, \mu = \{0, 1, \ldots, n - 1\}
\]  

(3.3)

Where \( \phi_\mu \) is the \( \mu^{th} \) orientation and \( \mu \) is the total number of orientations to be used. The computation can be reduced to half since the angle of response \([\pi \ 2\pi]\) is a complex conjugate on the responses on \([0 \ \pi]\) in case of real valued input.

Frequency selection is given as in Equation (3.4),

\[
k_v = f^{-v}k_{max}, \ v = \{0, 1, \ldots, m - 1\}
\]  

(3.4)

Useful values for \( f \) includes \( f=2 \) for octave spacing and \( f = \sqrt{2} \) for half octave spacing.

The Gabor kernels in Equation (3.1) are self-similar, since they can be generated from one kernel, the mother wavelet, by scaling and rotation via wave vector \( k_{\mu \nu} \). Each kernel is a product of a Gaussian envelope and complex plane wave, while the first term in square brackets in Equation (3.1) determines the sinusoidal part of the kernel and the second term is used to get zero DC response, that is the Gabor filter is made insensitive to the background luminance level, \( \sigma \) determines the ratio of Gaussian window width to wavelength.
Gabor wavelets of five different scales $v = \{0,1,2,3\}$ and eight different orientations $\mu = \{0,1,2,3,4,5,6,7\}$ are used for feature extraction.

B. **Local Binary Patterns (LBP)**

The LBP operator assigns a label to every pixel of an image by thresholding the $3 \times 3$ neighborhood of each pixel with the center pixel value and considering the result as a binary number. For example, as shown in Figure 3.1, by applying LBP “11010011” is the designed pattern of the center pixel.

![Figure 3.1 LBP Operator Defined in Three Neighborhood](image)

Operator to a facial image, one pattern map can be computed. The pattern map is then divided into many blocks and histogram is computed in each block and concatenated together to form the description of the input facial image (Ahonen et al 2006).

C. **Local Gabor XOR patterns**

The basic idea of this method is that, as shown in Figure 3.2, the phases are first quantized into different ranges and LXP operator is then applied to the quantize phases of the central pixel and each of its pixel. Finally, the resulting binary labels are concatenated together as the local pattern of the central pixel.
3.3 MACHINE LEARNING CLASSIFICATION

Image classification involves analyzing the numerical properties of various image features and organizing the data into categories. Usually, two phases of classification algorithms are employed: training and testing (Kekre et al 2013).

In the initial training phase, characteristic properties of typical image features are identified and based on these, a unique description of each classification category or training class, is created. The description of training classes is an extremely important component of the classification process.

The motivating criteria for constructing training classes are that they are,

- Independent: This means, a change in the description of one training class should not change the value of another.
- Discriminatory: Here different image features should have significantly different descriptions.
- Reliable: All the image features within a training group should share the common definitive descriptions of that group.
In testing, which is the next phase, accuracy of the classifier is measured. The accuracy can be determined by applying the classifier to an independent training set of objects with known classifications. Knowledge of the accuracy is necessary both in the application of the classifier and also in the comparison of different classifiers.

A convenient way of building a parametric description of this sort is via a feature vector, where ‘n’ is the number of attributes which describe each image feature and training class.

This representation allows consideration of each image feature as occupying a point, and each training class as occupying a sub-space (i.e. a representative point surrounded by some spread, or deviation), within the n-dimensional classification space. Viewed as such, the classification problem is that of determining to the sub-space class to which each feature vector belongs.

Classification can be linear or non linear. Linear classifiers are usually the fastest classifiers. It classifies the data on the basis of a linear combination of the characteristics based on which the classification is done. The classes are divided by a linear separator in the feature space. If the feature space is two dimensional, the separator is a line, in three dimensional the separator is a plane and if ‘p’ dimensional, the linear separator is then a (p – 1) dimensional hyper plane. Non-linear classification is required when the class boundaries cannot be approximated well with linear hyper planes. An example of a non-linear classifier is K Nearest Neighbor.

