

LUNG CANCER DETECTION USING MACHINE LEARNING

**A Project Report submitted in partial fulfillment of the requirements for the
award of the degree of**

**BACHELOR OF
TECHNOLOGY IN
COMPUTER SCIENCE AND
ENGINEERING**

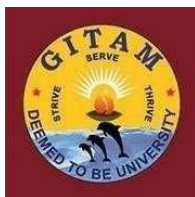
Submitted by

S. Teja Naidu	121910314020
J. Manish	121910314010
M. Sai Suraj Mohan	121910314045
G. Teja	121910314050

Under the esteemed guidance of

Dr. Prem Kumar Singh

Associate Professor



**DEPARTMENT OF COMPUTER SCIENCE &
ENGINEERING GITAM**

**(Deemed to be
University)**

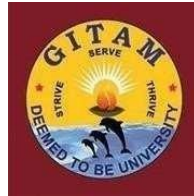
**VISAKHAPATNAM
NOVEMBER 2022**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

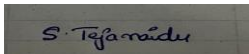
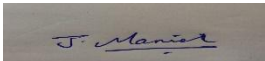

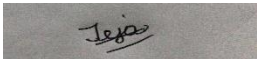
(Deemed to be University)



DECLARATION

I/We, hereby declare that the project report entitled “**Lung Cancer Detection using Machine learning**” is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering. The work has not been submitted to any college or University for the award of any degree or diploma.

Date:

Registration No(s).	Name(s)	Signature(s)
121910314020	S. Teja Naidu	
121910314010	J. Manish	
121910314045	M. Sai Suraj Mohan	
121910314050	G. Teja	

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM SCHOOL OF TECHNOLOGY**

GITAM

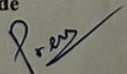
(Deemed to be University)



CERTIFICATE

This is to certify that the project report entitled "**LUNG CANCER DETECTION USING MACHINE LEARNING**" is a bonafide record of work carried out by **S. TEJA NAIDU (121910314020), MANISH JAVVADI (121910314010), M.SAI SURAJ MOHAN(121910314045), G.TEJA(121910314050)** students submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

Project Guide

A handwritten signature in black ink, appearing to be 'Dr. Prem Kumar Singh'.

Dr. PREM KUMAR SINGH

ASSOCIATE PROFESSOR

Head of the Department

Dr. R. SIREESHA

Professor

TABLE OF CONTENTS

1.	Abstract	5
2.	Introduction	6
3.	Literature Review	27
4.	Problem Identification	29
5.	Objectives	30
6.	Existing System	31
7.	Proposed System	32
8.	Overview of Technologies	39
9.	Implementation	41
10.	Results and Discussions	52
11.	Conclusion and Future scope	54
12.	References	55

1. ABSTRACT

Lung cancer is the deadliest diseases that is more probable to grow rapidly with the spread of metastasis. Metastasis is the formation of additional secondary malignant growths away from the primary cancer location. According to GLOBOCAN- 2020 assessment on cancer occurrences among people and fatalities produced that was conducted by the International Agency for Research on Cancer, approximately about 193 lakhs new cancer cases were diagnosed worldwide, with around 100 lakhs cancer deaths. Among the various types of cancer that caused death worldwide, Lung cancer is constantly being termed as the prime reason of cancer death, with an estimated 18 lakhs deaths (18%), followed by colorectal (9.4%), liver (8.3%), stomach (7.7%), and female breast (6.9%) cancers It has been identified as one of the world's prime causes of death.

The ability to recognize and diagnose the malignant nodules and categorize them as benign, malignant, or indeterminate(normal) on chest computed-tomography (CT) is extremely crucial for early lung cancer diagnosis and treatment. For that purpose, with the increasing advancement of technology various machine learning and deep learning techniques have entered the picture to diagnose lung cancer where the machines are taught to predict outcomes. Nowadays, researchers are utilizing computer algorithms to develop computer-aided programs that are better than radiologists or pathologists at detecting malignancy in CT scans. For an instance, in one AI study, researchers worked on an algorithm to identify two forms of lung cancer with 97% precision, as well as to detect cancer-related genetic alterations. By using such means, detecting the cancerous pulmonary lung nodules accurately can aid in the early manifestation of lung cancer. However, developing a reliable nodule detection approach is difficult due to the lack of consistency of patterns of lung nodules and the complex nature of the surrounding conditions. In our proposed computer-aided design system we perform cancerous nodule detection in 2 stages. The discovery of potential candidate nodules is the first of two processes. In this stage, we employ maximum intensity projection images to improve the effectiveness in automatic detection of cancerous pulmonary nodules, particularly the tiny lesions using convolutional neural networks (CNNs).

In this project we use the lung cancer screening thoracic computed tomography (CT) images from the IQ-OTHNCCD lung cancer dataset which is collected from Kaggle. The dataset contains 1190 images totally. These 1190 images are the CT scan slices of 110 cases. Each case approximately having 10 slices. These images are categorized into 3 classes: normal, benign, and malignant. Out of these, 40 cases are malignant cases; 15 cases are benign cases; and 55 cases are normal cases. In this project we try to build the Convolutional Neural Network to classify the images into one of the three classes.

2. INTRODUCTION

2.1. OVERVIEW

In our project, we are using deep learning techniques to identify and detect pulmonary lung nodules which helps in detecting lung cancer. One of the most predominant techniques that deep learning uses is the Artificial Neural Networks. Artificial neural network is the most practically used unit of deep learning. Deep learning is a part of ML which falls under AI.

2.2. ARTIFICIAL INTELLIGENCE

Artificial intelligence researches how to create intelligent programs and machines that can work on providing solutions for problems creatively, which was supposedly been done by humans all these days.

2.3. MACHINE LEARNING

Machine learning is the subset of AI that lays the concept of automating the learning process of machines by building models by feeding in the right data required for the models to automatically learn the situation and improve from experience without hard coding the programs.

2.4. DEEP LEARNING

Deep learning is the subset of machine learning which primarily works with artificial neural networks, which are algorithms that mimics the function of human brain.

Machine Learning and Deep Learning aids AI to solve data driven problems by providing various machine learning algorithms and neural networks.

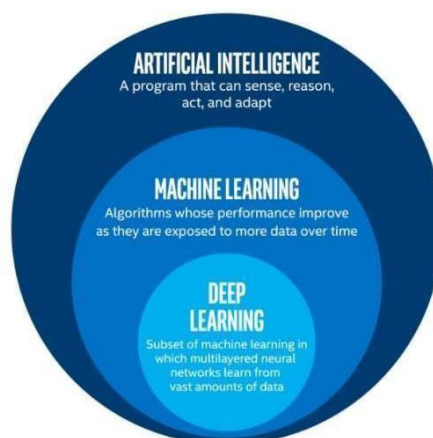


Figure 2.1: Picture showing the interrelationship between Artificial Intelligence, Machine Learning and Deep Learning.

To understand the working mechanism of our project, it is essential for us to understand a few terms.

2.5. NEURAL NETWORK

A neural network is the network of artificial neurons that mimics the functions the brain, which is made up of about 1000 million neurons with 6000 billion connections between network of neurons. A neuron has a soma which is the cell body, which houses the nucleus, numerous dendrites, which receive and transmit input signals, and synapses, which are simply connections between neurons. A dendrite is the input structure which perceives inputs. Soma is the spot where the calculation processes take place. Axon is the channel for output.

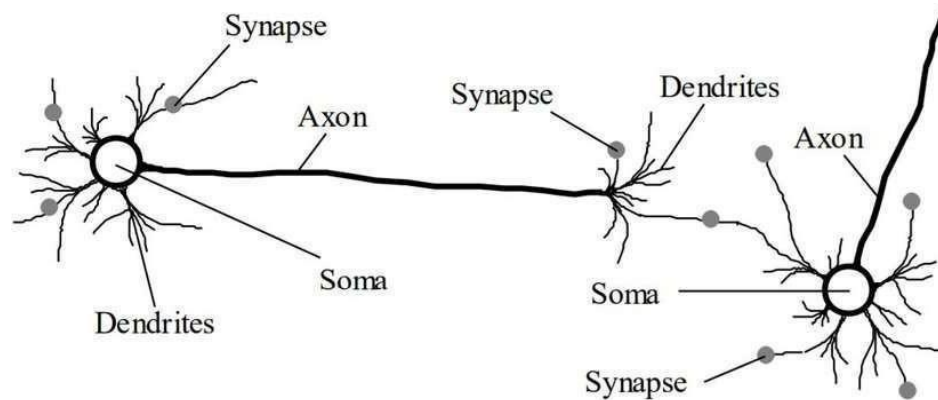


Figure 2.2: A Single Neuron.

2.5.1. Artificial Neuron

A neural network, like a brain neuron, is made up of artificial neurons. The artificial neurons are also called as perceptron. Every artificial neuron has an entity, the weight, which is used to determine how strong a connection is, a linear function that is computed and an activation function that calculates the weighted sum of the linear function which is then compared to a threshold value. The incoming connections of the neuron receives the input, the activation function makes non-linear decisions and the outgoing connections deliver the activation signal.

