ABSTRACT

Lung cancer is one of the most killer diseases in the developing countries and the detection of the cancer at the early stage is a challenge. Analysis and cure of lung malignancy have been one of the greatest difficulties faced by humans over the most recent couple of decades. Early identification of tumour would facilitate in sparing a huge number of lives over the globe consistently.

The chance of survival at the advanced stage is less when compared to the treatment and lifestyle to survive cancer therapy when diagnosed at the early stage of the cancer. Manual analysis and diagnosis system can be greatly improved with the implementation of image processing techniques.

This project presents an approach which utilizes a Convolutional Neural Network (CNN) to classify the tumours found in lung as malignant or benign. The accuracy obtained by means of CNN is 96%, which is more efficient when compared to accuracy obtained by the traditional neural network systems.

Key Words: Lung cancer, Computed Tomography (CT) Scan, Chest CT image, Neural Network, Deep Learning, Convolutional Neural Network.

CHAPTER 1
INTRODUCTION

Lung cancer is one of the most dreadful diseases in the developing countries and its mortality rate is 19.4% [1]. Early detection of lung tumour is done by using many imaging techniques such as Computed Tomography (CT), Sputum Cytology, Chest X-ray and Magnetic Resonance Imaging (MRI). Detection means classifying tumour in two classes
(i) Non-cancerous tumour (benign) and (ii) Cancerous tumour (malignant)[2].

Manual analysis and diagnosis system can be greatly improved with the implementation of image processing techniques.

1.1 IMAGE:

An image may be defined as a two-dimensional function f(x, y), where x & y are spatial coordinates, & the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image.

1.1.1 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

f (0, 0) f (0, 1) ……….. f (0,N-1)
f (1,0) f (1, 1) ………… f (1, N-1)
f (xylem) = f (M-1, 0) f(M-1, 1).............f (M-1, N-1)

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

f (1, 1) f (1, 2) f (1, N)

f (2, 1) f (2, 2) f (2, N)

. . f = f (M, 1) f(M, 2) .... f (M, N)

Where f (1, 1) = f (0, 0) (note the use of a moonscape font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation f (p, q) denotes the element located in row p and the column q. For example f (6, 2) is the element in the sixth row and second column is the element in the sixth row and second column of the matrix f. typically we use the letters M and N respectively to denote the number of rows and columns A 1xN matrix is called a row vector whereas an Mx1 matrix is called a column vector. A 1x1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A, a, RGB, real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using monoscope characters. We use conventional Roman, italic notation such as f(x, y), for mathematical expressions.

1.2 DIGITAL IMAGE PROCESSING:

The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low- level process is characterized by the fact that both its inputs & outputs are images. Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images.

Finally higher- level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision. Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

CHAPTER 2

COMPUTED TOMOGRAPHY (CT) IMAGES IN CANCER DETECTION

Lung cancer is one of the most dangerous forms of cancer because it claims more than a million precious lives every year. So, lung nodule detection in chest Computed Tomography (CT) images becomes very necessary in the present clinical world. Thus the Computer Aided Diagnosis (CAD) system is very essential for early detection of lung cancer.

CT uses special x-ray equipment to get image data from various angles around the human body, and then utilizes computer processing of the information to demonstrate a cross-section of tissues and organs.

CT imaging is very useful as it can display various types of tissues and organs with high clarity, when an intravenous contrast (x-ray dye) is utilized. Moreover, tissues like kidney or gallstones can be accurately detected with CT, and abnormal fluid or enlarged lymph nodes in the abdomen or pelvis can also be identified with great accuracy. Some organs like stomach are not that much accurately assessed by the CT, but it is used to indirectly diagnose these organs by detecting abnormalities in the adjacent soft tissues. CT Colonography (Virtual Colonoscopy) is a novel technique which allows primary assessment of the distended colon to detect polyps, the precursor to colon cancer, and is very effective in screening for this disease.

CT is becoming one of the most popular and effective methods for diagnosing many diseases including diverticulitis and appendicitis, and for visualizing the liver, spleen, pancreas and kidneys as it is a non-invasive procedure that provides detailed, cross-sectional views of all types of tissue. CT can quickly identify the source of pain in cases of acute abdominal distress. The speed, ease and accuracy of a CT examination can minimize the risk of serious complications, when pain is due to infection and inflammation.

CT is widely used for diagnosing various types of cancer, including kidney and pancreatic cancer. CT is very effective as the image gives full information about the presence of a tumor and to measure its size, precise location, and the extent of the tumor involvement with other nearby tissue (staging the tumor). CT assessments of the lower Gastro Intestinal (GI) tract are very much useful to plan and properly administer radiation treatment for tumors. CT also plays a crucial role in abdominal trauma assessment, since it is very sensitive at picking up bleeding within and around the solid organs. CT is also very effective in the detection, diagnosis and treatment of vascular disorders through a new approach called CT Angiography.

2.1 TYPES OF CT SCANS:

i. Sequential CT:

By scanning a crosswise section of the body from various angular positions, a cross-sectional image is obtained while the tube and detector rotate 360° around the patient with stable table. The image is reconstructed from the resulting projection data.

The data attained from the various angular positions are no longer constant if the patient moves during the acquisition. The image is degraded by motion artifacts and may be of limited diagnostic value. The tomographic approach is appropriate only to a limited degree for the diagnosis of anatomical regions with automatism functions (i.e. the involuntary functioning of an organic process, especially muscular, without apparent neural stimulation) such as the heart or the lung.

ii. Spiral CT:

Spiral CT is also called as “volume scanning”. It is clearly different from usual CT and the tomographic technique is used in Spiral CT. Spiral CT uses a different scanning principle. The patient on the table is moved constantly through the scan field in the z direction while the gantry performs multiple 360° rotations in the same direction. The X-ray draws a spiral rotation around the body and constructs a data volume. This volume is created from a multitude of three-dimensional picture components, i.e. voxels. The movement of the table in the ‘z’ direction will usually produce inconsistent sets of data, which causes every image reconstructed directly from a volume data set to be degraded by artifacts. Software applications facilitate the use of spiral CT even for regions which are subject to involuntary movements.

2.2 PRODUCTION OF CT IMAGE:

A thin, needle-like beam linearly scans the object. A sort of shadow image is produced which is called as “attenuation profile” or “projection”. It is recorded by the detector and the image processor. The tube and the detector are further rotated by a small angle. A second shadow image is produced by linearly scanning the object from another direction. This process is repeated until the object has been scanned for a 180° rotation.

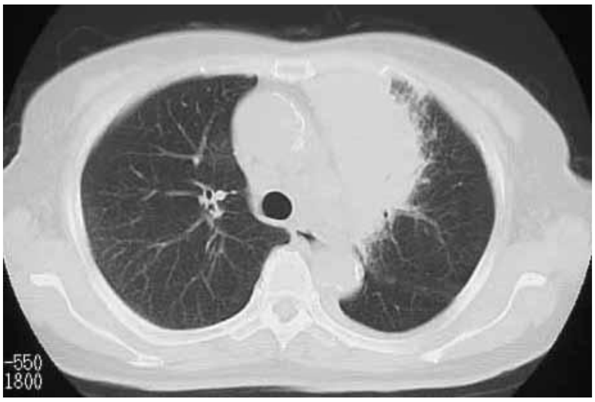


Figure 2.1 An Example of CT image

2.3 BENEFITS OF CT SCAN:

- CT scan can easily detect the causes of abdominal pain with very high accuracy, enabling faster treatment

- CT scanning provides thorough views of many types of tissues, including the lungs, bones, soft tissues and blood vessels.

- CT scanning is painless, non-invasive and accurate.

- CT examinations are very simple and rapid.

- Detection and diagnostics done with CT eliminates the need for invasive exploratory surgery and surgical biopsy.

- Scanning done using CT identifies both normal and abnormal structures, making it a useful tool to guide radiotherapy, needle biopsies etc.

- CT is cost-effective imaging tool for a wide range of clinical problems.

Computed Tomography (CT) is a dominant tool which allows very quick creation of x-ray images of the body with high-resolution cross- sectional imaging. The quick, detailed result has made CT very valuable, especially in the emergency department (ED). High-quality, cross-sectional images are available in a quick time that helps to define the medical status of the patient.

CHAPTER 3
USING DEEP LEARNING FOR LUNG CANCER DETECTION

Deep Learning is achieved using “Neural Networks”. Neural networks have been effectively applied across a range of problem domains like finance, medicine, engineering, geology, physics and biology. From a statistical viewpoint, neural networks are interesting because of their potential use in prediction and classification problems.

Deep learning composed of several layers of nonlinear nodes, combine input data with a set of weights so that assigning significance to inputs for the corresponding task the algorithm is attempting to learn in supervised and/or unsupervised behavior. The sum of product of these input and weights is passed through activation function of nodes. [10][12].The output of each layer’s is fed simultaneously as input to the subsequent layer starting from input layer [8]. Learning can be performed in multiple levels of representations correspond to various levels of abstraction.

3.1 NEURAL NETWORKS:

Neural Networks (NNs) have been widely used for variety of applications. Some of them are classification problems, recognizing speech, and predicting the secondary structure of globular proteins. In time-series applications, NNs have been used in predicting stock market performance. These issues are solved through classical statistical methods, such as discriminant analysis, logistic regression, Baye’s analysis, multiple regressions etc.

3.1.1 CONVOLUTION NEURAL NETWORKS (CNN):

A CNN is type of a DNN consists of multiple hidden layers such as Convolutional layer, RELU layer. Pooling layer and fully connected a normalized layer. CNN shares weights in the convolutional layer reducing the memory footprint and increases the performance of the network. The important features of CNN lie with the 3D volumes of neurons, local connectivity and shared weights. A feature map is produced by convolution layer through convolution of different sub regions of the input image with a learned kernel. Then, anon-linear activation function is applied through ReLu layer to improve the convergence properties when the error is low. In pooling layer, a region of the image/feature map is chosen and the pixel with maximum value among them or average values is chosen as the representative pixel so that a 2x2 or 3x3 grid will be reduced to a single scalar value. This results a large reduction in the sample size. Sometimes, traditional Fully-Connected (FC) layer will be used in conjunction with the convolutional layers towards the output stage.