There are two main ways of classification, supervised and unsupervised (Ratika et al 2010). The difference between them lies in, how the data is classified.
**Supervised Classification**

In supervised classification, there are predetermined classes. Statistical processes (based on an a priori knowledge of probability distribution functions) or distribution free processes can be used to extract class descriptors. These classes can be regarded as a previously decided finite set. After classification, certain segment of data will be labeled with these classes. The task of the algorithm is to search for patterns and construct mathematical models. These models are then used to find out the measure of variance in the data and classify it. The examples of supervised classification are Decision tree induction and Naive Bayes classifier.

**Unsupervised Classification**

Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In this type of classification, the classes are not pre decided. The basic task of the classifier is to automatically develop the classes or the labels. The algorithm is not told how the data is to be grouped; it is something it has to arrive at by itself. This is a difficult decision to make. The classifier looks for similarities between the data and then determines which of these can form a group and can be classified under one label. The classes are also called clusters. The example of unsupervised classification is K-means classification. In K-means, the classifier is given in advance, the number of clusters to be formed.

In unsupervised classification, since there is no defined strategy, the algorithm starts from one point and performs iterative repetitions to reach a stable configuration that makes sense. The results can vary widely and depend largely on the first few steps taken.
Support Vector Machine (SVM)

SVM is an example of supervised learning classification. It is a binary linear classifier which takes an input and decides to which of the two classes it belongs to the classifier which is trained first using a set of training examples. The training examples are pre-marked as belonging to one of the two categories and based on these examples, the SVM classifier builds a model that assigns new examples to their suitable classes. The training examples are represented as points in space and are mapped such that there is a clear gap which divides the examples belonging to separate classes. The new examples are then mapped into the same space by analyzing the class which suits them better.

Apart from classification, a support vector machine can also be used for regression etc. It constructs a hyper plane that separates the two classes with gap as wide as possible. A hyper plane is regarded as ‘good’ if it has the largest distance with the nearest training data point of any class. This reduces the error of the classifier.

Figure 3.3 Hyper Planes Separating Two Classes
In the above Figure 3.3, although both red and the blue hyper planes \((H_1, H_2, H_3)\) are separating the two classes \((x_1, x_2)\) entirely, the red hyper plane is doing it such that its distance from the nearest point of the two classes is maximum. Hence, the red hyper plane is the most optimum hyper plane. The data points which are closest to the hyper plane are called support vectors.

**Minimum Distance Classifiers**

The minimum distance classifiers, as the name suggests, classifies the data on the basis of the distance between the data points in the feature space. It classifies in such a manner, so as to minimize the distance between the data and the class in multi-feature space. The index of similarity here is the distance so that the minimum distance is identical to the maximum similarity.

**Artificial Neural Networks (ANN)**

ANN classifiers are inspired by biological neural systems. The nerve cells in the human brain are called neurons. These are linked with the other neurons via strands of fiber called axons. Whenever the neurons are stimulated, axons transmit nerve impulses from one neuron to another. Extensions from the cell body of the neuron are called dentrites. Dentrites connect one neuron to the axons of the other neurons. The connection between a dentrite and an axon is called a synapse. It has been discovered that the human brain learns by changing the strength of the synaptic connection between the neurons when repeatedly stimulated by the same impulse. The ANN has a structure analogous to the human brain. It is composed of an interconnected assembly of nodes and direct links which is shown in Figure 3.4. These models can be trained for the purpose of classification. The simplest model is called perceptron.
The perceptron consists of two kinds of nodes, the input nodes and the output node. Input nodes are used to represent the input attributes and the output node represents the model output. The nodes in the Neural Network are called neurons. In a perceptron, each of the input nodes is connected to the output node via a weighted link. The weight of the link is used to emulate the strength of the synaptic connection between the neurons. The perceptron computes the output value by calculating a weighted sum of the inputs, subtracting a bias factor 't' from the sum and then examines the sign of the result. Training of the perceptron network involves changing and adapting the weight of the links until they fit the input output relationship of the training data which means the outputs of the perceptron become consistent with the true outputs of the training data.