2.5.2 Artificial Neural Network

Artificial Neural Networks are the networks of perceptron that are motivated by the biological neuron system that mimic the functions of the human brain to resolve complex problems that we encounter and can perform a variety of tasks with a large amount of data.

To solve complex problems using artificial neural networks, understanding the relationships in a given set of data is a prerequisite. For this purpose, different learning algorithms like supervised, unsupervised and reinforcement learning are employed with the help of which the machines learn the patterns from the data that is given to the model and try to learn the features and patterns in the data which allows us to make predictions or classify out input into the right output spaces. Neural networks must be trained on numerous data so that they learn the patterns and gain accuracy over time. By training the network on a number of epochs, different correlations and hidden patterns in raw data can be identified which are used to cluster and classify the data which aids in accurate prediction of future results.

2.5.2.1. Working Of Artificial Neural Network

Artificial neural networks (ANNs) are composed of an input layer to which input is given, one or more number of hidden layers where processing takes place, and an output layer. Every node in the neural network is considered as a neuron(perceptron). The artificial neurons communicate with each other and create a network which is nothing but the neural network. Every node in the network connects to other nodes and has a respective weight and a limit value. A specific neuron gets activated when the resultant of that individual neuron exceeds the specified value. After the neuron gets activated, the data from that node is forwarded to the node of the next layer in the network. Otherwise, no data is passed forward. Neural networks functions only when some data is provided as input to it. The data that is fed to the neural network passes through several layers in order to obtain a certain output.

LEARNING TECHNIQUES USED IN NEURAL NETWORKS

2.6. FORWARD PROPAGATION

Forward propagation is a learning technique in which data is sequentially passed between the corresponding input layer, hidden layers, and output layer. It is the preliminary step of training a neural network model. Here we move from left to right from input to the output layer by applying the necessary activation function like the SoftMax, sigmoid or tanh which is used to which is used to calculated the weighted sum of inputs and bias and then activate a neuron accordingly.

2.6.1. Perceptron Learning

A perceptron is a basic unit to build an Artificial neural network system. The perceptron model is given an input, calculate the weighted sum and the system returns 1 when the weighted sum is greater than the mentioned threshold value else the system returns 0.

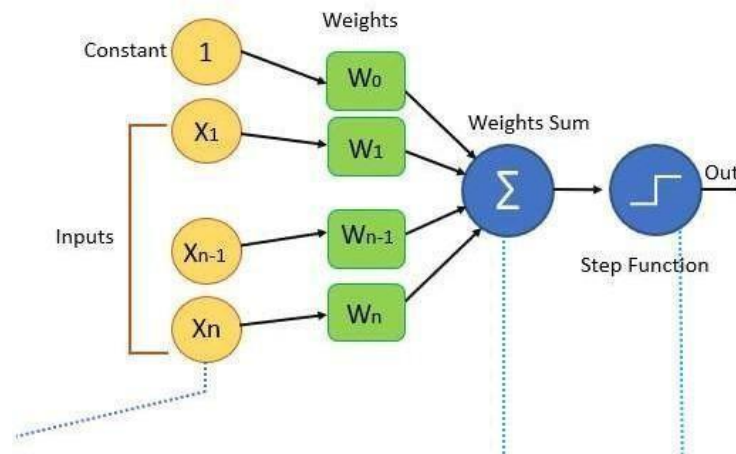


Figure 2.3: Picture depicting Perceptron Learning Model.

2.6.1.1. Working Of Perceptron Learning

All of the characteristics and features that we wish to train the neural network with, will be provided to the model as input. Initially we assign random values as weights of every neuron, then the model is trained and in the process of training the weights automatically gets updated after each training error. Each input value is multiplied with the weight assigned to it at the start, and the sum of all multiplied values is known as a weighted sum.

We also use activation function and bias in this process.

a) ACTIVATION FUNCTION

The activation functions is the crucial element in a neural network. They enable the models to solve problems by generating nonlinear functions. Sigmoid, Tanh, and SoftMax are the three most commonly employed activation functions.

By comparing it to the activation function thresholds, these functions enable or disable neurons from getting activated and fired to next layer.

Activation functions are utilized in both forward and backward propagation, where they are used while the loss calculation when the resultant of a particular function is compared to an assigned number in forward propagation and to update the neural network parameters accordingly in backward propagation.

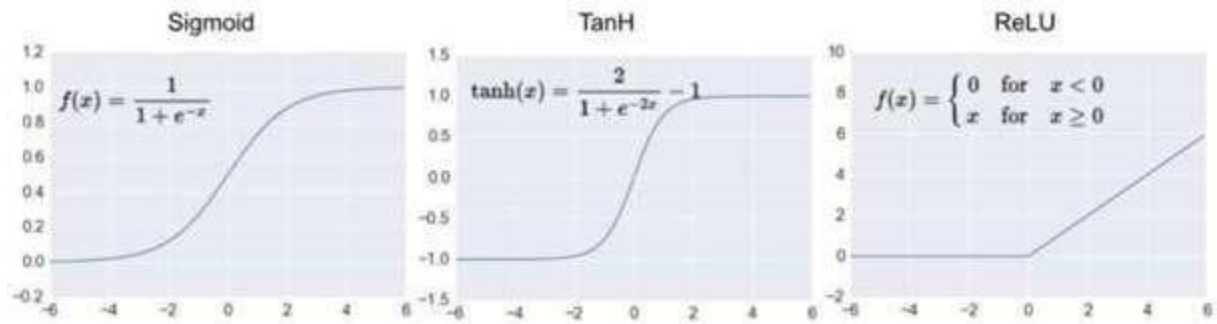


Figure 2.4: Common Activation Functions.

b) BIAS

Bias decides how high the weighted sum needs to be before the neuron activates. The bias value decides whether the activation function needs to be shifted to the left or to the right, to better fit the data.

Linearly separable functions can be implemented using the Single layer perceptron model.

In a such perceptron model, neurons are arranged in a single tier. It has only one input and output layer. To increase the efficiency or capability of perceptron we tend to add hidden layers in between the two existing layers. Thus forms the multilayer perceptron.

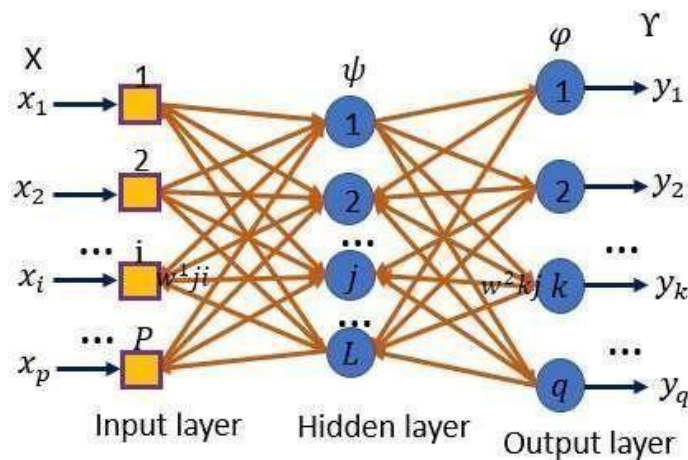


Figure 2.5: An Artificial Neural Network also known as a Multi-Layer Perceptron.

2.7. BACK PROPAGATION

Back propagation is the opposite of Forward propagation. In Perceptron learning no feedback is given from output layer to input layer which helps in minimizing the errors produced during the process. To overcome this concern, a back propagation learning algorithm is employed. Following the error calculation that is done in the first pass, the back propagation method performs backward pass repetatively and tries to discover the ideal weight values to reduce the error value. After generating the ideal weight, feedback isn't passed backward and the network moves forward.

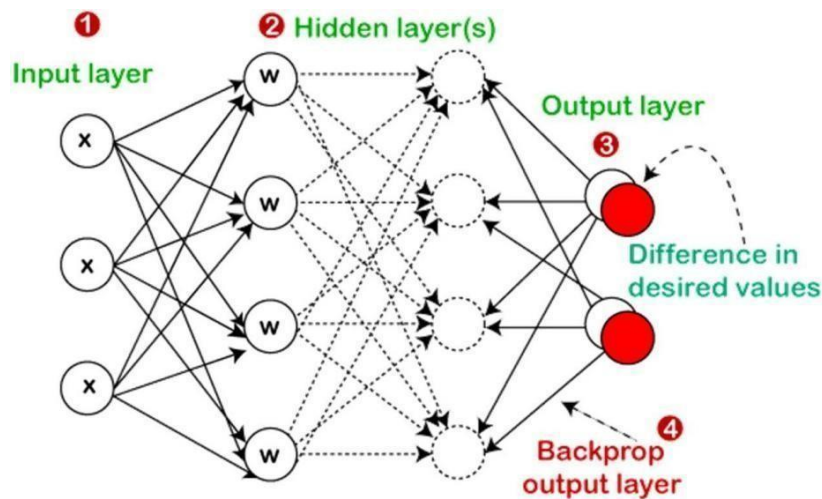


Figure 2.6: Back Propagation Learning.