In CNN architecture, usually convolution layer and pool layer are used in some combination. The pooling layer usually carries out two types of operations viz. max pooling and means pooling. In mean pooling, the average neighbourhood is calculated within the feature points and in max pooling it is calculated within a maximum of feature points. Mean pooling reduces the error caused by the neighbourhood size limitation and retains background information. Max pooling reduces the convolution layer parameter estimated error caused by the mean deviation and hence retains more texture information.

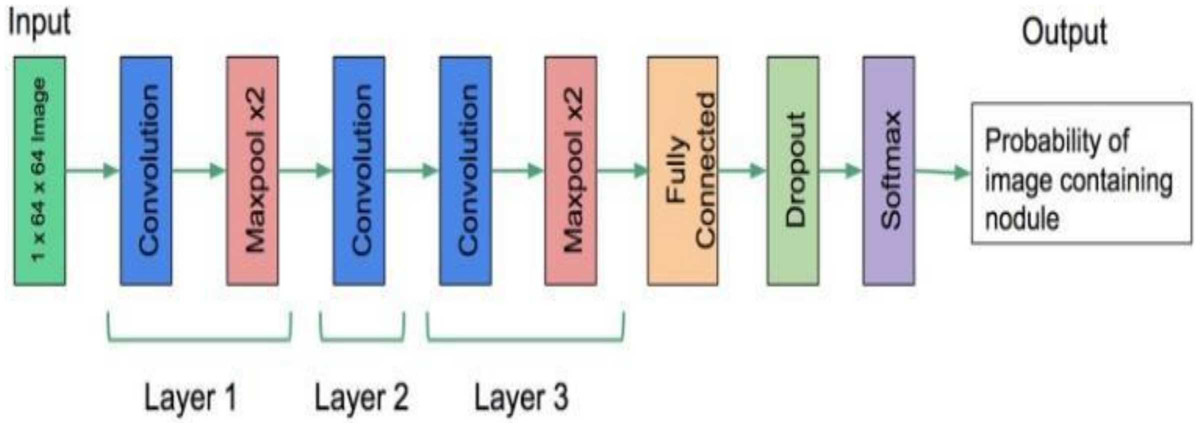


Fig 3.1: Architecture of CNN

3.1.2 ARTIFICIAL NEURAL NETWORKS:

An Artificial Neural Network (ANN) is a technique based on emulation of biological neural system. ANN is composed of a group of interconnected artificial neurons and processes information using a connectionist technique to calculation. It is an adaptive system that transforms its structure depending on external or internal information that flows through the network during the learning phase. Modern NNs are non-linear statistical data modeling approaches which are mainly used to model complex relationships between inputs and outputs or to determine patterns in data. Figure 1.8 represents the complete architecture of artificial neural networks.

Artificial neural networks (ANNs) are non-linear data driven self adaptive types. ANN is a dominant tool for modeling, particularly when the underlying data relationship is unknown. Correlated patterns between input data sets and corresponding target values can be recognized and studied by ANN. After training, the outcomes of new independent input data can be predicted by ANN. ANNs reproduce the learning process of the human brain and can process issues involving non-linear and complex data even if the data are vague. Thus they are very much suited for the modeling of even agricultural data which are known to be complex and often non-linear.

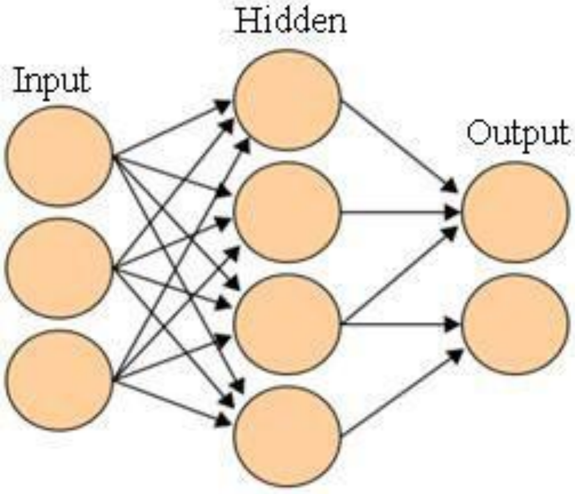


Figure 3.2: Architecture of ANN

A key feature of these networks is their adaptive nature. This feature makes such computational techniques very attractive in application domains. Networks are “neural” which may be inspired by neuroscience but not necessarily as they are realistic techniques of biological neural or cognitive fact. Most of the networks are more closely related to traditional mathematical and statistical models such as non-parametric pattern classifiers and statistical regression models than they are to neurobiology models.

The main advantage of ANNs is the capability to be used as an arbitrary function approximation approach which learns from observed data. But, a relatively good understanding of the underlying theory is necessary as using ANN is not straightforward.

Moreover, the ANNs are flexible and adaptive, learning and adjusting with each different internal or external stimulus. Artificial Neural Networks are mostly applied in sequence and pattern recognition systems, Data processing, Robotics and Modeling. There are various types of Neural Networks like feed forward Neural Network, Radial Basis Function (RBF), Kohonen Self-organizing Network, Recurrent Neural Network etc.

Learning and generalizing are the most important abilities of Artificial Neural Networks. ANN obtains knowledge from their surroundings by adapting to internal and external parameters. It learns from previous examples and adapts to situations based on its observations and results. It generalizes knowledge to provide appropriate responses to unknown situations. They have the capability of solving complex problems that are difficult to manage by approximation.

A linear or a non-linear response can be produced by a computational neuron. A non-linear ANN consists of interconnection of non- linear neurons. The inputs of the Non-linear systems are not proportional to the outputs. This function makes the network to obtain knowledge through learning in an effective way. This is a unique advantage of ANN over other traditional linear networks.

ANN has the ability of greater fault tolerance than a traditional network. The network has the capability of regenerating a fault in any of its components without any loss of stored data. It uses results of the past to reassemble the functioning of a damaged node.

ANN works on the concept of abstract learning. There are three learning paradigms available to equip the network for adaptive learning. They are:

i. Supervised learning: It generates a function that maps inputs to desired outputs.

ii. Unsupervised learning: This models a set of inputs, like clustering.

iii. Reinforcement learning: It learns how to act given an observation of the world.

Neuron networks are trained through specialized algorithms like Non-parametric methods, Expectation Maximization (EM), Simulated Annealing and evolutionary methods. An evolutionary algorithm (EA) is defined as a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. Simulated annealing (SA) is defined as a general probabilistic metaheuristic for the global optimization crisis of applied mathematics. An expectation-maximization is a method for finding maximum likelihood estimates of parameters in statistical models, where the approach corresponds to unobserved latent variables. The neurons of an ANN are very much flexible to various input signal patterns and can adapt to a variety of unknown situations. ANN updates itself by learning from the past and replacing the older techniques and procedures.

3.3 Applications of ANN:

- Pattern Classification
- Clustering/Categorization
- Function approximation
- Prediction/Forecasting
- Optimization
- Content-addressable Memory
- Control

CHAPTER 4
LITERATURE SURVEY

There are several techniques available in the literature for the detection of cancer. Many researchers have contributed their ideas in the detection of cancer.

This chapter mainly discusses about the existing cancer detection techniques available in the literature. Several domains and concepts are used in the detection of cancer. The main domains used in this detection technique include neural networks, image processing, nanotechnology etc.

4.1 CAD SYSTEM IN CANCER DETECTION:

Different computer aided diagnosis systems exist for detection of lung cancer. Some of those techniques are discussed in this section.

Yamomoto et al (1996) proposed image processing for Computer-Aided Diagnosis (CAD) of lung cancer by CT (LSCT). LSCT is the newly built mobile-type CT scanner mainly for the purpose of mass screening of lung cancer. In this new LSCT technique, one main complexity is the raise in the image information to around 30 slices per person from 1 X-ray film. In order to ignore this problem, the author attempts to reduce the image information considerably to be displayed for the doctor with the help of image processing techniques.

YenyYim et al (2005) stated about Hybrid lung segmentation in chest CT images for computer-aided diagnosis. The proposed system consists of three phases. In the first phase, lungs and airways are separated by an inverse seeded region growing and linked component labeling. In the second phase, trachea and large airways are eliminated from the lungs by three-dimensional region growing. In the final phase, exact lung region borders are obtained by subtracting the outcome of the second phase from that of the first phase.

Penedo et al (1998) put forth a computer-aided diagnosis scheme that depends on two-level artificial neural network (ANN) architecture. The initial artificial neural network performs the identification of suspicious regions in a low-resolution image. The input provided to the second artificial neural network is the curvature peaks calculated for all pixels in each suspicious region. This is found out from the reality that small tumors possess and identifiable signature in curvature-peak feature space, where curvature is the local curvature of the image data when sighted as a relief map. The outcome of this network was thresholded at a particular level of importance to provide a positive identification.

Kanazawa et al (1996) described Computer aided diagnosis system for lung cancer based on helical CT images. This technique will reduce the time complexity and increase the diagnosis confidence. This method consists of an analysis stage and a diagnosis stage. In the analysis stage, the lung and pulmonary blood vessel regions are extracted and examined the features of these regions with the help of image processing methods. In the diagnosis stage, diagnostic rules were determined according to the features, and tumor regions are identified using these diagnostic rules.

Yamamoto et al (2000) explained Computer aided diagnosis system with functions to assist comparative reading for lung cancer based on helical CT image. This work provided a new and automatic system for an early detection of lung cancer based on a computer aided diagnosis system in which all the CT images are read. Additionally, the CAD system is provided with functions to automatically identify the suspicious regions from chest CT images, and to provide the comparative reading in retrospect. The slice matching algorithm is used in this approach for comparing every slice image of the current and previous CT scans, and an interface to show some features of the suspicious regions.