In an ANN model, an input node simply transmits the value it receives to the outgoing link without any transformation. On the other hand, the output node is a mathematical device which computes the weighted sum of its inputs, subtracts the bias term and produces an output that depends on the sign of the result. The sign function is an activation function for the output neuron, its value is +1 if positive and -1 if negative.
Once the model is trained, it can now be used to test new examples and classify them. The Perceptron learning algorithm converges to an optimum solution for linearly separable classification problems. If the problem is not linearly separable the algorithm fails to converge. In that case, Multilayer Artificial Neural Network is needed.

A multilayer Neural Network has a more complex structure than a perceptron. It contains several intermediate layers between its input and output layers. These intermediate layers are called hidden layers and the nodes in these layers are called hidden nodes. There are two types of networks, feed forward and recurrent networks. In feed forward networks, the nodes in one layer are connected only to the nodes in the next layer, while in a recurrent network, the links may connect the nodes in the same layer or in the previous layers. The activation function may be other than sign function, also like linear or sigmoid function.

These activation functions allow the hidden and the output nodes to produce non-linear outputs. This type of complex structure can classify problems which are not linearly separable and have non linear solutions. This model will converge to the right solution when sufficient training data is provided.

**K-Nearest Neighbor (k-NN) Classifier**

In pattern recognition and classification, the k-nearest neighbor algorithm (k-NN) is an algorithm for classifying data objects based on closest training examples in the feature space. The k-NN is a type of instance based or lazy learning, where the function is only approximated locally and all the computations are postponed till classification. The k-nearest neighbor algorithm is the simplest among all the machine learning algorithms. An object is classified by a majority vote of its neighboring data, and the test
object is assigned to the class which is most common amongst its k nearest neighbors (k is a small positive integer).

If ‘k’ is chosen to be 1, the test object is then simply assigned to the class to which its nearest neighbor belongs to. Although there is no explicit need for training in this algorithm, the neighbors can be regarded as training examples and are chosen from a set of objects for which the correct classification is known. The k-nearest neighbor algorithm is affected by the local structure of the data. The nearest neighbor rules implicitly compute the decision boundary effectively.

The training examples can be regarded as vectors in a multidimensional feature space, each belonging to a class. Hence, in the training phase of the algorithm, it is required to store only the feature vectors of the training samples along with their class.

In the classification phase, k is a constant decided by the user and a test vector (can also be called query or a test point) is classified by assigning it the label which is most common among the k training data nearest to that particular test point. The class, to which its nearest neighbor belongs, is called the nearest neighbor algorithm.

In the present research, the capability of a new data mining scheme called ELM to be a decision making tool in the field of Material and Engineering industry, for classification has been investigated. The ELM modeling scheme is a new framework. Unlike the standard Neural Network, it is a SLFNs which randomly chooses the input weights and analytically determines the output weights of SLFNs, (Huang et al 2005). In theory, this algorithm tends to provide the best generalization performance at an extremely fast learning speed.
3.4 BACKGROUND AND OVERVIEW OF ELM

This is extremely good as in the past, it seems that there exists an unbreakable virtual speed barrier which classic learning algorithms cannot break through and therefore feed-forward network implementing them take a very long time to train itself, independent of whether the type of application is simple or complex. Also, ELM tends to reach the minimum training error as well as considers the magnitude of weights which is opposite to the classic gradient-based learning algorithms which only intends to reach minimum training error but do not consider the magnitude of weights. Unlike the classic gradient-based learning algorithms which only work for differentiable activation functions, ELM learning algorithm can also be used to train SLFNs with non-differentiable activation functions. According to Huang et al (2004), “Unlike the traditional classic gradient based learning algorithms facing several issues like local minimum, improper learning rate and over fitting, the ELM tends to reach the solutions straightforward without such trivial issues”.

The ELM has several interesting and significant features that are different from the traditional popular gradient-based learning algorithms for Feed-forward neural networks. The learning speed of ELM is extremely fast. In simulations reported in literature, the learning phase of ELM can be completed in seconds or less than seconds for many applications.