2.8. FEED- FORWARD NEURAL NETWORK

Generally, the perceptron are arranged in layers, the layer that takes the inputs is the first layer and the last layer produces the outputs. The intermediate layers have no connection and does the processing work and are not visible outside. Thus, they form hidden layers. Each neuron in a tier is connected to other neurons in the immediate layer. Perceptron in the same layer is not connected. As all the neurons are connected, data from each layer is constantly "fed forward" to the next. So, these architectures are called feed- forward networks. Using these networks, we can implement non linearly separable functions also.

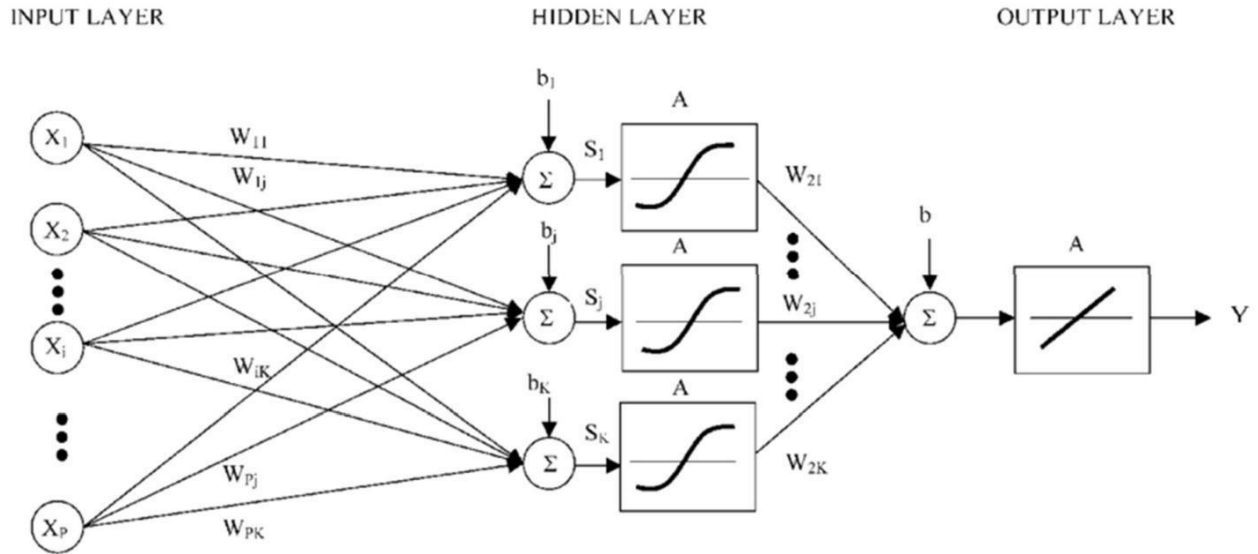


Figure 2.7: Feed-forward neural network with sigmoid activation function

These are the various learning approaches that are employed to teach computers how to learn on their own.

2.9. WHY DEEP LEARNING OVER MACHINE LEARNING

Though machine learning works very well for classifications or predictions, it tends to have very poor performance with images, audios, and other unstructured data types. Machine learning teaches computers to perform tasks without human intervention but when it comes to complex tasks like gathering data from an image or video the process of classification becomes difficult as features that are essential to make right classification are expected to be manually fed to the system. On the other hand, deep learning works on the concept of artificial neural networks where it takes the data connection between all the artificial neurons and adjusts the neurons according to data patterns. More neurons are added if the data is large, thus more features are expected to be learnt just like how humans tend to do.

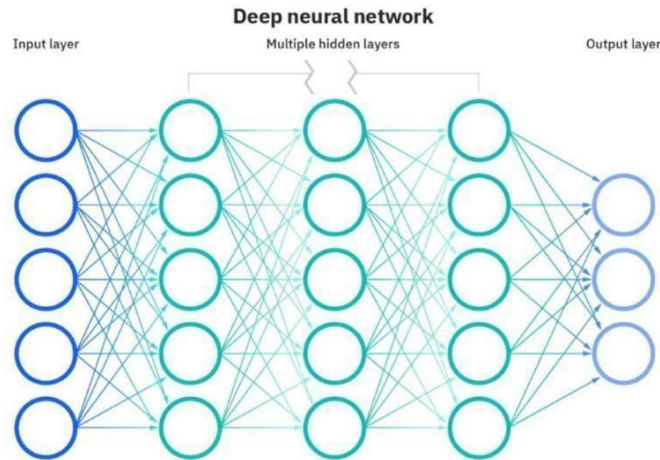


Figure 2.8: Pictorial Representation of Deep Learning Network.

It automatically learns at multiple levels and thereby allowing systems to learn underlying patterns in the dataset without depending on any specific algorithm. In case of deep learning the more the training data we feed the machine with, the more accurately the machine predicts the outputs.

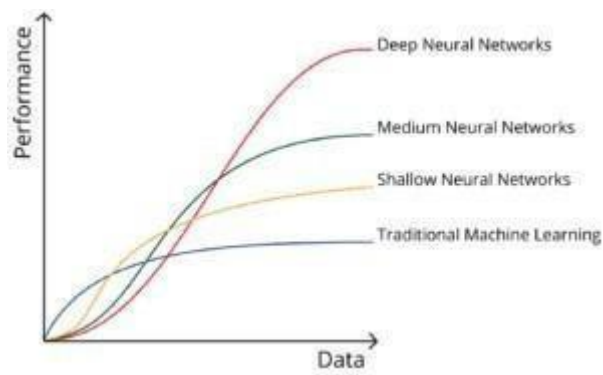


Figure 2.9: Neural Networks performance compared to Traditional Learning Algorithms

In our project we will be performing medical image analysis on the images of lung CT scans, analyse them and perform lung cancer detection. One particular deep learning model that has more significance when it comes to dealing with medical image analysis is the Convolutional Neural Networks.

2.10. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Networks are a class of Deep Neural Networks which are majorly used for image analysis and classification. CNNs are used to learn features of image inputs through the use of convolutional layers. CNNs are also used for computer vision tasks as they involve visual tasks i.e., Images and Video. CNNs are also majorly used for medical imaging as they tend to deal very well with CT scans, MRI scans and x-rays.

Convolution is a process where two or more images represented in matrix format and are multiplied. The obtained output is used to extract patterns from the input image.

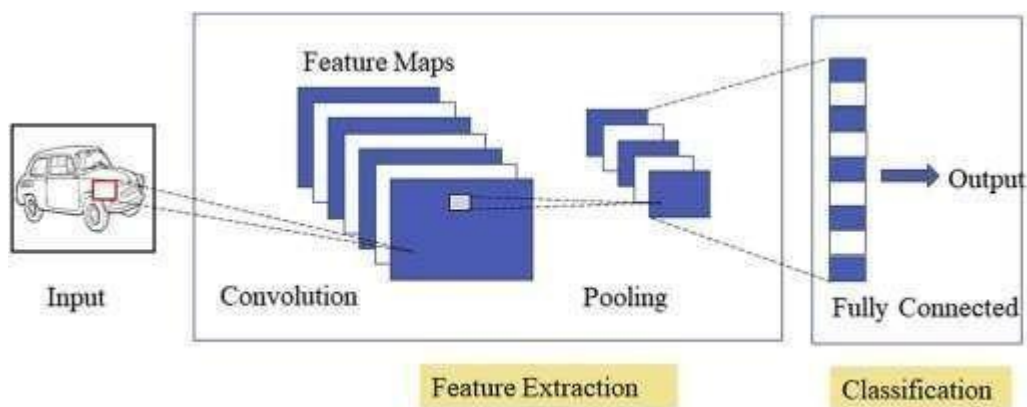


Figure 2.10: Picture depicting the different layers in Convolutional Neural Networks.

2.10.1. Different Layers of Convolutional Neural Network

A) CONVOLUTIONAL LAYER

A convolutional layer is the linear function employed in a Convolutional Neural Network. Here we use various image processing techniques. Using the suitable image processing feature detectors, each node in the hidden layer extracts various characteristics. A kernel is used to extract these traits. The original image is at the bottom, and the output of the convolutions is at the top. One of the advantages is that the convolutions' output reduces the original image's dimension.

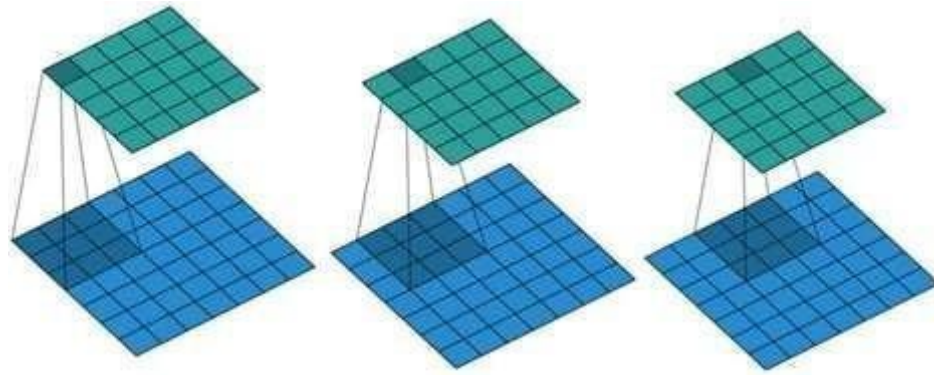


Figure 2.11: Stride Convolutions.

b) POOLING LAYER

After the convolutional layer, the output that is obtained from the convolutional layer is passed through the pooling layer. In pooling layer, we try to reduce the size of the representation to speed up the computation as well as make some of the features that is needed to be detected a bit more robust. There are different types of pooling, like max pooling and average pooling.