Cheran et al (2005) provided computer aided diagnosis for lung CT using artificial life models. This CAD system is based on several techniques such as 3D region growing, active contour and shape models, centre of maximal balls but it can be believed that at the center of this technique are the biological models of ant also called as artificial life models. In the starting stage, the 3D region growing algorithm is applied on the image for detecting the ribcage. After the ribcage is identified an active contour is used for building a confined area for the incoming ants that are set up to make clean and perfect restoration of the bronchial and vascular tree. Next the branches of the newly restored trees are verified to detect whether they comprise nodulesor not with the help of active shape models and also for determining if there are any nodules connected to the pleura of the lungs. The next phase is to remove the trees in order to provide a cleaner technique for localizing the nodules which is performed with the help of snakes and dot enhancement techniques.

A New CAD system for early diagnosis of lung nodules was proposed by El-Baz et al (2007). The growth rate is identified by calculating the volumetric difference in the identified lung nodules over time, so that it is essential to perfectly calculate the volume of the nodules to quantify their growth rate over time. The proposed CAD system consists of five main steps. They are:

1. Segmentation of lung region from CT images
2. Determining the lung nodules from segmented lung region
3. A non-rigid registration method to align two successive LDCT scans and to correct the motion artifacts caused by breathing and patient motion
4. Segmentation of the detected lung nodules
5. Quantification of the volumetric changes

Lin et al (2002) provided a neural fuzzy model to formulate the diagnosis rules for identifying the pulmonary nodules. Initially, a series of image processing methods like thresholding, morphology closing and labeling were performed to segment the lung area and region of interests are obtained. Next, three main features such as circularity, size of area, and mean brightness were obtained from region of interest and the nodules are detected with diagnosis rules that are formed with the help of neuro fuzzy model.

Armato et al (2001) developed a fully automated computerized technique for the identification of lung nodules in helical computed tomography scans of the thorax. This technique is based on two-dimensional and three-dimensional analysis of the image obtained during diagnostic CT scans. Lung segmentation was carried out on a section-by-section process to create a segmented lung volume within which further analysis is carried out. Multiple gray-level thresholds are supplied to the segmented lung volume for producing a series of thresholded lung volumes. An 18-point connectivity technique was implemented to detect contiguous three-dimensional structures within every thresholded lung volume, and those structures that satisfy a volume criterion are chosen as initial lung nodule candidates. Morphological and gray-level features were calculated for every nodule candidate. After a rule-based technique is used to decrease the number of nodule candidates that corresponds to non-nodules, the features of other candidates were combined through linear discriminant analysis.

Fiebich et al (2001) proposed an improved method for computer- assisted detection of pulmonary nodules in CT of the chest. All air outside the patient, soft tissue and bony structures are removed with the help of CAD technique. In the remaining lung field three-dimensional region identification is carried out and rule-based analysis is carried to identify possible lung nodules.

Armatur et al (1992) applied the Hopfield neural network for the multispectral unsupervised classification of MR images. Winner-take-all neurons were used to obtain a crisp classification map with the help of proton density-weighted and T2-weighted images in the head. Yamamoto et al., (1996) proposed a new technique named Quoit filter (Q-filter) to obtain the isolated but low amplitude shadow situated in the background which has very high amplitude variation. This filter was applied to identify the cancer candidate shadow automatically in the CT cross sections of lung areas, intending to decrease considerably the number of cross sections to be diagnosed by the doctor.

4.2 IMAGE PROCESSING IN CANCER DETECTION:

An image processing approach using Massive Training Artificial Neural Networks (MTANN) was proposed by Kenji Suzuki et al (2006). The technique is very much useful for the radiologists to identify these nodules if the lung nodule overlaps with the ribs in chest radiographs. MTANN is a non-linear filter that can be trained by use of input chest radiographs and the equivalent “teaching” images. A linear-output back-propagation (BP) algorithm is used that was derived for the linear-output multilayer ANN approach in order to train the MTANN. The dual-energy subtraction is a technique used in this work for separating bones from soft tissues in chest radiographs by using the energy dependence of the x-ray attenuation by different materials.

Kazunori Okada et al (2005) proposed a statistical estimation and verification for illustrating the ellipsoidal geometrical structure of pulmonary nodules in the Multi-slice X-ray computed tomography (CT) images. A multi- scale joint segmentation and model fitting solution is proposed in this approach. This technique extends the robust mean shift-based analysis to the linear scale-space theory. A quasi-real-time three-dimensional nodule illustration scheme was developed using this approach and was validated with two clinical data sets of thin-section chest CT images. This proposed approach has three different but successive stages namely model estimation, model verification and volumetric measurements.

Ingrid Sluimer et al (2005) proposed a Segmentation-by- registration approach. A scan with normal lungs is elastically registered to a scan containing pathology in this approach. Segmentation-by-registration approach makes use of an elastic registration of inclusive scans using mutual information as a similarity measure. The performance of the four segmentation approaches namely Refined Segmentation-by-Registration,Segmentation by Rule-Based Region growing, Segmentation by Interactive Region growing, and Segmentation by Voxel Classification were compared. From the results of the performance comparison, it is clear that refined registration approach enjoys an additional benefit since it is independent of a pathological (hand-segmented) training data.

A genetic algorithm for segmentation of medical images was proposed by Payel Ghosh et al (2006). The author proposed a genetic algorithm for automating the segmentation of the prostate on two-dimensional slices of pelvic Computed Tomography (CT) images. In this technique the segmenting curve is represented using a level set function, which is evolved using a Genetic Algorithm (GA). Shape and textural priors obtained from manually segmented images are used to limit the development of the segmenting curve over successive generations.

Some of the existing medical image segmentation approaches were reviewed by the author. Also they compared the performance of the approach with a simple texture extraction algorithm (Laws texture measures) and GA-based segmentation tool called GENIE. The primary tests were conducted on a small population of segmenting contours. It is observed that, a significant result by converging on the prostate region was obtained. The future enhancements can be achieved by incorporating spatial relationships between anatomical landmarks, and extending the approach to three dimensions.

A novel approach for lung nodule detection was described by Antonelli et al (2005). A CAD approach was described for automated detection of pulmonary nodules in computed-tomography (CT) images. Combinations of image processing approaches were used for extraction of pulmonary parenchyma. A region growing approach based on 3D geometric features was applied to detect nodules after the extraction of pulmonary parenchyma. From the experimental observation, it was noted that implementation of this nodule detection method detects all malignant nodules effectively with a very low false-positive detection rate.

Xujiong Ye et al (2009) proposed a new CT lung nodule CAD approach. The technique can be applied for identifying both solid nodules and ground-glass opacity (GGO) nodules. The first step in this approach is to segment the lung region from the CT data using a fuzzy thresholding approach. The next step is the computation of the volumetric shape index map and the “dot” map. The former mentioned map is based on local Gaussian and mean curvatures, and the latter one is based on the Eigen values of a Hessian matrix. They are computed for each Voxel within the lungs to improve objects of a specific shape with high spherical elements. The combination of the shape index and “dot” features offers a good structure descriptor for the initial nodule candidate generation. This approach has several benefits like high detection rate, fast computation, and applicability to different imaging conditions and nodule types makes it reliable for clinical applications.

4.3 RADAR CANCER DETECTION:

Munawar et al (2008) proposed an initial investigation for breast cancer detection using a special mode of bistatic radar system known as forward scattering radar (FSR). The proposed approach analyzes the Doppler frequency in the received signal scattered from the tumor for cancer detection and localization. Three systems of architectures were examined which are determined by the mechanical movement of transmitter or receiver or both. This approach also describes an initial simulated result by using CST Microwave Studio as a feasibility study of utilizing FSR for breast cancer detection. It is shown that a cancer can be predicted by investigating the unique character of Radar Cross Section (RCS) for breast tissue and tumor of FSR. Electromagnetic sample including fatty tissue and a tumor were simulated to obtain RCS parameter and analyzed as well as compared with whose fatty tissue without cancerous lesion to pinpoint the presence of tumor from its FSR signature. From the experimental results, a significant difference was observed between these two models in FS RCS.

4.4 NANO TECHNOLOGY IN CANCER DETECTION:

Various research and developments (Young-Eun Choi et al 2010) in the area of nanotechnology have been conducted and many nanomaterials are employed for cancer detection at early stages. Nano materials have exclusive properties like physical, optical and electrical and are very effective in sensing. Quantum dots, gold nanoparticles, magnetic nanoparticles, carbon nanotubes, gold nanowires have been developed to lower the detection limit of cancer biomarkers. Proteins, antibody fragments, DNA fragments, and RNA fragments are the base of cancer biomarkers and used as targets in cancer detection and monitoring. It is highly anticipated that it is possible to detect cancer at a very early stage, providing a much higher chance of treatment.

Due to the widespread occurrence, high death rate, and recurrence after treatment of the cancer, Cancer diagnosis and treatment have been of great importance in recent times. According to the National Vital Statistics Reports, it is indicated that cancer is wide-spread among all races and communities. Lung cancer, breast cancer and prostate cancer cause many deaths. Cancer is considered deadly due to recurrences, as though treatable, tumors can return after a period of time, even after chemotherapy, surgery, or radiotherapy.

Biomarkers and nanotechnology are the two important fields in the development of dominant diagnostic approaches which are extensively studied. A biomarker is a pointer of a biological state of disease. It is the feature of a particular state and thus can be used as a marker for a target disease. These biomarkers can be applied to learn cellular processes, and monitor or recognize disruption or alterations in the cellular processes of cancer cells. A biomarker can be a protein, a fragment of a protein, DNA, or RNA. Biomarkers, especially cancer biomarkers, are an indication of cancer and the presence of the particular cancer can be verified with the help of those biomarkers. Alongside the development of proteomic technologies, many protein biomarkers have been discovered for many types of cancer. As well, with DNA methylation analysis researchers have also been able to discover DNA biomarkers for some of the widely spread cancers.