Previously, there existed a virtual speed barrier which most (if not all) classic learning algorithms could not break through and it was not unusual to take a very long time to train a Feed-forward network using classic learning algorithms even for simple applications.

- The ELM has better generalization performance in most cases than the gradient-based learning, such as, Back propagation. The traditional classic Gradient-based learning algorithms and
some other learning algorithms may face several issues like local minima, improper learning rate and over fitting. To avoid these issues, methods such as Weight decay and Early stopping methods may need to be often used in these classical learning algorithms.

- The ELM tends to reach the solutions straightforward without such trivial issues. The ELM learning algorithm looks much simpler than most learning algorithms for Feed-forward Neural Networks. Unlike the traditional classic gradient-based learning algorithms which work only for differentiable activation functions, as easily observed the ELM learning algorithm could be used to train SLFNs with many non-differentiable activation functions (Jain et al 2000).

### 3.5 PROPOSED TECHNIQUE FOR THE IDENTIFICATION OF CAVITY

The block diagram of the proposed approach is shown in Figure 3.5. In this Figure, some sample chest CT scan images are taken with COPD and without COPD. The sample images are then preprocessed and sent for segmentation process. There, segmenting the lung and cavity regions was done. After the lung and cavity regions are segmented from the sample images, some parameters are chosen to train the classifier. First, the preprocessing is done to find whether the COPD is affected or not. After the preprocessing process, the lung and cavity region need to be segmented. After that, the chosen parameters are given to the classifier. Here, the ELM classifier is used. The ELM classifier then identifies whether the input chest CT scan image is affected by COPD or not by comparing the parameters from the sample images and from the input image (Magesh et al 2011).
Pre-processing the Sample Images using Laplacian Gaussian Filter

Segmenting the Lung Region

Segmenting the Cavity Region

Feature Extraction

Output

ELM Classifier

Extracting the Features

Pre-Processing the Sample Images using Laplacian Gaussian Filter

Segmenting the Lung Region

Segmenting the Cavity Region

Sample Images for Training

Input Image to Test

Figure 3.5 Block Diagram of Efficient ELM Based Approach for Screening of COPD with Laplacian Gaussian Filter

3.5.1 Pre-processing

The input image is subjected to the pre-processing steps to make the image suitable for further process. The pre-process is used to load the input image to the MATLAB environment and it will remove the noise present in the input image. Here, the Laplacian Gaussian filter is used as a pre-processing technique. The image is passed through the Gaussian filter and then to the Laplacian Gaussian filter to lower the noise and to get a better image. The Laplacian Gaussian filter will also increase the image quality and the corner of the images.
Laplacian Gaussian Filter

The Laplacian is a 2-D isotropic quantity measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights the regions of rapid intensity change and is therefore used often for edge detection. The Laplacian is frequently applied to an image that has been smoothed first with something approximating a Gaussian smoothing filter to reduce its sensitivity to noise, and hence the two variants will be described together. The operator normally takes a single gray level image as input and produces another gray level image as output.

The Laplacian \( L(x, y) \) of an image with pixel intensity values \( I(x, y) \) is given in Equation (3.5),

\[
L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]  

(3.5)

Since the input image is denoted as a set of discrete pixels, a discrete convolution kernel has to be found out that can estimate the second derivatives in the definition of the Laplacian. Two commonly used small kernels are shown in Figure 3.6.

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>4</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>-1</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>8</td>
<td>-1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Figure 3.6 Two Commonly Used Discrete Approximations to the Laplacian Filter
As these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

A. **Lung Segmentation**

Lung segmentation is a process of segmenting the lungs from the chest CT scan image. The normal process of the region growing technique for segmenting the lungs is shown in the Figure 3.7. A pixel from the chest CT scan image is first chosen as default. A threshold value needs to be set for comparison to find the pixel intensity for the lung area in the chest CT scan.

The default pixel which is chosen is compared with the adjacent pixel values. If the difference between the default pixel and the adjacent pixel is greater than the threshold value, that adjacent pixel has to be excluded. If the difference between the default pixel and the adjacent pixel is less than the threshold value, that adjacent pixel have to be included for region growing. All the pixels except the left pixels are compared with its adjacent pixels by keeping one pixel as default. The process of normal region growing technique is shown in the Figure 3.7.