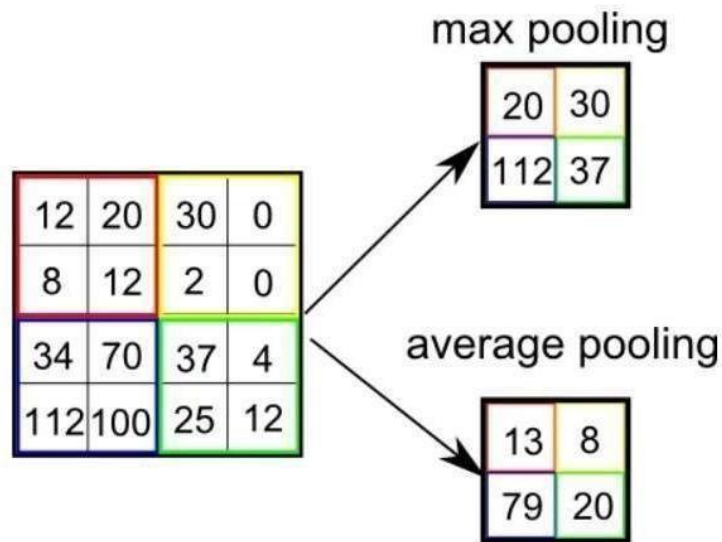


Figure 2.12: Figure showing Max Pooling and Average Pooling.

c) FULLY CONNECTED LAYER

Fully Connected Layer in CNN is an important part of CNN architecture. The purpose of fully connected layer is to classify the detected features into a category and also to learn to associate detected features to a particular label. Fully Connected Layer is just like an artificial Neural Network, where every neuron in it, is connected to every other neuron in the next layer and the previous layer.

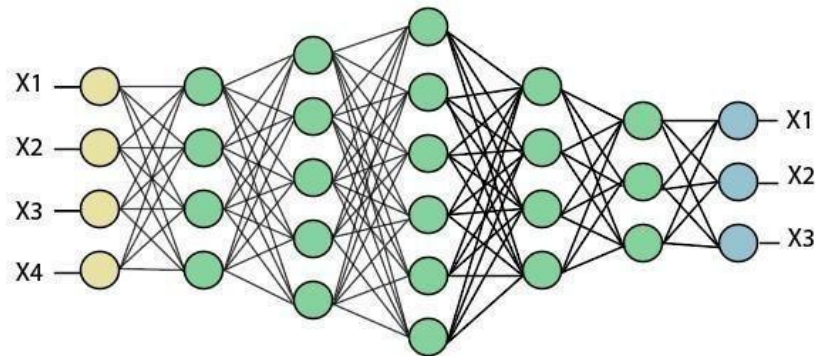


Figure 2.13: Figure showing Fully Connected Layers.

2.10.2. Working Of Convolutional Neural Networks

Firstly, in the Convolution Layer, we divide the image into parts by sliding the filter over the input image and extract the various features from the input images.

The output of this layer is termed as Feature map.

We then pass the feature MAP through the Soft max layer. Soft max is used usually for hidden layers; it avoids vanishing gradient problems by applying a suitable activation function.

Pass the output from the Soft max layer through the pooling layer, where we shrink our image.

A fully connected layer that utilizes the output from the convolution process, stacks up all the matrices and obtain a final vector and predicts the class of the image based on the features extracted in previous stages.

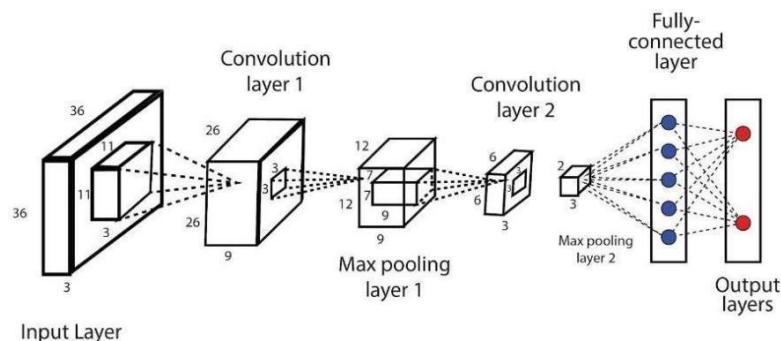


Figure 2.14: Picture depicting the different layers of Convolutional Neural Networks.

2.11. CANCER

Cancer is nothing but the uncontrollable development of abnormal cells in any part of the body. There are over 200+ types of cancers that are detected till date.

2.11.1. How Cancer Begins?

Cancer may start from any cell in the body. We have billions and trillions of cells in our body, which regenerated, reproduces and dies when aged and are replaced by new ones. But few unnecessary cells keep on multiplying and over powers the normal cells and when this process effects the wellbeing of our body it might lead to cancer.

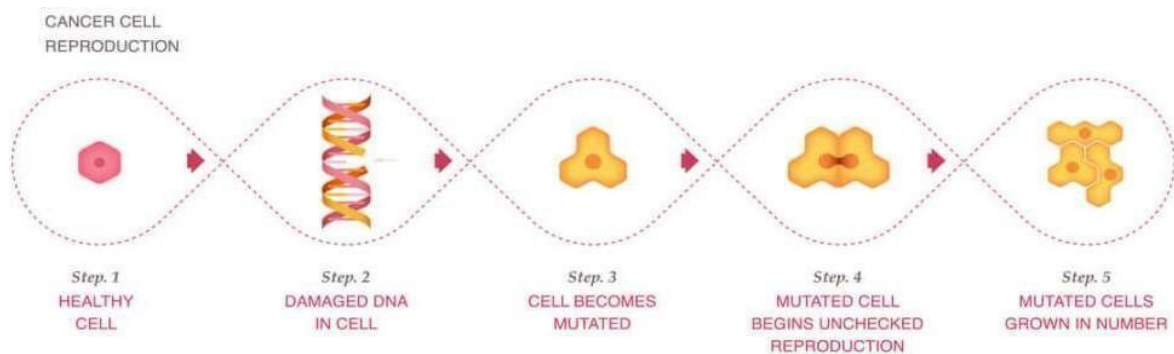


Figure 2.15: Picture depicting Cancer Cell Reproduction.

2.12. LUNG CANCER

Lung cancer is the kind of cancer that develops in the tissues and cells of the lungs. It is the deadliest diseases that is more probable to grow rapidly with the spread of metastasis. Metastasis is the formation of additional secondary malignant growths away from the primary cancer location.

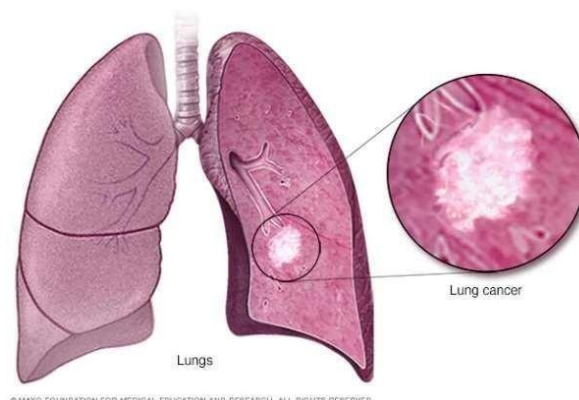


Figure 2.16: Picture depicting cancerous tumours in the lungs.

2.13. TYPES OF LUNGS CANCER

There are two main types of cancer.

- 1) Small cell lung cancer
- 2) Non- small cell lung cancer

Each of these types of cancer, are treated and handled different bases on the severity. The most frequent type of lung cancer is the non-small cell lung cancer.

2.13.1. Non- Small Cell Lung Cancer (NSCLC)

About 80% to 85% of the total lung cancers are non-small cell lung cancers There are various sub types of NSCLC such as Adenocarcinoma, squamous cell carcinoma, and large cell carcinoma.