The field of Nanotechnology has seen tremendous development recently and the properties of nanomaterials are being widely studied and many attempts are made to fabricate suitable nanomaterials. The unique optical, magnetic, mechanical, chemical and physical properties of the nanomaterials are used for more sensitive and precise biomarker detection. Nanomaterials that are applied to sensing cancer biomarkers vary from gold nanoparticles, quantum dots, magnetic nanoparticles, carbon nanotubes and nanowires (Ferrari M et al 2005, Zhang et al 2009).

Nanomaterials have unique features that are attractive, and can be applied to biosensing. The development of various nanomaterials and nanotechnology has enabled detection of cancer biomarkers with great precision and sensitivity that could not be achieved before. The low detection limit obtained by nanotechnology is expected to contribute immensely to the early detection and accurate prognosis of cancers.

4.5 MACHINE LEARNING APPROACHES IN CANCER DETECTION:

A novel technique for the detection of clustered microcalcifications (MCs) in mammograms is proposed by Xinsheng Zhang et al (2009). MCs are the vital early sign of breast cancer. The accurate detection of MCs is a vital factor in CAD scheme. In order to enhance the performance of detection, a Bagging and Boosting based twin support vector machine (BB-TWSVM) to identify MCs was proposed. The approach consists of three modules namely the image pro-processing, the feature extraction component and the BB-TWSVM module. The ground truth of MCs in mammograms is taken to be known as 'a priori'. Firstly, a simple artifact removal filter and a well designed high-pass filter are used for the preprocessing of each MC. Then the combined image feature extractors are used to extract 164 image features. In the combined image features space, the MCs detection procedure is devised as a supervised learning and classification problem, and the trained BB-TWSVM is used as a classifier to determine for the presence or absence of MCs. The experimental observation clearly shows the significance of the technique for CAD of breast cancer.

Two vital problems in mammogram analysis for breast cancer in MR-images were described by Behnamghader et al (2007). The first category is between normal and abnormal cases and then, classification between benign and malignant in cancerous cases. This proposed technique extracts textural and statistical descriptive features that are fed to a learning engine based on the use of Support Vector Machine (SVM) learning framework to categorize them. The experimental observations provide excellent accuracy in both classification problems, that proves the appropriate combination of the features and selecting powerful classifier i.e. SVM leads us to a brilliant outcome.

Microcalcification (MC) clusters in mammograms can be a pointer for breast cancer in women. El-Naqa et al (2002) proposed the use of SVM learning for automated detection of MCs in digitized mammograms. In this framework, MC detection is devised as a supervised-learning issue and the approach of SVM is employed to develop the detection approach. The technique was evaluated using a database of 76 mammograms containing 1120 MCs. To estimate detection performance, Free-Response Receiver Operating Characteristic (FROC) curves are used. From the experimental observation, it was clearly noted that the proposed SVM framework offers the best performance when compared with the existing approaches.

Machine learning is a branch of Artificial Intelligence (AI) that uses a diversity of statistical, probabilistic and optimization approaches that allows computers to "learn" from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. Therefore, machine learning is often used in cancer diagnosis and detection. In the research work by Osareh et al (2010), SVM, K-nearest neighbors and probabilistic neural network classifiers were combined with signal-to-noise ratio feature ranking, sequential forward selection-based feature selection and principal component analysis feature extraction to distinguish between the benign and malignant tumors of breast. The overall accuracy for breast cancer diagnosis achieved levels of 98.80% and 96.33% respectively using SVM classifier models against two widely used breast cancer benchmark datasets.

Mu et al (2005) proposed a technique to apply v-SVM learning instead of c-SVM learning to breast cancer detection, and perform v-SVM parameter selection based on the restricted leave-one-out error estimate using grid search with no need for validation data. An effective approach of Radial Basis Function Networks (RBFN) based on the self-organizing clustering results has also been applied to enhance the detection performance of using

only self-organizing maps. To evaluate the performance of this approach Wisconsin diagnosis breast cancer dataset was used. Experimental observation shows that the proposed approach offers significant performance compared to other existing approaches.

Ireaneus Anna Rejani et al (2009) proposed a tumor detection technique from mammogram. Their approach focuses on the solution of two problems: Detection of tumors as suspicious regions with a very weak contrast to their background and extracting features which categorize tumors. The tumor detection approach follows the technique of (a) mammogram enhancement (b) The segmentation of the tumor area (c) The extraction of features from the segmented tumor area (d) The use of SVM classifier.

The improvement is the alteration of the image quality to a better and further understandable level. The mammogram enhancement process consists of filtering, top hat operation and DWT. Then the contrast stretching is used to raise the contrast of the image. The segmentation of mammogram images plays a key role to enhance the detection and diagnosis of breast cancer. The well known segmentation approach used is thresholding. The features are extracted from the segmented breast area. Next stage categorizes the regions using the SVM classifier. The approach was tested on 75 mammographic images, from the mini-MIAS database. This approach obtained a sensitivity of 88.75%.

AFLP show is a DNA-based genotyping assay that has been mainly used for strain typing in plants and bacteria as proposed by Vos et al (1995). It identifies DNA restriction fragments by using PCR amplification and required relatively little starting genetic material. Moreover, AFLP is able to survey a target genome quickly and effectively, without the need of prior sequence knowledge. It is mostly used in botanical and microbiological fields, but, AFLP screening is also applied on the human genome as provided by Prochazka et al (2001).

Waiming Kong et al (2004) proposed that SVM can be applied to separate the AFLP data of cancer and the normal tissues. SVM is an efficient technique for general purpose supervised pattern recognition as proposed by Vapnik et al (1998) and has been applied successfully to many biological data recently including the identification of unknown genes using the gene expression data from DNA microarray hybridization experiments by Brown et al (2000) and classification of ovarian cancer tissue and prediction of protein structural test (Yu-Dong Cai et al 2001). Moreover, SVM was applied in searching translation initiation sites (Zien et al 2000) and for splice site recognition (Sonnenburg et al 2002). SVM was used to cluster the AFLP data and from the result it is observed that SVM can be used to differentiate cancer tissues from non-cancer tissues with a high level of accuracy.

SVMlight is an implementation of Support Vector Machines (SVMs) in SVMlight (Joachims 1999) was used to implement the SVM clustering. The jackknife test was used to observe the efficiency of SVM to differentiate among the data. Jackknife test is also known as the leave-one-out test, in which each data in the dataset is singled out as a tested data while all remaining data are used to train the SVM. One out of the 74 data was chosen to be the test data. The SVM was trained with the linear kernel on the remaining cancer and normal data, as positive and negative data respectively. After training, the test data was used to find the forecast of the SVM. This procedure was repeated for each of the 58 cancer data and 16 normal data. After the test of the entire data set, the sensitivity and the specificity of the result were calculated.

Machine learning techniques have been extensively applied in CAD to learn a hypothesis from diagnosed samples to aid the medical experts in making a diagnosis. In order to learn a well-performed hypothesis, a large amount of diagnosed models are required. Though the models can be easily collected from routine medical examinations, it is generally impossible for medical experts to make a diagnosis for each of the collected samples. If a theory could be learned in the presence of a large amount of undiagnosed samples, the heavy burden on the medical experts could be relieved.

A novel semi supervised learning algorithm named Co-Forest was proposed by Ming Li et al (2007). It expands the co-training example by using a well-known ensemble approach named Random Forest, which enables Co-Forest to evaluate the labeling confidence of undiagnosed samples and easily produce the final hypothesis. The experiments of the proposed approach are carried out on benchmark data sets to evaluate the effectiveness. Case studies on three medical data sets and a successful application to microcalcifications detection for breast cancer diagnosis illustrate that undiagnosed approaches are helpful in building CAD systems, and Co-Forest is able to improve the performance of the hypothesis that is learned on only a small amount of diagnosed samples by utilizing the available undiagnosed samples.

4.6 SUMMARY:

This chapter mainly summarized recent developments in cancer detection techniques. Various techniques have been used in the cancer detection approaches to improve the efficiency of cancer detection. Various applications like neural networks, image processing, CAD, nanotechnology are used in the cancer detection techniques. Each approach has its own uniqueness, advantages and limitations. As it is of immense importance to diagnose cancer as early as possible, many researches are still being done. These innovations are mainly due to the properties and the characteristics of existing methods.