In the present research, the normal region growing technique has been compared with the LGXP based region growing technique to segment the lungs from the chest CT scan image (Shufu et al 2010). The LGXP technique is used to find the texture image.
Figure 3.7 Block Diagram of Normal Region Growing Technique

The LGXP based region growing technique is as follows. In LGXP technique, the Gabor Phase Technique is applied on every pixel in the chest CT scan image. The Gabor Phase Technique will convert all the pixel values to phase values (0 to 360). After converting all the pixel values to phase values, the quadrant in which these phase values come has to be found. Each quadrant has certain values. For the first quadrant, the value is zero and for the second quadrant, the value is one and for the third quadrant, the value is two and for the fourth quadrant, the value is three. After that, a default phase value of a pixel is chosen and the quadrant in which this phase value comes is checked and respective quadrant value is assigned to that pixel. After assigning the respective quadrant value to the default pixel, the adjacent pixel’s phase values are to be checked and respective quadrant values assigned to those adjacent pixels.

The adjacent pixel’s value is then converted as zero which has the same quadrant value as that of the default pixel. If the adjacent pixels value does not have the same quadrant value as that of the default pixel, the adjacent pixel’s value is converted as one. Now, the pixel values would be like binary values as zeros and ones.
After converting the pixel values as binary format, that binary format is made as a decimal value and that decimal value is applied to the default pixel. The process of taking the binary value is shown in the Figure 3.8. Likewise, this LGXP process is applied for all the pixels in the chest CT scan by keeping one pixel as default. The sample process of LGXP technique is shown in the Figure 3.8.

![LGXP Technique Diagram](image)

**Figure 3.8 LGXP Technique**

After applying the LGXP technique in all the pixels, the region growing technique is implemented for segmenting the lungs using the phase value of the pixels from the LGXP process. The normal region growing technique and LGXP based region growing technique are then compared. By comparing both the techniques, it is checked whether the same pixel is the default.
During this process, if the difference between the adjacent pixel and the default pixel get the value as less than the threshold value on both the techniques separately, that adjacent pixel is included for region growing, or else that adjacent pixel needs to be excluded. However, the adjacent pixel and the default pixel which have been chosen to compare should be same on both the techniques.

B. Local Gabor XOR Pattern (LGXP)

The fundamental idea of the technique is to ease the sensitivity of Gabor phase to the differing positions. It should be confirmed whether the same local feature must be determined in a lenient way. Specifically, if two phases belong to the same interval (for instance: 00, 900), they are believed to have similar local features or else they reflect different local features. In this section, the LGXP descriptor is presented.

![Figure 3.9 Example of LGXP Method Where the Phase is Quantized into Four Ranges](image)

The Figure 3.9 shows an instance for the LGXP encoding method where the phase is quantized into four ranges. In LGXP technique, the phases are first quantized into disparate ranges and the LGXP operator is applied to the quantized pixels of the central pixel and each of its neighboring pixels and eventually the result of the binary labels are concatenated together as a local pattern of the central pixel. In the Figure 3.9, (a) is the matrix with the initial phase of the pixels after applying the Gabor filter (b) is the result after quantization (c) is the result after XOR comparison with the center quantized value. From the matrix got after XOR comparison, the binary of 01011101
can be deduced and its equivalent decimal value is 93. The pattern of LGXP in binary and the decimal form is as in Equation (3.6),

\[ LGX(P_{\mu\nu}, P_c) = [LGX(P_{\mu\nu}^N, LGX(P_{\mu\nu}^{N-1}), \ldots, LGX(P_{\mu\nu}^1)]_{\text{binary}} \]

\[ = \left( \sum_{i=1}^{N} 2^{i-1} \cdot LGX(P_{\mu\nu}^i) \right)_{\text{decimal}} \quad (3.6) \]