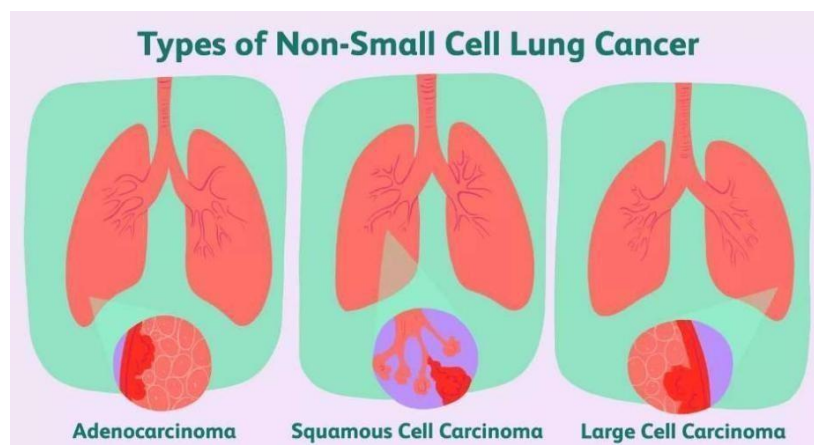


Figure 2.17: Picture showing types of Non-Small Cell Lung Cancer.

2.13.2. Small Cell Lung Cancer (SCLC)



Figure 2.18: Small cell lung cancer depiction

About 10% to 15% of the cancers are small cell lung cancer. SCLC tends to spread and affect other organs more quickly when compared to NSCLC. In most of the cases it becomes very difficult to diagnose this kind of cancer as the cancer might have spread to various other parts of the body hence making diagnosis and treatment very difficult.

Chemotherapy and radiation treatment work well for this cancer because it develops fast. But this kind of cancer, tends to attack people again even after treatment.

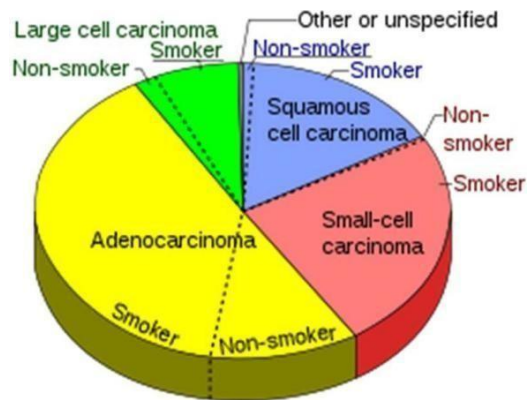


Figure 2.19: Incidences of NSCLC

2.13.3 Cancers That Spread to Lungs

Cancers that start from the lungs can occasional also spread within and beyond lungs. The spread of cancer can cause several problems and sometimes get out of hands and become very difficult to treat.

Sometimes the cancer can spread within the chest or even to other organs which makes it much difficult to notice the metastasis were cancer spreads rapidly to various cells or various other organs.

2.14. STAGES OF LUNG CANCER

- ❑ **Localized:** The cancer is present only in the lungs.
- ❑ **Regional:** The cancer has spread to the glands within the chest.
- ❑ **Distant:** The cancer has spread (or metastasized) to other parts of the body.

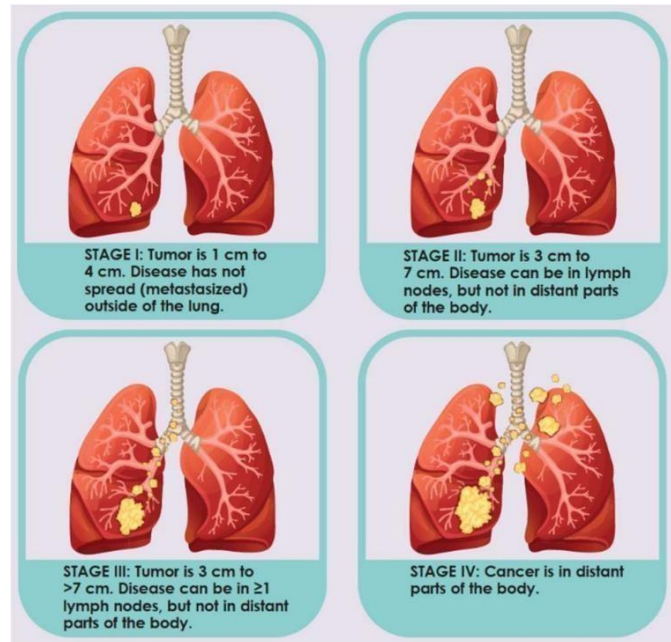


Figure 2.20: Stages of Lung Cancer.

2.15. TESTS TO DIAGNOSE LUNG CANCER

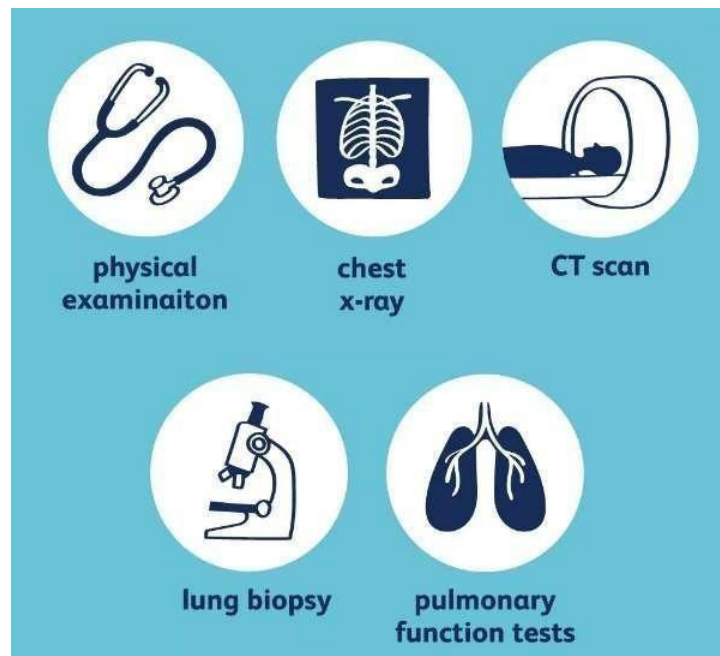


Figure 2.21: Lung Cancer Diagnosis.

2.16. CAUSES OF LUNG CANCER



Figure 2.22: Causes of Cancer.

2.17.1. Inherited Gene Changes

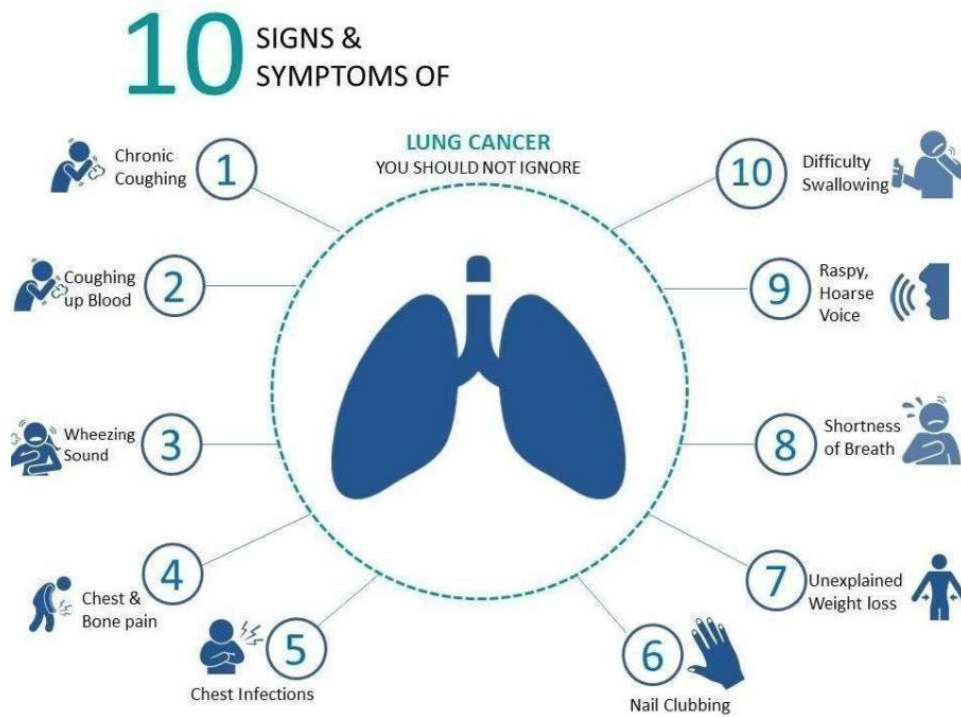
Some people tend to get lung cancer genetically due to the fact that they obtain their DNA mutations from their parents who might already have a greater risk of developing certain cancers.

2.17.2. Acquired Gene Changes

Acquired mutations in lung cells is often caused due to having some habits that might eventually lead to developing cancer like smoking, drinking etc. Sometimes there is a fair chance that a person can get cancer due to some abnormality in the cells mutation which out any intervention of outside factors, but happens naturally.

2.18. SIGNS AND SYMPTOMS OF LUNG CANCER

Figure 2.23: Symptoms of Lung Cancer.