CHAPTER 5
PROPOSED IMPLEMENTATION

This project presents lung cancer detection based on chest CT images using CNN. In the first stage, lung regions are extracted from CT image and in that region each slices are segmented to get tumors. The segmented tumor regions are used to train CNN architecture. Then, CNN is used to test the patient images. The main objective of this study is to detect whether the tumor present in a patient's lung is malignant or benign. Figure 6.1 shows the

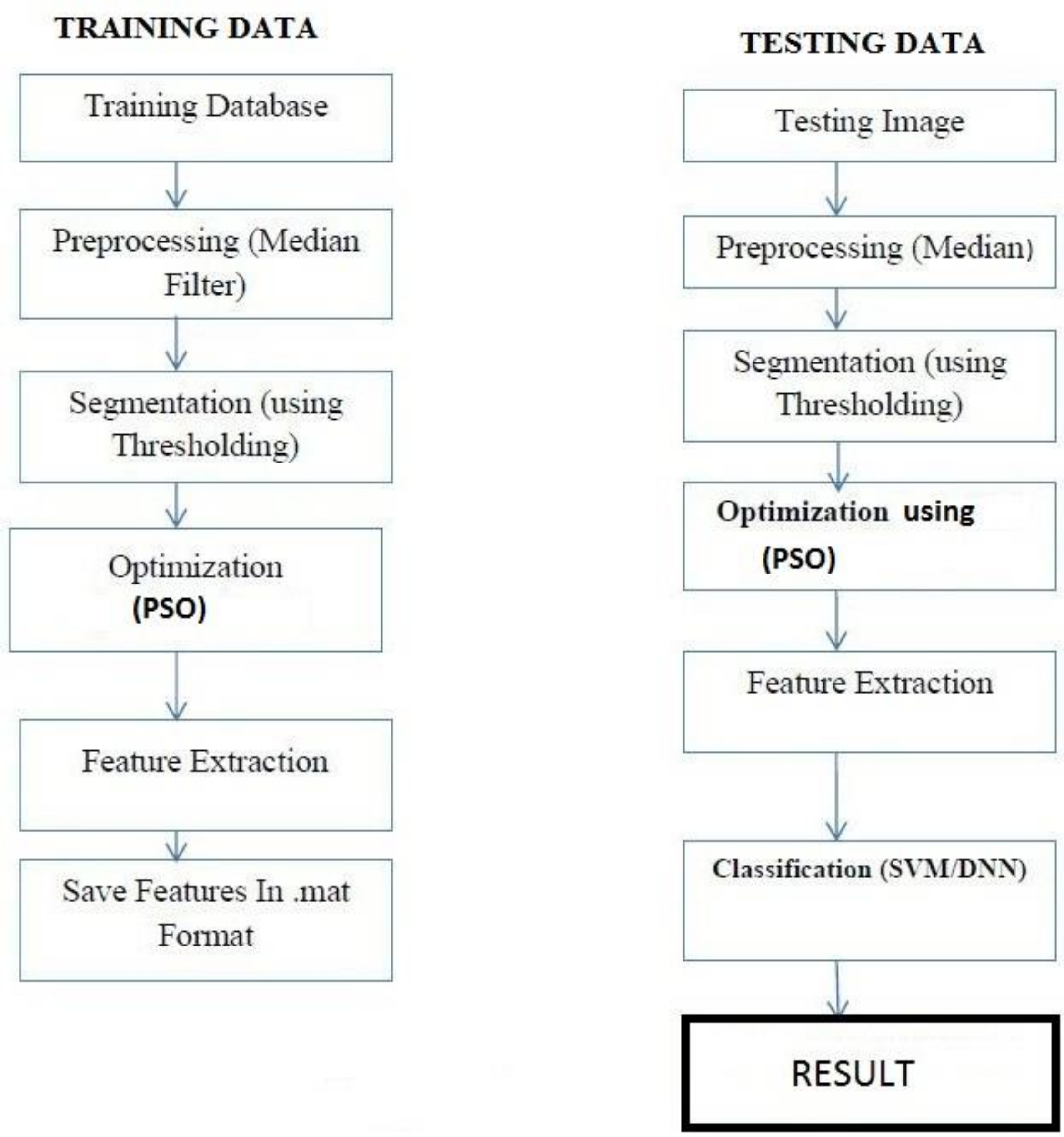


Fig 5.1: Block Diagram of the Proposed System

block diagram of the proposed system. As shown in the below figure, the trained system will able to detect the cancerous presence in lung CT image.

5.1 TRAINING DATA:

Dataset for training is obtained from Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI). LIDC and IDRI consist of 1000 CT scans of both large and small tumors saved in Digital Imaging and Communications in Medicine (DICOM) format [5].

5.1.1 TRAINING:

Back-propagation algorithm is used to train the Deep CNN to detect lung tumours in CT image of size 5×20×20. It consists of two phases. In the first phase, a CNN consists of multiple volumetric convolution, rectified linear units (ReLU) and max pooling layers is used to extract valuable volumetric features from input data. The second phase is the classifier. It has multiple FC and threshold layers, followed by a SoftMax layer to perform the high-level reasoning of the neural network. No scaling was applied to the CT images of the dataset to preserve the original values of the DICOM images as much as possible. During training, the random sub-volumes extracted from the CT images of the training set and are normalized according to an estimate of the normal distribution of the voxel values in the dataset.

5.2 PREPROCESSING:

In preprocessing stage, the median filter is used to restore the image under test by minimizing the effects of the degradations during acquisition. Various preprocessing and segmentation techniques of lung nodules are discussed in [6].The median filter simply replaces each pixel value with the median value of its neighbors including itself. Hence, the pixel values which are very different from their neighbors will be eliminated [7].

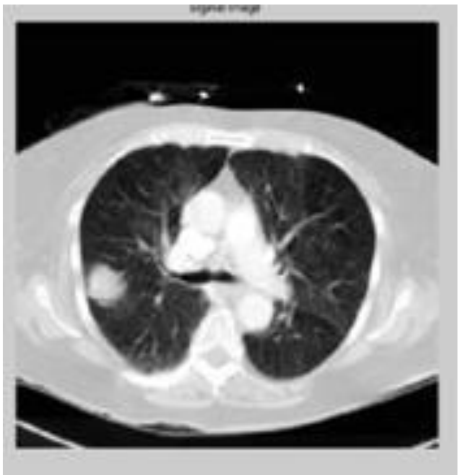


Fig 5.2: Input Image

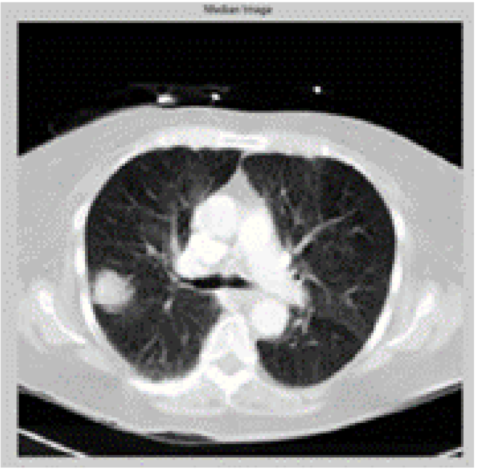


Fig 5.3: Median Filtered Image (Preprocessed)

5.3 IMAGE SEGMENTATION USING THRESHOLDING:

Image Segmentation is mainly to isolate the cancer cells from the background image and Image enhancement is to enhance the contrast between the Cancer Cells and the complete scan image of the lung.

The initial step is to segment the image. Segmentation subdivides an image into its essential parts of objects. The level to which this subdivision is carried depends on the problem being solved. It means that the segmentation should stop when the edge of the tumor is detected. Thus, the main aim is to isolate the tumor from its background.

It is very useful to separate out the regions of the image corresponding to objects which are necessary, from the regions of the image that correspond to background. Thresholding frequently offers a simple and suitable way to execute this segmentation on the basis of the diverse intensities or colors in the foreground and background regions of an image.

A grey scale or color image is the input to a Thresholding operation. The output in case of a simple implementation is a binary image representing the segmentation. Background is represented by black pixels and white pixels represent the foreground. The segmentation in simple implementation is decided by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold.

The pixel is set to white, in the output, if the pixel's intensity is higher than the threshold. On the other hand, if it is less than the threshold, it is set to black.

Segmentation is achieved by scanning the whole image pixel by pixel and labeling each pixel as object or background based on its binarized gray level.

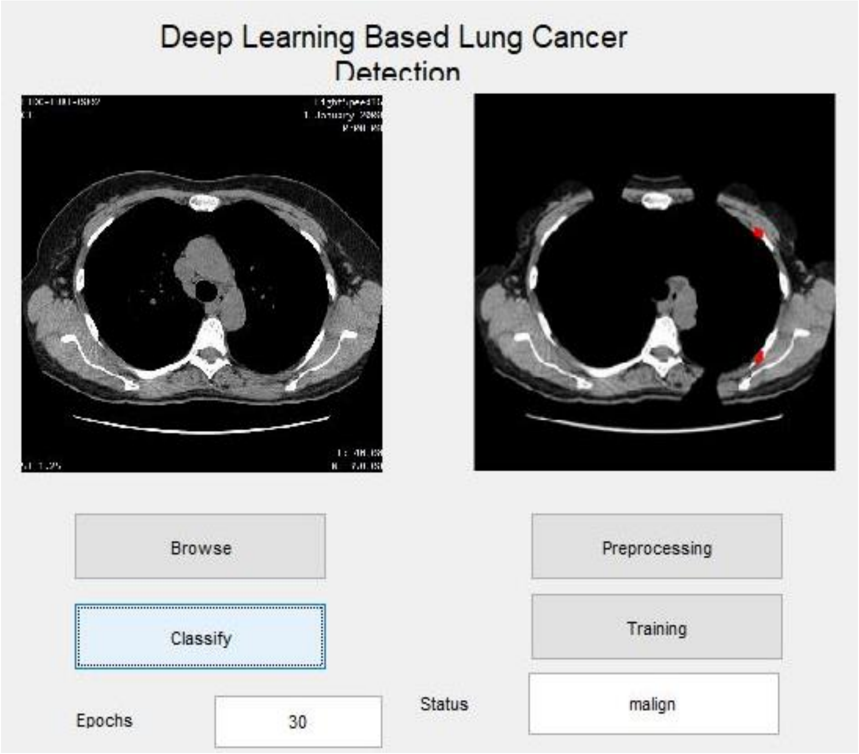
5.4 FEATURE EXTRACTION:

Feature extraction refers to several quantitative measurements of medical images classically used for decision making regarding the pathology of a structure or tissue. Feature extraction can be carried out in the spectral or the spatial domain.

After the extraction of features, selection of a subset of the most robust features is necessary, to improve classification accuracy and to reduce the overall complexity. Feature selection approaches use search techniques, which can be classified as Exhaustive, Heuristic, Non-deterministic etc.