Where, \( P_c \) denotes the central pixel in the Gabor phase map with scale \( \mu \) and orientation \( \nu \), \( N \) is the size of the neighborhood and \( LGX(P_{\mu\nu}^i) \) (\( i = 1, 2, \ldots, N \)) denotes the pattern calculated between \( P_c \) and its neighbor \( P_i \), which is computed as in Equation (3.7),

\[ LGX(P_{\mu\nu}^i) = q\left( \Phi(P_{\mu\nu}(P_c)) \right) \oplus q\left( \Phi(P_{\mu\nu}(P_i)) \right), \quad i = 1, 2, \ldots, N \quad (3.7) \]

Where \( \Phi(P_{\mu\nu}) \) denotes the phase, \( \oplus \) denotes the LXP operator, which is based on XOR operator, \( q \) denotes the quantization operator which calculates the quantized code of the phase according to the number of phase ranges defined in Equations (3.8 and 3.9),

\[ a \oplus b = \begin{cases} 0, & \text{if } a = b \\ 1, & \text{else} \end{cases} \quad (3.8) \]

\[ q\left( \Phi(P_{\mu\nu}(\cdot)) \right) = i, \quad \text{if } \frac{360^\circ i}{e} \leq \Phi(P_{\mu\nu}(\cdot)) < \frac{360^\circ (i + 1)}{e}, \quad i = 0, 1, \ldots, b - 1 \quad (3.9) \]

Where, \( e \) denotes the number of phase ranges. With the pattern explained above, one pattern map is computed for each Gabor kernel. Thereafter, each pattern map is split into \( m \) non overlapping sub blocks and the histograms of all the sub blocks of scales and the orientations are concatenated to form the proposed LGXP descriptor of the input face image given in Equation (3.10),
\[ H = \left[ H_{k,\gamma(v_0, \ldots, v_0)}, \ldots, H_{k,\gamma(v_0, \ldots, v_0), 1, \ldots, 1, \gamma(v_0, \ldots, v_0)} \right] \] \hspace{1cm} (3.10)

Where \( H_{i,j,v} \) \((i = 1, 2, \ldots, m)\) denotes the histogram of the \(i^{th}\) sub block of the LGXP map with scale \(v\) and orientation.

C. Cavity Segmentation

After the lung segmentation, the cavities in the lung are identified. The cavities present in the lung region are essential to identify the COPD affected lung. To identify the cavity in the lung, an adaptive threshold value is to be set. The threshold value is chosen by calculating the pixel range in the lung region and dividing that pixel range by two. After that, the threshold is compared with all the pixels. While comparing the pixels to the threshold value, if the pixel value is greater than the threshold value, it would be then the cavity region and if the pixel value is less than the threshold values it would be then the lung region. The Figure 3.10 shows the block diagram for segmenting the cavity region from the lung region.

![Figure 3.10 Block Diagram of Cavity Segmentation](image-url)
D. Feature Extraction

After finding the regions, some features are extracted to diagnose the disease in the lung. To discover the disease in the lung, the extracted feature have to be fed into the classifier, because the extracted features will give vital information about the region which is used to train the classifier. In the present research, an ELM classifier is used for feature extraction. The features that need to be extracted are the number of cavities in the lung region, minimum area of the cavity region, maximum area of the cavity region, total number of pixels in each cavity, the maximum repeated pixel intensity in the cavity region and maximum repeated pixel in the lung region to find the total number of cavities in the lung region. As the normal lung would also have some cavities present in its region, to distinguish the normal lung image and the COPD affected lung the total numbers of cavities present in the lung region must be found out and the result given to the ELM classifier.

ELM meant for SLFNs will randomly select the input weights and analytically determine the output weights. This algorithm tends to afford the best generalization performance at an extremely fast learning speed. The structure of ELM network is shown in Figure 3.11. ELM contains an input layer, hidden layer and an output layer.

![Figure 3.11 Structure of ELM](image)
The ELM has several interesting and significant features different from the traditional popular learning algorithms for Feed forward Neural Networks as shown Figure 3.12. These include the following,

- The learning speed of ELM is quicker than the other classifier. The learning process of ELM can be performed in seconds or less than a second for several applications.