2.19. SIDE EFFECTS OF LUNG CANCER

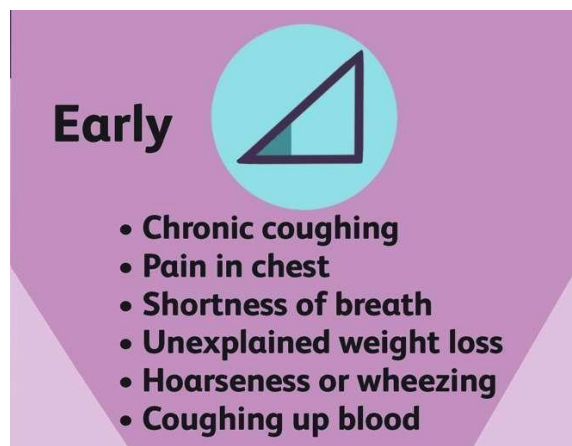


Figure 2.24: Side effects of lung cancer when detected early.

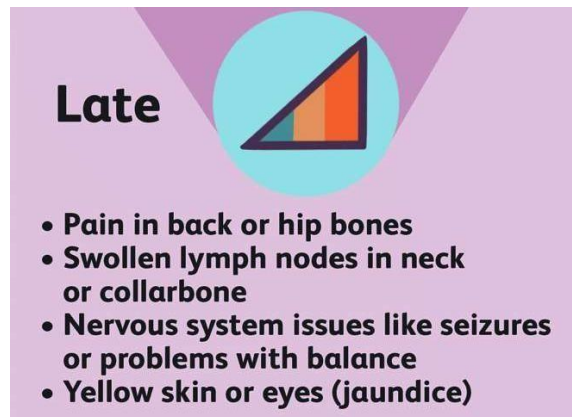


Figure 2.25: Side effects of lung cancer when detected late.

2.20. GENERAL LUNG CANCER STATISTICS

Lung cancer is one of the prime causes of fatality around the world. About two-thirds of all cancers related deaths accounts to lung cancer. There were 181 lakhs new cancer cases and 95 lakhs cancer-related deaths that was recorded worldwide in 2018. By 2040, it is expected that the number of cancer cases might rise upto an approximate of 295 lakhs, with 164 lakhs cancer-related mortalities.

Lung cancer is one the severe tumors that affects a person's health drastically in a very short span of time as it can easily spread to one part of the body to other part of the body which showing any severe symptoms in the early stages. Lung cancer tends to kill more people each year than breast, colon, and prostate cancer combined. It has been predicted to be one of the greatest single causes of mortality among the American population. In the United States, a projected 1,806,590 new instances of cancer were predicted to be identified, with 606,520 persons at risk of dying from the disease. According to a National Institutes of Health assessment of lung cancer cases and deaths from 2013 to 2017, the yearly rate of new cancer cases (cancer incidence) is

442.4 per 100,000 men and women, and the annual rate of cancer death (cancer mortality) is 158.3 per 100,000 men and women. According to the NIH, about 16,850 children and adolescents between the ages of 0 and 19 will be diagnosed with cancer in 2020, with 1,730 of them dying.

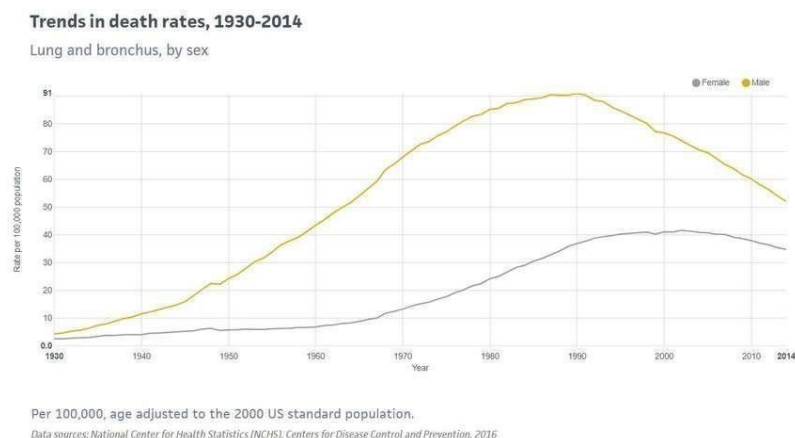


Figure 2.26: Death Rate Trend of Lung Cancer in the US.

2.21. OVERVIEW OF THE SCENARIO

Only 15% of lung nodules are detected in their early stages. Various methods are being followed to treat this disease like chemotherapy etc. However, lung cancer patients with various clinical stages have drastically varying prognoses. Patients in stage IA groups who have a survival rate of 5 years are more than 90%, while patients in stage IV who have a survival rate 5-years is fewer than 10%.

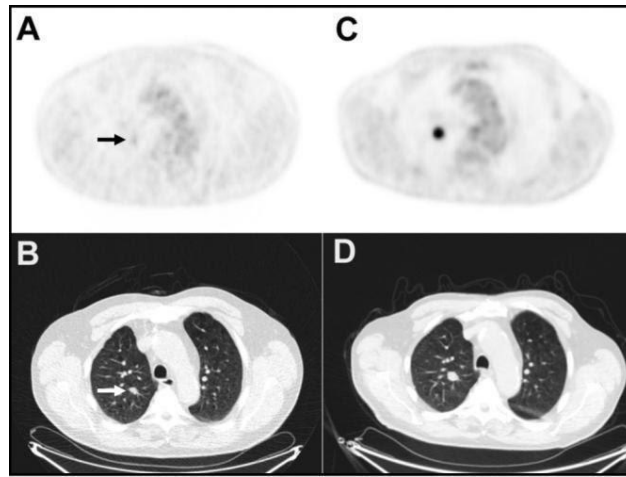


Figure 2.27: CT scans identified a lung nodule which, over 11 months progressed and was confirmed as lung cancer later.

However, the survival probability further falls to 3.5% when cancer tends to spread to different other organs. Thus, faster diagnosis of lung cancer is a critical step to provide improved chances of survival. Early detection of this cancer depends on how accurately the malignant nodules present in the lungs are detected in CT scans.

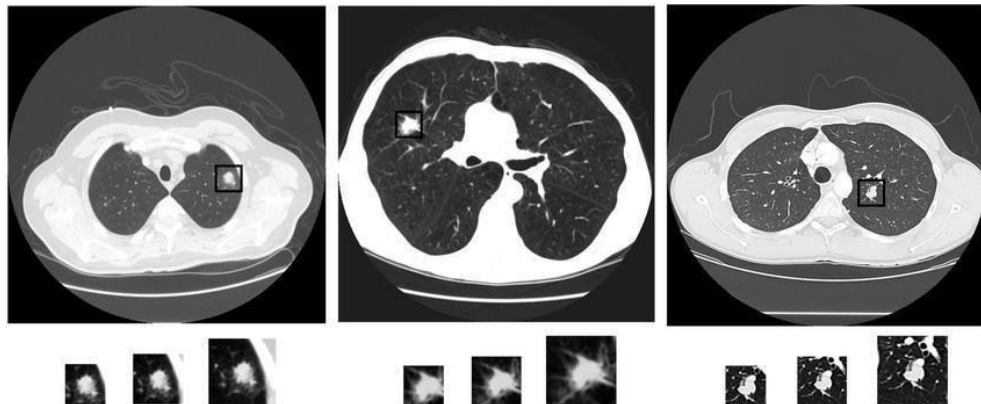


Figure 2.28: CT examples with lung nodules in different categories benign, primary malignant, and metastatic malignant.

A nodule is a "spot on the lung," which usually is an abnormal growth that forms in a lung that is seen on an CT scan. We may have one nodule on the lung or several nodules. This little round or oval solid overgrowth of tissue is surrounded by normal lung tissue. These nodules can either be benign or malignant. Nodules are very common. Not all nodules are malignant. About 95% of lung nodules are benign which form due to respiratory problems or other illnesses which do not require treatment. With increasing advancement in technology, the job of classifying lung nodules as benign or malignant automatically has become more feasible with the help of computer-aided design systems. Radiologists tend to use computer-aided detection mechanisms to detect the pulmonary lung nodules and to improve the accuracy in finding lesions that are tiny in size and are usually left undetected. Continuous effort is being taken by developers to design a well-performing pulmonary CAD system that detects malignant pulmonary lung nodules effectively and reduces the chances of obtaining false positives. As pulmonary nodules tend to have a variety of complex features such as tumor size, shapes, and calcification patterns, it becomes difficult for CAD systems to accurately identify the lesions and diagnose lung cancer. Thus, it is extremely essential to feed the convolutional neural network system with high-quality images to improve the accuracy in finding small malignant tumors.

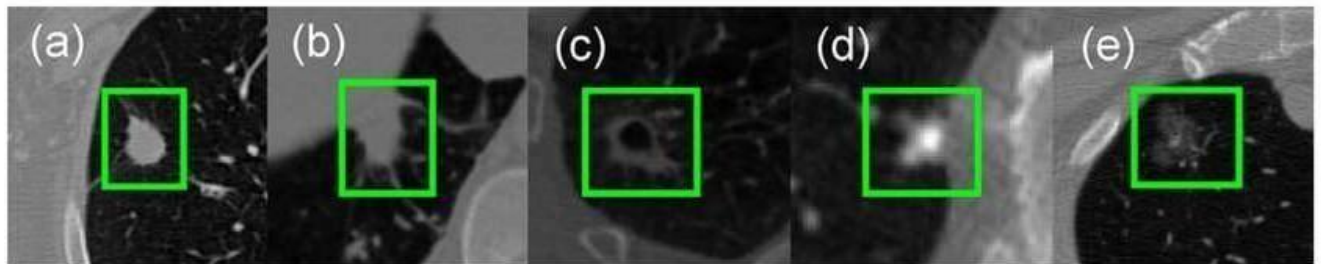


Figure 2.29: Example images of lung nodules with different locations and shapes in CT: (a) common isolated nodule. (b)Juxta pleural nodule (c) cavitary nodule. (d) calcific nodule. (e) ground-glass opacity (GGO) nodule.