CHAPTER 6
RESULT

The neural network based on convolutional and watershed segmentation has been implemented in MATLAB and the system is trained with sample data sets for the model to understand and familiarize the lung cancer. Sample images are fed as input to the trained model and the model at this stage is able to tell the presence of cancer and locate the cancer spot in the sample image of a lung. The process involves the feeding the input image, preprocessing, feature extraction, identifying the cancer spot and indicate the results to the user. The tumor is indicated with red dots as shown in the below figures,

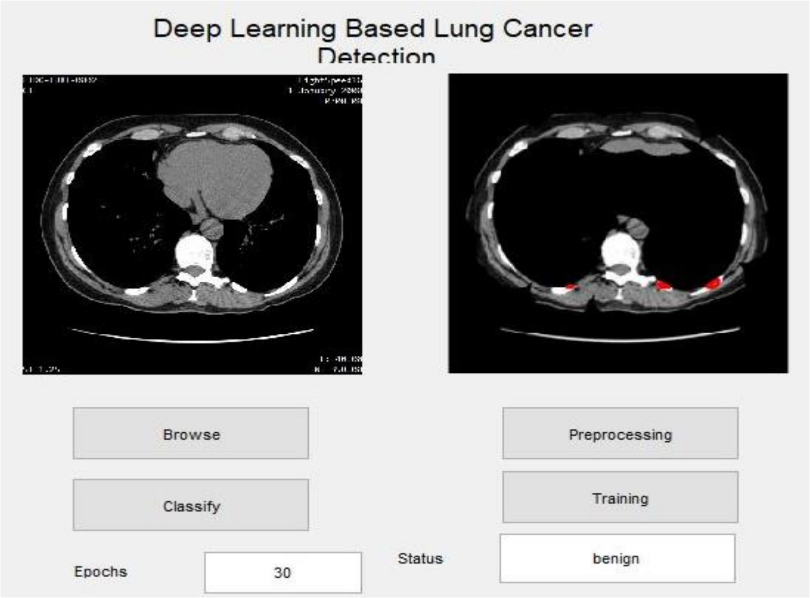


Epochs 30

Status malign

Fig 6.1: Malignant tumor present in the Lung Image.

- The above figure, Fig 6.1 shows the input image, and it shows the output as "Malignant Tumor". Malignant tumor is a tumor with cancerous cells present.



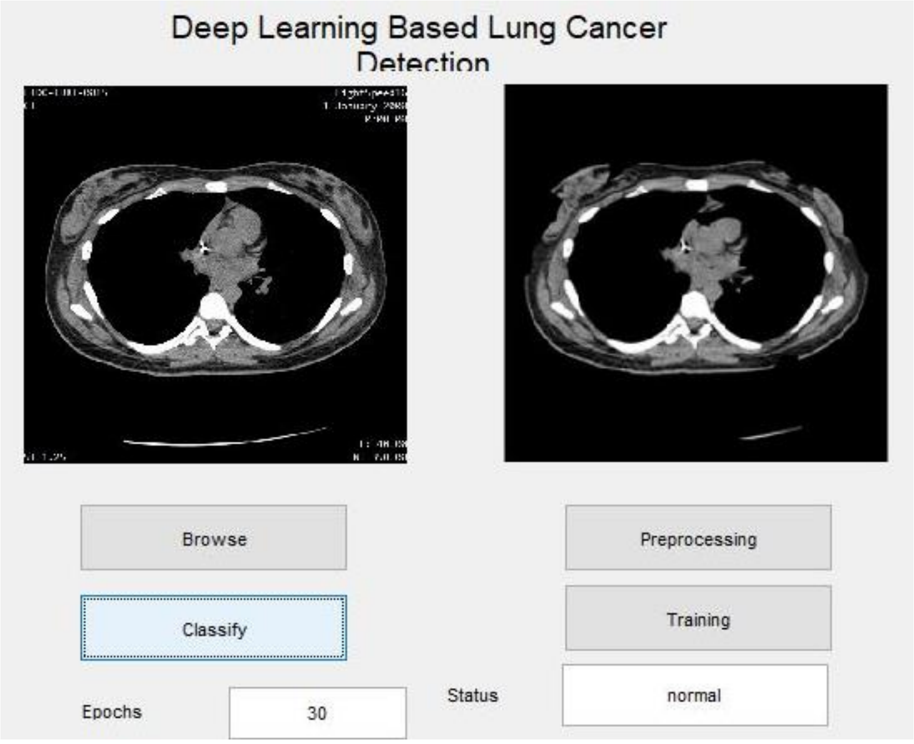
Deep Learning Based Lung Cancer Detection

Epochs 30

Status benign

Fig 6.2: Benign tumor present in the Lung Image.

- The above figure, Fig 6.1 shows the input image, and it shows the output as “Benign Tumor”. Benign tumor is a non-cancerous tumor.



Deep Learning Based Lung Cancer Detection

Epochs 30

Status normal

Fig 6.3: No tumor is present in the Lung Image.

- In the above figure, there is no tumor present in the Lung Image.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

A Convolutional Neural Network based system was implemented to detect the malignancy tissues present in the input lung CT image. Lung image with different shape, size of the cancerous tissues has been fed at the input for training the system. The proposed system is able to detect the presence and absence of cancerous cells with accuracy of about 96%.

In addition to Deep Convolutional Network, the same dataset was classified by multilayer perceptron network Back propagation algorithm with using GLCM features. The results show only 93% accuracy [10].

In this proposed work, the specificity obtained is 100% which shows that that there is no false positive detection. Also, the accuracy, sensitivity and specificity of the proposed system is high when compared to previously available conventional neural network based systems.

In the near future, the system will be trained with large datasets to diagnose the type of cancer with its size and shape. The overall accuracy of the system can be improved using 3D Convolutional Neural Network and also by improving the hidden neurons with deep network.

REFERENCES

[1] http://globocan.iarc.fr/Pages/fact_sheets_ cancer.aspx.

[2] https://www.livemint.com/Politics/3eXX60XBig4bWZ25Kr1iQO/India-recordedabout39-million-cancer-cases-in-2016data.html

[3] Using Deep Learning for Classification of Lung Tumors on Computed Tomography Images.

[4] M.S. Al-Tarawneh, "Lung cancer detection using image processing techniques," Leonardo Electronic Journal of Practices and Technologies, vol. 20, pp. 147– 58, May 2012.

[5] LUNA16, "Lung tumor analysis 2016." https:// luna16.grand-challenge.org/.

[6] A Manikandarajan, S Sasikala, Detection and Segmentation of Lymph Nodes for Lung Cancer Diagnosis. National Conference on System Design and Information Processing – 2013.

[7] Albert Chon, Peter Lu, NiranjanBalachandar "Deep Convolutional Neural Networks for Lung Cancer Detection".

[8] Wavelet Recurrent Neural Network for Lung Cancer Classification":3rd ICSTcomputer,2017.

[9] S Sasikala, M Ezhilarasi, Combination of Mammographic Texture Feature Descriptors for Improved Breast Cancer Diagnosis. Asian Journal of Information Technology, 2016.

[10] A.Kavitha, Anusiyasaral and P.Senthil," Design Model of Retiming Multiplier For FIR Filter &its Verification", International Journal of Pure and Applied Mathematics, Vol116 No12, 2017, pp. 239-247.