- The ELM will attain the results directly without any difficulties. The ELM learning algorithm is simpler than the other learning.

Figure 3.12 Block Diagram of ELM
ELM training Algorithm, uses a finite number of input-output samples for training. If there are N samples considered in the lung cavity region as $\mathbf{X}_i = [X_{i1}, X_{i2}, \ldots, X_{in}]^T \in \mathbb{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{in}]^T \in \mathbb{R}^n$ are the cavity and the area of the cavity region, the standard SLFN with N hidden neurons and activation function $g(x)$ is then defined as in Equation (3.11),

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

Where $w_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ represents the weight vector that links the ith hidden neuron and the input neurons. $\beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{in}]^T$ represents weight vector that links the ith neuron and the output neurons, and $b_i$ represents the threshold of the ith hidden neuron. The “.” in $w_i \cdot x_j$ indicates the inner product of $w_i$ and $x_j$. The SLFN tries to reduce the difference between $o_j$ and $t_j$. More in a matrix format as $H \beta = T$,

Where

$$H(a_1, \ldots, a_n, b_1, \ldots, b_N, X_1, \ldots, X_N) = \begin{bmatrix} g(a_1, b_1, X_1) & \cdots & g(a_N, b_N, X_1) \\
\vdots & \ddots & \vdots \\
g(a_1, b_1, X_N) & \cdots & g(a_N, b_N, X_N) \end{bmatrix}_{N \times N}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{bmatrix}_{N \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$$

The result reduces the norm of this Least squares equation as defined in Equation (3.13),

$$\hat{\beta} = H^+T \quad (3.13)$$
Where $H^+$ the Moore-Penrose is generalized inverse (Karpagacheli et al 2011) of the hidden layer output matrix $H$. The ELM algorithm which consists of only three steps, can then be summarized as,

Step 1: Given a training set

$$\mathcal{N} = \{(X_i, t_i) | X_i \in \mathbb{R}^m, i = 1, \ldots, N\}$$ activation function $g(x)$, and hidden number node $N$,

1) Assign random hidden nodes by randomly generating parameters $(a_i, b_i)$ according to any continuous sampling distribution, $i = 1, \ldots, N$

2) Calculate the hidden layer output matrix $H$.

3) Calculate the output weight $\hat{\beta} \hat{\beta} = H^+ T$

The maximum repeated pixel intensity in the cavity regions of a lung is then found out. To discover the maximum repeated pixel, the intensities of all the pixels in each cavity of a lung are found out by implementing histogram and thereafter all the pixels of every cavity are compared with each other. After discovering the maximum repeated pixel in the cavities of a lung, the result must be given to the classifier. Similarly, the maximum repeated pixel in the whole lung region is to be found and the result given to the classifier. The classifier detects that the lung is affected by COPD or not by comparing all the features.

**Training and Testing using ELM Classifier**

Some of the data features are to be taken to identify the normal lung region and COPD affected lung. By this, the classifier is trained. The data features will then train the classifier and the classifier will find whether the
given CT scan image is normal or abnormal. The data features which have been chosen for training the ELM classifier are the number of cavities in the lung region, maximum area of cavity in the lung region, minimum area of cavity in the lung region, total number of pixels in each cavity, the maximum repeated pixel in the cavity regions together and the maximum repeated pixel in the lung region. After computing all the data features, the values have to be given to the classifier. For instance, by choosing three normal CT scan images and three abnormal CT scan images, all the six data features need to be calculated separately for all the CT scan images chosen. After calculating all the six data features for every chosen CT scan images, the result has to be given to the ELM classifier. Using those results, the classifier is trained to identify the normal and abnormal lung from the given CT scan image. After the training of the ELM classifier, a new CT scan image is trained to find whether it has COPD or not. Thereafter, the six data features such as the number of cavities in the lung region, maximum area of the cavity region, minimum area of the cavity region, total number of pixels in each cavity, maximum repeated pixel in the cavity region and maximum repeated pixel in the lung region are to be computed for the new CT scan image. The computed values of all the six data features are given to the ELM classifier. The values of all the six data features are then compared by the ELM Classifier with the stored values of the normal and abnormal CT scan images. As during training all the six data features of the five normal CT scan images and five abnormal CT scan images have been stored, the ELM classifier will identify after comparison whether the given CT scan image comes under the normal category or abnormal category.