2.22. DATASET

In this project we use the lung cancer screening thoracic computed tomography (CT) images from the IQ-OTHNCCD lung cancer dataset. The source of the dataset is Kaggle which is a data science and artificial intelligence Platform. The dataset contains 1190 images totally. These 1190 images are the CT scan slices of 110 cases. Each case approximately having 10 slices. These images are categorized into 3 classes: normal, benign, and malignant. Out of these, 40 cases are malignant cases; 15 cases are benign cases; and 55 cases are normal cases. In this project we try to build the Convolutional Neural Network to classify the images into one of the three classes. Primarily the nodules with a diameter less than 3mm

are considered as non-nodule and tiny nodule and are not taken into consideration since they have no clinical significance. and nodules with a diameter greater than or equal to 3mm were taken into consideration.

3. LITERATURE REVIEW

[1] “Multi-Resolution CNN and Knowledge Transfer for Candidate Classification in Lung Nodule Detection”, 32510 VOL. 7, 2019.

In this paper, to help in the identification of lung nodules, the authors propose a model that transfers and enhances a multi-resolution CNN for lung nodule candidate categorization through knowledge transfer. Small nodules with poor resolution and big nodules with high-resolution can both be identified using this approach. The methodology used in the research includes various steps namely rough candidate nodule finding and judgment. The results of the experiments performed by the authors during the course of this research suggest that it is possible to overcome the challenge posed by the wide range of sizes and forms of lung nodules, as well as diverseness in finding nodules using this approach.

[2] “Multi-Task Learning for Lung Nodule Classification on Chest CT”, VOLUME 8, 2020 180325.

This paper provides a unique multitask convolutional neural network (MT-CNN) architecture for distinguishing and finding the cancerous and non-cancerous nodules on Lung CT scans. To increase lung nodule classification performance, an image regeneration methodology is applied as an supplementary work from nine two-dimensional (2-D) images deconstructed from various angles of each nodule, this model learns three-dimensional (3-D) tumor segmentation characteristics. Each 2-D MT-CNN model includes two tasks: one for nodule classification (the primary job) and the other for the image reconstruction (the secondary task) (auxiliary task).

[3] Hybrid Segmentation Network for Small Cell Lung Cancer Segmentation”, 75591 VOLUME 7, 2019.

A Hybrid Segmentation Network (HSN) model is built which is a neural network that combines that incorporates lightweight 3D CNN for learning deep 3D structural and dimensional information with a 2D CNN for capturing fine-detailed semantic information of various slices of the CT scans in a single network.

In this HSN, spatiotemporal-separable 3D (S3D) convolutions are employed to deal with the complex dimensional features of CT scans and reduces the cost that is required for working with 3D CNN, Also, dilated convolutions in 2D CNN so as to memorize a plethora of semantic information about minor things. Moreover, to combine both 2D and 3D features effectively, a hybrid features fusion module is designed in this HSN network.

[4] “Multi-Branch Ensemble Learning Architecture Based on 3D CNN for False Positive Reduction in Lung Nodule Detection”, 67382 VOLUME 7, 2019.

The authors presented a Multi-Branch Ensemble Learning architecture based on three-dimensional (3D) convolutional neural networks (MBEL-3D-CNN) to handle the difficult task of reliably categorizing nodules in this study. Three fundamental concepts are combined in this method: The first step is developing a 3D-CNN to maximize the use of structural and dimensional features of lung lesions in 3D space; The second stage involves incorporating a MBEL-3D-CNN that is well suited to lung nodule diverseness and the final stage is to use ensemble learning to improve the 3D- CNN model's generalization performance. Furthermore, the authors employed offline hard mining techniques to enable the model to handle indistinguishable positive and negative samples.

4. PROBLEM STATEMENT

Lung cancer begins in the lung cells of a person. We have millions and billions of cells in our body that grow, divide and die throughout their lifetime. These cells usually die when they are of no use. But not always the process goes right, and sometimes that cells which needed to be perished continues producing new cells in our body and keeps multiplying abnormally. And these cells sometimes can be dangerous and overpower the normal cells. This is how cancer starts and spreads by means of metastasis.

It is very much important to diagnose and detect lung cancer at the early stages itself. Lung cancer is can be treated much easily and efficiently treated if its is diagnosed at an early stage, when it is tiny and has not spread. Otherwise, the chances of metastasis increases where the cancer cells tend to multiply and spread to various parts. But lung cancer is usually diagnosed at a late stage because lung cancer frequently has no signs until it has progressed and late diagnosis reduces survival chances and raises health-care expenses. To reduce the number of fatalities and enhance the likelihood of a successful therapy in order to treat lung cancer, it is critical to recognize lung cancer as soon as possible.

Accurate cancer segmentation aids doctors in better understanding the location and extent of cancer and in making more accurate diagnostic judgments. Manual segmentation of lung malignancies from enormous volumes of medical imaging, on the other hand, is a time-consuming and difficult process because it is exceedingly difficult to accurately describe nodule characteristics by manual feature engineering since nodule traits vary greatly including shape, texture and margin. To overcome this problem and to assist clinicians in diagnosing cancerous tumors by detecting the malignant lung nodules from computed tomography (CT) images, we utilize technology. Artificial intelligence can be used to locate malignant tumors. AI-assisted lung cancer screening might speed up and streamline the procedure, allowing more patients to be diagnosed earlier.

The goal of this project is to create a CAD system to help in the early detection of lung cancer.

In this project, we will create an application, a lung cancer detection system, to assist clinicians in making better and more informed judgments when treating lung cancer. This will aid in early identification of lung cancer, which will reduce the number of fatalities caused by tumor severity.

5. OBJECTIVES

- 1) The primary goal of this project is to develop a reliable lung nodule detection system that will aid in the faster and timely detection and diagnosis of lung cancer.
- 2) To create a system that reliably predicts cancer by accurately diagnosing tumors of all sizes, especially the small lesions that go undiagnosed most of the time, as well as spotting the tumor site in the lungs.
- 3) To enhance the accuracy of identifying the cancerous nodules in the lungs in clinical assessment with CT scans by utilizing maximum intensity projection images instead of using traditional images.
- 4) To implement efficient and accurately predicting deep neural network model using Convolutional Neural Networks.

6. EXISTING SYSTEM

When it comes to lung cancer diagnosis, the size and the form of a nodule are crucial signs of malignancy. A vast number of researchers have created networks of models for nodule detection in order to give benefits for early diagnosis. Many researchers have used traditional machine learning techniques, whilst a few also used deep learning approaches to design lung nodule detection systems. However, obtaining the structural information of the nodule from CT images in a CAD system is a difficult and challenging task with a high risk of losing little potential information. This is also one of the key problems with the current system.

In our existing system a novel lung nodule detection neural network model was proposed, which used multiple CT scan image views with various slab thicknesses. The study aimed to explore a variety of types of images that can be fed to the CNN model to enhance lung nodule identification.

The existing system which we have taken as reference has explored the feasibility of using different images to see what kinds of CT scan images could be fed into the CNN model to enhance lung nodule detection ability.

In the existing system, the malignant nodule is detected in two phases. The first is the identification of nodule candidates, and the second is the minimization of false positives.

The current system was designed to treat pulmonary nodules of all sorts. As the system was not built keeping in mind all the different types of nodules, such as subsolid, juxta-vascular or juxta-pleural nodules, the system performed moderately well on nodule detection with less precision. Because the existing system was fed with diverse types of images rather than traditional CT scan images, the system was able to detect nodules with fewer false positives.

Furthermore, the prior approach relied on 2-D CNNs to locate nodules inside its structure. Compared to 2D-CNN, 3D-CNN is expected to provide better results as it could extract more structural and dimensional information from with lesions better discriminating but they require more computing capacity and training time provided. Though 3D CNN requires more time and processing power than 2D CNN, it provides more accurate and correct results, which is critical for lung cancer detection since it concerns with the health of the person. In addition to this, the system's detection capability also needs improvement since it has been observed that the present system fails to identify tiny nodules and ground nodules, causing them to be disregarded during screening.