APPENDIX-A
SOURCE CODE

```
function varargout = guidemo(varargin)
% GUIDEMO MATLAB code for guidemo.fig
%      GUIDEMO, by itself, creates a new GUIDEMO or raises the existing
%      singleton*.
%
%      H = GUIDEMO returns the handle to a new GUIDEMO or the handle to
%      the existing singleton*.
%
%      GUIDEMO('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUIDEMO.M with the given input arguments.
%
%      GUIDEMO('Property','Value',...) creates a new GUIDEMO or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before guidemo_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to guidemo_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help guidemo

% Last Modified by GUIDE v2.5 23-Jan-2019 10:39:48

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                'gui_Singleton',  gui_Singleton, ...
                'gui_OpeningFcn', @guidemo_OpeningFcn, ...
                'gui_OutputFcn',  @guidemo_OutputFcn, ...
                'gui_LayoutFcn',  [] , ...
                'gui_Callback',   []);
if nargin && ischar(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end
 if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before guidemo is made visible.
function guidemo_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to guidemo (see VARARGIN)

% Choose default command line output for guidemo
handles.output = hObject;
   label='0';
set(handles.edit1,'String',label);clc
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes guidemo wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = guidemo_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Preprocessing.
function Preprocessing_Callback(hObject, eventdata, handles)
% hObject    handle to Preprocessing (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
 filename='.jpg';
filename1='.png';
datapath=pwd;
string1='\beningn\';
string2='\malign\';
string3='\normal\';
 for i=1:5
   filename2=strcat(num2str(i),filename);
   filename2=strcat(datapath,string1,filename2);
   filename3=strcat(num2str(i),filename1);
   a=imread(filename2);
   b=rgb2gray(a);
   c=b>150;
   c=medfilt2(c,[3 3]);
   se = strel('sphere',30);
   dilatedBW = imdilate(c, se);
   L = watershed_old(dilatedBW,1);
   L1=L.*dilatedBW;
   PreprocessesImages=zeros(512,512,3);
   PreprocessesImages(55:480,1:512,1)=a(55:480,1:512,1);
   PreprocessesImages(55:480,1:512,2)=a(55:480,1:512,2);
   PreprocessesImages(55:480,1:512,3)=a(55:480,1:512,3);
   a=double(PreprocessesImages);
   Inewc = a.*repmat(dilatedBW,[1,1,3]);
   PreprocessesImages=imresize(Inewc,[227 227]);
   PreprocessesImages=uint8(PreprocessesImages);
   cd 'PreprocessedDatabase\beningn'
   imwrite(PreprocessesImages,filename3);
   cd ..
   cd ..
end
 for i=1:5
      filename2=strcat(num2str(i),filename);
   filename2=strcat(datapath,string3,filename2);
   filename3=strcat(num2str(i),filename1);
   a=imread(filename2);
    b=rgb2gray(a);
   c=b>150;
   c=medfilt2(c,[3 3]);
   se = strel('sphere',30);
   dilatedBW = imdilate(c, se);
   L = watershed_old(dilatedBW,1);
   L1=L.*dilatedBW;
   PreprocessesImages=zeros(512,512,3);
   PreprocessesImages(55:480,1:512,1)=a(55:480,1:512,1);
   PreprocessesImages(55:480,1:512,2)=a(55:480,1:512,2);
   PreprocessesImages(55:480,1:512,3)=a(55:480,1:512,3);
   a=double(PreprocessesImages);
   Inewc = a.*repmat(dilatedBW,[1,1,3]);
   PreprocessesImages=imresize(Inewc,[227 227]);
   PreprocessesImages=uint8(PreprocessesImages);
   cd 'PreprocessedDatabase\normal'
   imwrite(PreprocessesImages,filename3);
   cd ..
   cd ..
end
for i=1:5
      filename2=strcat(num2str(i),filename);
   filename2=strcat(datapath,string2,filename2);
   filename3=strcat(num2str(i),filename1);
   a=imread(filename2);
    b=rgb2gray(a);
   c=b>150;
   c=medfilt2(c,[3 3]);
   se = strel('sphere',30);
   dilatedBW = imdilate(c, se);
   L = watershed_old(dilatedBW,1);
   L1=L.*dilatedBW;
   PreprocessesImages=zeros(512,512,3);
   PreprocessesImages(55:480,1:512,1)=a(55:480,1:512,1);
   PreprocessesImages(55:480,1:512,2)=a(55:480,1:512,2);
   PreprocessesImages(55:480,1:512,3)=a(55:480,1:512,3);
   a=double(PreprocessesImages);
   Inewc = a.*repmat(dilatedBW,[1,1,3]);
   PreprocessesImages=imresize(Inewc,[227 227]);
   PreprocessesImages=uint8(PreprocessesImages);
   cd 'PreprocessedDatabase\malign'
   imwrite(PreprocessesImages,filename3);
   cd ..
```

```
    cd ..
end
helpdlg('PreProcessing Completed')


% --- Executes on button press in Training.
function Training_Callback(hObject, eventdata, handles)
% hObject    handle to Training (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
datapath=pwd;
Datasetpath=fullfile(datapath,'PreprocessedDatabase');
trainingDS = imageDatastore(Datasetpath,'IncludeSubfolders',1,...
    'LabelSource','foldernames');
tbl = countEachLabel(trainingDS);
 layers=[imageInputLayer([227 227 3])
        convolution2dLayer(5,20)
        reluLayer
        maxPooling2dLayer(2,'stride',2)
        convolution2dLayer(5,20)
        reluLayer
        maxPooling2dLayer(2,'stride',2)
        convolution2dLayer(5,20)
        reluLayer
        maxPooling2dLayer(2,'stride',2)
        fullyConnectedLayer(3)
        softmaxLayer
        classificationLayer()]
  maxEpochs=900;
lr = 0.0001;
miniBatchSize = 64; % lower this if your GPU runs out of memory.
 opts = trainingOptions('sgdm', ...
    'LearnRateSchedule', 'none',...
    'InitialLearnRate', lr,...
    'MaxEpochs', maxEpochs, ...
     'MiniBatchSize', miniBatchSize,...
     'Plots','training-progress');
  analyzeNetwork(layers)
net = trainNetwork(trainingDS, layers, opts);
 save('trainedNet.mat','net')
 disp('exit');
% This could take over an hour to run, so lets stop and load a pre-traiend
% version that used the same data
warndlg('Training Completed');

% --- Executes on button press in Browse.
function Browse_Callback(hObject, eventdata, handles)
% hObject    handle to Browse (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 [filename, pathname] = uigetfile('*.*', 'Pick a Image');
    if isequal(filename,0) || isequal(pathname,0)
      warndlg('User pressed cancel')
   else
   filename=strcat(pathname,filename);
   a=imread(filename);
   axes(handles.axes1);
   imshow(a);
   handles.a=a;
%        handles.pathname=pathname;
    label='0';
set(handles.edit1,'String',label);clc
% set(handles.pathname,'String',pathname);clc
    end
% Update handles structure
```

```
guidata(hObject, handles);
 % --- Executes on button press in Classify.
function Classify_Callback(hObject, eventdata, handles)
% hObject    handle to Classify (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
  [filename, pathname] = uigetfile('*.*', 'Pick a Image');
load('trainedNet.mat');
im=handles.a;
b=rgb2gray(im);
    c=b>150;
    c=medfilt2(c,[3 3]);
    se = strel('sphere',30);
    dilatedBW = imdilate(c, se);
    L = watershed_old(dilatedBW,1);
    L1=L.*dilatedBW;
    PreprocessesImages=zeros(512,512,3);
    PreprocessesImages(55:480,1:512,1)=im(55:480,1:512,1);
    PreprocessesImages(55:480,1:512,2)=im(55:480,1:512,2);
    PreprocessesImages(55:480,1:512,3)=im(55:480,1:512,3);
    a=double(PreprocessesImages);
    Inewc = a.*repmat(dilatedBW,[1,1,3]);
    PreprocessesImages=imresize(Inewc,[227 227]);
    PreprocessesImages=uint8(PreprocessesImages);
im=PreprocessesImages;
[M ,a]=plotfeatures(im);
[B,L,N] = bwboundaries(M);
axes(handles.axes2);
imshow(a);
% imshow(b);
hold on;
 label = char(classify(net,im)); % classify with deep learning
```

```
label=char(label);
 nn;
if strcmp(label,'malign')
    for k=1:length(B),
    boundary = B{k};
    if(k > N)
    axes(handles.axes2);
     plot(boundary(:,2), boundary(:,1), 'g','LineWidth',2);
    else
    axes(handles.axes2);
     plot(boundary(:,2), boundary(:,1), 'r','LineWidth',2);
    end
    end
    elseif strcmp(label,'beningn')
     for k=1:length(B),
    boundary = B{k};
    if(k > N)
    axes(handles.axes2);
     plot(boundary(:,2), boundary(:,1), 'g','LineWidth',2);
    else
    axes(handles.axes2);
     plot(boundary(:,2), boundary(:,1), 'r','LineWidth',2);
    end
    end
    end

% imshow(im);
set(handles.edit1,'String',label);clc
 disp(label)
%  end

function edit1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'String') returns contents of edit1 as text

%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.

function edit1_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called


% Hint: edit controls usually have a white background on Windows.

%       See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),

get(0,'defaultUicontrolBackgroundColor'))

   set(hObject,'BackgroundColor','white');

end


function edit2_Callback(hObject, eventdata, handles)

% hObject    handle to edit2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

 % Hints: get(hObject,'String') returns contents of edit2 as text

%        str2double(get(hObject,'String')) returns contents of edit2 as a double

 % --- Executes during object creation, after setting all properties.

function edit2_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

 % Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double
 % --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
 % Hint: edit controls usually have a white background on Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

APPENDIX – B
MATLAB

B.1 INTRODUCTION:

MATLAB is a high-performance language for specialized registering. It coordinates calculation, portrayal, and programming in an easy to-use condition where issues and courses of action are imparted in normal numerical documentation. MATLAB speaks to network research focus and was formed at first to give basic access to system programming made by LINPACK (straight structure pack) and EISPACK (Eigen structure group) adventures. MATLAB is thus founded on a foundation of refined system programming in which the basic segment is bunch that does not require pre-dimensioning which to deal with various specific enlisting issues, especially those with structure and vector definitions, in a modest quantity of time. MATLAB highlights a group of uses explicit arrangements called tool compartments. Important to most clients of MATLAB, tool stash permits learning and applying specific innovation. These are complete accumulations of MATLAB capacities (M-records) that stretch out the MATLAB condition to take care of specific classes of issues. Territories in which tool stash are accessible incorporate sign handling, control framework, neural systems, fluffy rationale, wavelets, reproduction and numerous others. Run of the mill employments of MATLAB include: Math and calculation, Algorithm improvement, Data obtaining, Modeling, recreation, prototyping, Data investigation, investigation, representation, Scientific and designing illustrations, Application advancement, including graphical UI building.

B.2 BASIC BUILDING BLOCKS OF MATLAB:

The basic building block of MATLAB is MATRIX. The basic information type is the cluster. Vectors, scalars, genuine lattices and complex network are dealt with as explicit class of this essential information type. The implicit capacities are improved for vector tasks. No measurement articulations are required for vectors or exhibits.

B.2.1 MATLAB WINDOW: The MATLAB works based on five windows: Command window, Workspace window, Current catalog window, Command history window, Editor Window, Graphics window and Online-help window.

B.2.1.1 Command Window:

The command window is where the user types MATLAB directions and articulations at the brief (>>) and where the yield of those directions is shown. It is opened when the application program is propelled. All directions including client composed projects are composed in this window at MATLAB prompt for execution.

B.2.1.2 Work Space Window :

MATLAB defines the workspace as the arrangement of factors that the client makes in a work session. The workspace program demonstrates these factors and some data about them. Double tapping on a variable in the workspace program dispatches the Array Editor, which can be utilized to get information.

B.2.1.3 Current Directory Window:

The current Directory tab shows the substance of the present index, whose way is appeared in the present catalog window. For instance, in the windows working framework the way may be as per the following: C:\MATLAB\Work, showing that catalog "work" is a subdirectory of the fundamental index "MATLAB"; which is introduced in drive C. Tapping on the bolt in the present index window demonstrates a rundown of as of late utilized ways. MATLAB utilizes a pursuit way to discover M-records and other MATLAB related documents. Any document keeps running in MATLAB must dwell in the present registry or in an index that is on search path.