3.6 ADVANTAGES OF PROPOSED ELM

- ELM is a simple tuning-free three-step algorithm.
- The learning speed of ELM is extremely fast.
Unlike the traditional classic gradient-based learning algorithms which only work for differentiable activation functions.

Unlike the traditional classic gradient-based learning algorithms facing several issues like local minima, improper learning rate and overfitting, the ELM tends to reach the solutions straightforward without such trivial issues.

The ELM learning algorithm looks much simpler than many learning algorithms such as neural networks and support vector machines.

3.7 RESULTS AND DISCUSSION

The experiment has been done for 50 COPD lung images in the MATLAB environment.

MATLAB (Matrix LABoratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. The core MATLAB package comes with several rudimentary functions that can be used to load, save, and perform custom functions on images. However, it is often necessary to perform more complicated operations on images.
The Figure 3.13 shows the normal and abnormal lung images.

![Figure 3.13 Sample Images of the Normal and Abnormal Lungs Images](image)

(a) Normal Image  
(b) Abnormal Image

The sample images are taken and the images are filtered using Laplacian Gaussian filter. The filtering technique is used to remove the noise and it improves the quality of the images as shown in the Figure 3.14.

![Figure 3.14 Sample Image of Lungs After Filtering Process](image)

(a) Normal Image After Filtering Process  
(b) Abnormal Image After Filtering Process

After applying the filtering technique the sample images are given to the process of lung segmentation. The lung segmentation process segments only the lung region from the sample CT scan images. The Figure 3.15 shows a sample image of the segmented lungs with COPD and without COPD.
After the lung is segmented from the sample images, the cavities have to be segmented from the lung region. Using the cavities in the lung region, whether the lung is COPD affected or not has to be identified. The Figure 3.16 shows a sample image of the segmented cavities and segmented cavities with CT scan image for the COPD affected lung.

3.8 PERFORMANCE ANALYSIS USING EVALUATION METRICS

The evaluation of the COPD identification of the images is carried out using the following metrics shown in Equations (3.14, 3.15 and 3.16),
Where,

True Positive is TP, True Negative is TN, False Negative is FN, False Positive is FP.

Sensitivity is the proportion of true positives that are correctly identified by a diagnostic test. It shows how good the test is at detecting a disease.

Specificity is the proportion of true negatives that are correctly identified by a diagnostic test. It shows how good the test is at rejecting a disease.

Accuracy is the proportion of true results, either true positive or true negative, in a population. It measures the degree of veracity of a diagnostic test on a condition.

Table 3.1 Comparative Analysis of Existing SVM Technique with ELM Technique

<table>
<thead>
<tr>
<th>Techniques</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing SVM</td>
<td>19</td>
<td>18</td>
<td>7</td>
<td>6</td>
<td>76</td>
<td>72</td>
<td>74</td>
</tr>
<tr>
<td>Proposed ELM</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>80</td>
<td>80</td>
<td>80</td>
</tr>
</tbody>
</table>
Table 3.1 shows the comparison accuracy, specificity and sensitivity between the proposed ELM technique and the existing SVM technique. The Figure 3.17 shows that in terms of evaluation metrics the proposed technique gives better performance than the existing technique.

![Figure 3.17 Comparative Evaluation Metrics Analysis of Existing SVM and ELM Techniques](image)

3.9 SUMMARY

This chapter clearly discusses briefly about the proposed algorithm for finding the COPD disease in lungs. The process and the other general information about the proposed techniques are thoroughly explained in this chapter. This chapter discusses about the performance evaluation of the proposed ELM approach. The performance of this proposed ELM approach is compared with the existing SVM technique with certain parameters like sensitivity and accuracy values. It is observed from the experimental results that the proposed ELM technique provides the better results.