B.2.1.4 Command History Window:

The Command History Window contains a record of the directions a client has entered in the order window, including both present and past MATLAB sessions. Recently entered MATLAB directions can be chosen and re-executed from the order history window by right tapping on an order or arrangement of directions. This is valuable to choose different alternatives notwithstanding executing the directions and is helpful element when trying different things with different directions in a work session.

B.2.1.5 Editor Window: The MATLAB editor is both a text proofreader particular for making M-documents and a graphical MATLAB debugger. The editorial manager can show up in a window independent from anyone else, or it very well may be a sub window in the work area. In this window one can compose, alter, make and spare projects in records called M-documents.

MATLAB manager window has various draw down menus for errands, for example, sparing, survey, and troubleshooting documents. Since it plays out some basic checks and furthermore uses shading to separate between different components of code, this word processor is suggested as the device of decision for composing and editing M-functions.

B.2.1.6 Graphics or Figure Window:

The output of all graphic commands typed in the command window is seen in this window.

B.2.1.7 Online Help Window:

MATLAB provides online help for all it's worked in capacities and programming language develops. The chief method to get help online is to utilize the MATLAB help program, opened as a different window either by tapping on the question mark image (?) on the work area toolbar, or by composing help program at the brief in the order window. The assistance Browser is an internet browser incorporated into the MATLAB work area that shows a Hypertext Markup Language (HTML) archives.

The Help Browser comprises of two sheets, the assistance guide sheet, used to discover data, and the presentation sheet, used to see the data. Clear as crystal tabs other than guide sheet are utilized to perform a search.

B.3 MATLAB FILES:

MATLAB has three types of files for storing information. They are: M-files and MAT-files.

B.3.1 M-FILES:

These are standard ASCII content document with 'm' expansion to the record name and making own frameworks utilizing M-documents, which are content records containing MATLAB code. MATLAB editorial manager or another content manager is utilized to make a document containing similar proclamations which are composed at the MATLAB order line and spare the record under a name that finishes in .m. There are two sorts of M-records:

1. Content Files

It is a M-record with a lot of MATLAB directions in it and is executed by composing name of document on the order line. These documents take a shot at worldwide factors right now present in that condition.

2. Capacity Files

A capacity document is likewise a M-record aside from that the factors in a capacity document are on the whole neighborhood. This kind of records starts with a capacity definition line.

B.3.2 MAT-Files:

These are double information records with '.tangle' expansion to the document that are made by MATLAB when the information is spared. The information written in a unique organization that no one but MATLAB can peruse. These are situated into MATLAB with 'load' order.

B.4 THE MATLAB SYSTEM:

The MATLAB framework comprises of five fundamental parts:

B.4.1 DEVELOPMENT ENVIRONMENT:

This is the arrangement of apparatuses and offices that help you use MATLAB capacities and documents. A large number of these devices are graphical UIs. It incorporates the MATLAB work area and Command Window, an order history, a supervisor and debugger, and programs for survey help, the workspace, documents, and the pursuit way.

B.4.2 THE MATLAB MATHEMATICAL FUNCTION:

This is a huge accumulation of computational calculations extending from rudimentary capacities like total, sine, cosine, and complex number juggling, to progressively modern capacities like framework reverse, lattice eigen esteems, Bessel capacities, and quick Fourier changes.

B.4.3 THE MATLAB LANGUAGE:

This is an abnormal state framework/cluster language with control stream articulations, capacities, information structures, input/yield, and article situated programming highlights. It permits both "programming in the little" to quickly make straightforward discard projects, and "programming in the enormous" to make total huge and complex application programs.

B.4.4 GRAPHICS:

MATLAB has broad offices for showing vectors and lattices as charts, just as commenting on and printing these diagrams. It incorporates abnormal state capacities for two-dimensional and three-dimensional information representation, picture preparing, activity, and introduction designs. It likewise incorporates low-level capacities that enable you to completely redo the presence of designs just as to fabricate total graphical UIs on your MATLAB applications.

B.4.5 THE MATLAB APPLICATION PROGRAM INTERFACE (API):

This is a library that enables you to compose C and FORTRAN programs that cooperate with MATLAB. It incorporates offices for calling schedules from MATLAB (dynamic connecting), calling MATLAB as a computational motor, and for perusing and composing MAT-records.

B.5 SOME BASIC COMMANDS:

pwd prints working registry

Demo exhibits what is conceivable in Mat lab

Who records the majority of the factors in your Mat lab workspace?

Whose rundown the factors and portrays their framework

size clear deletes factors and capacities from memory

clear x eradicates the grid 'x' from your workspace near to itself, shuts the present figure window

figure makes an unfilled figure window

hang on holds the present plot and all pivot properties so ensuing charting directions add to the current diagram

hold off sets the following plot property of the present tomahawks to "supplant" discover files of nonzero components e.g.:

d = find(x>100) restores the files of the vector x that are more prominent
than 100 break end execution of m-record or WHILE or FOR circle

load filename.dat loads the substance of filename.dat into the variable filename

xlabel(' ') : Allows you to mark x-hub

ylabel(' ') : Allows you to mark y-hub
title(' ') : Allows you to give title for plot

subplot() : Allows you to make numerous plots in a similar window

B.6 SOME BASIC PLOT COMMANDS:

Kinds of plots:

plot(x,y) creates a Cartesian plot of the vectors x and y plot(y) makes a plot of y versus the numerical estimations of the components in the y-vector

semilogx(x,y) plots log(x) versus y

semilogy(x,y) plots x versus log(y)

loglog(x,y) plots log(x) versus log(y)

polar(theta,r) makes a polar plot of the vectors r and theta where theta
is in radians.

bar(x) makes a visual chart of the vector x. (Note likewise the
order stairs(x))

bar(x, y) makes a structured presentation of the components of
the vector y, finding the bars as per the vector components of
'x'

Plot portrayal:

matrix creates a framework on the designs plot

title('text') places a title at top of illustrations plot

xlabel('text') writes 'content' underneath the x-hub of a plot

ylabel('text') writes 'content' adjacent to the y-hub of a plot

text(x,y,'text') writes 'content' at the area (x,y)

text(x,y,'text','sc') composes 'content' at point x,y accepting lower left corner is (0,0) what's more, upper right corner is (1,1)

axis([xmin xmax ymin ymax]) sets scaling for the x-and y-tomahawks on the current plot.

B.7 ALGEBRAIC OPERATIONS IN MATLAB:

Scalar Calculations:

+ Addition

- Subtraction

* Multiplication

/ Right division (a/b means a ÷ b)

\ left division (a\b means b ÷ a)

^ Exponentiation

For example: 3*4 executed in 'matlab' gives ans=12

4/5 gives ans=0.8

Cluster items: Recall that expansion and subtraction of grids included expansion or subtraction of the individual components of the lattices. Now and then it is wanted to just duplicate or partition every component of a grid by the relating component of another framework 'exhibit activities".

B.8 MATLAB WORKING ENVIRONMENT:

B.8.1 MATLAB DESKTOP :

Matlab Desktop is the main Matlab application window. The work area contains five sub windows, the order window, the workspace program, the present catalog window, the direction history window, and at least one figure windows, which are demonstrated just when the client shows a realistic.

The direction window is the place the client types MATLAB directions and articulations at the brief (>>) and where the yield of those directions is shown. MATLAB characterizes the workspace as the arrangement of factors that the client makes in a work session.

The workspace program demonstrates these factors and some data about them. Double tapping on a variable in the workspace program dispatches the Array Editor, which can be utilized to get data and salary occurrences alter certain properties of the variable.

The present Directory tab over the workspace tab demonstrates the substance of the present catalog, whose way is appeared in the present index window. For instance, in the windows working framework the way may be as per the following: C:\MATLAB\Work, showing that index "work" is a subdirectory of the fundamental catalog "MATLAB"; WHICH

IS INSTALLED IN DRIVE C. tapping on the bolt in the present catalog window demonstrates a rundown of as of late utilized ways. Tapping on the catch to one side of the window enables the client to change the present catalog.

MATLAB utilizes a hunt way to discover M-records and other MATLAB related documents, which are compose in registries in the PC record framework. Any document keeps running in MATLAB must dwell in the ebb and flow index or in a catalog that is on inquiry way. As a matter of course, the records provided with MATLAB and math works tool stash are incorporated into the hunt path. The most straightforward approach to see which registries are soon the inquiry way, or to include or change a pursuit way, is to choose set way from the File menu the work area, and after that utilization the set way exchange box.
It is great practice to add any ordinarily utilized registries to the pursuit way to maintain a strategic distance from over and over having the change the present registry.

The Command History Window contains a record of the directions a client has entered in the order window, including both present and past MATLAB sessions. Recently entered MATLAB directions can be chosen and re-executed from the order history window by right tapping on an order or grouping of directions.

This activity dispatches a menu from which to choose different choices notwithstanding executing the directions. This is valuable to choose different choices notwithstanding executing the directions. This is a helpful element when trying different things with different directions in a work session.

B.8.2 USING THE MATLAB EDITOR TO CREATE M-FILES:

The MATLAB editor is both a word processor specific for making M-documents and a graphical MATLAB debugger. The proofreader can show up in a window independent from anyone else, or it very well may be a sub window in the work area. M-records are meant by the augmentation .m, as in pixelup.m.

The MATLAB editorial manager window has various draw down menus for undertakings, for example, sparing, survey, and troubleshooting records. Since it plays out some straightforward checks and furthermore uses shading to separate between different components of code, this content manager is prescribed as the apparatus of decision for composing and altering M-capacities.

To open the editorial manager, type alter at the brief opens the M-document ‘filename.m’ in a supervisor window, prepared for altering. As noted before, the record must be in the current directory, or in a directory in the search path.

B.8.3 GETTING HELP:

The principal way to get help online is to utilize the MATLAB help program, opened as a different window either by tapping on the question mark image (?) on the work area toolbar, or by composing help program at the brief in the order window. The assistance Browser is an internet browser incorporated into the MATLAB work area that shows a Hypertext Markup Language (HTML) records. The Help Browser comprises of two sheets, the assistance guide sheet, used to discover data, and the showcase sheet, used to see the data. Clear as crystal tabs other than guide sheet are utilized to perform a search.