7. PROPOSED SYSTEM

7.1. UML DIAGRAMS

7.1.1. USE CASE 1: Abstract Of The Working Of The System

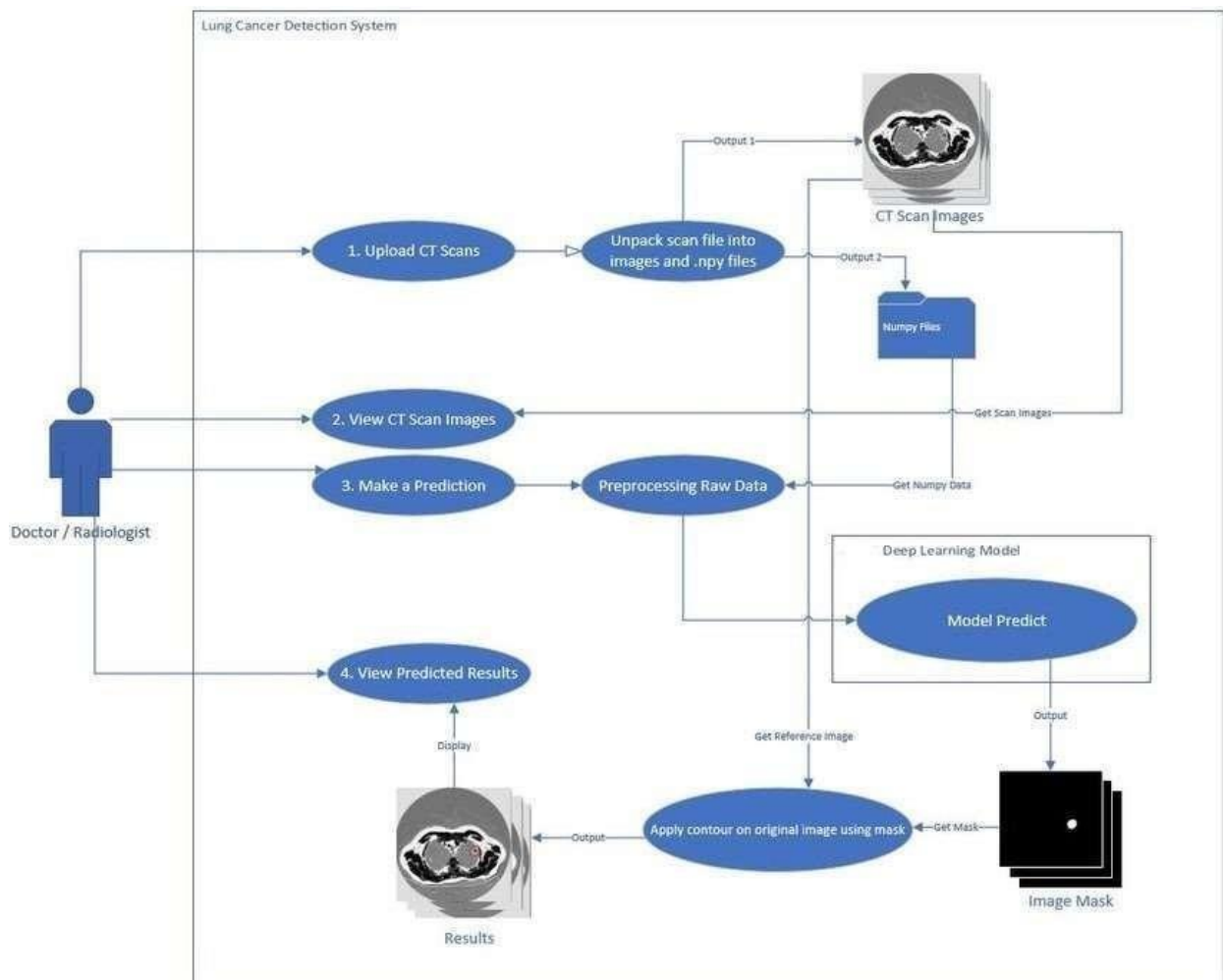


Figure 7.1: Use Case Diagram of the System

The technique for detecting lung cancer is depicted in this diagram. The user does a CT scan, feeds the CT scan to the machine as input and then reviews the findings of the scan as well as cancer diagnosis which the machine outputs.

7.1.2. USE CASE 2: User Inputting The Ct Scan (Test Input)

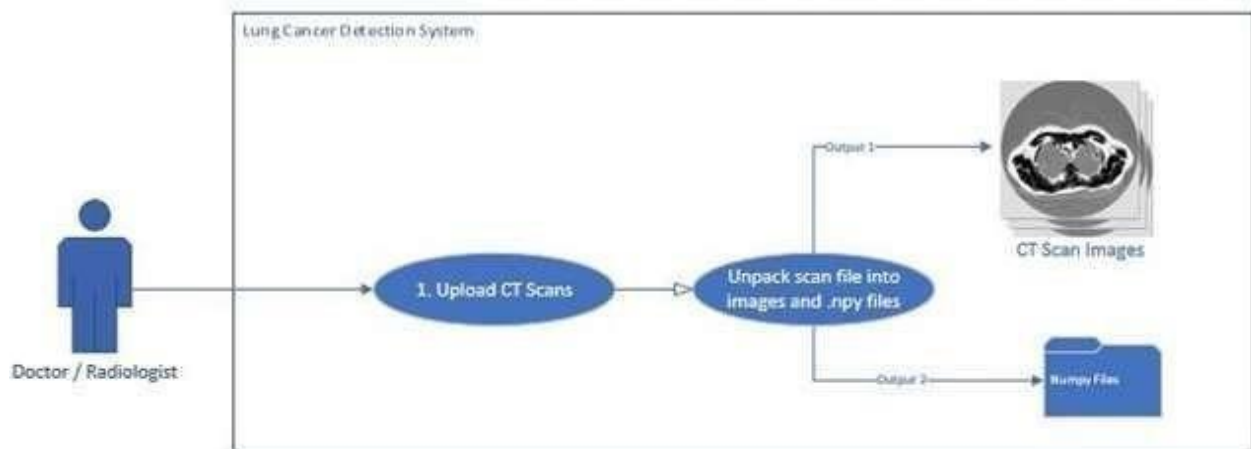


Figure 7.2: Feeding CT scans to the system

The system is fed with the image data of various CT scans. Later OpenCV and Numpy are used to save the inputted data as image files (.png) and image data arrays (.npy).

7.1.3. USE CASE-3: Making Predictions

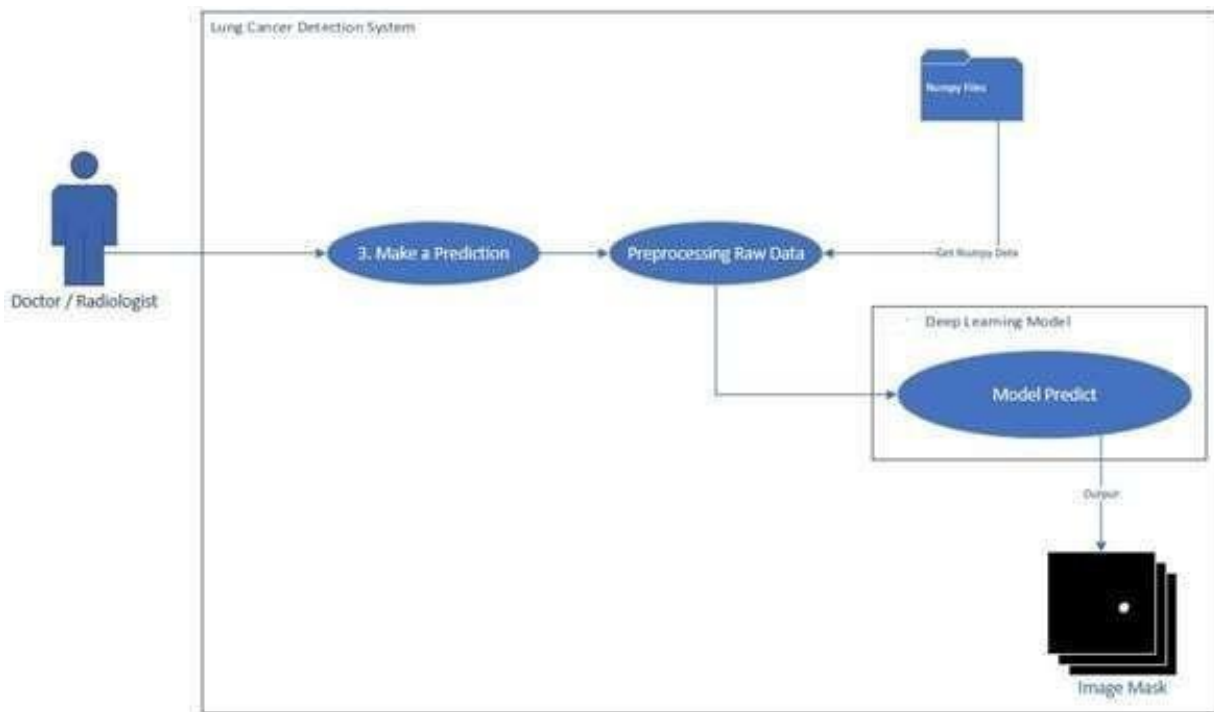


Figure 7.3: Use Case 3: Make Predictions

After obtaining the image files.png and image data array using OpenCV and NumPy, the raw image data arrays (NumPy files) are pre-processed. The processed data is then fed to the deep learning model which makes predictions and produces the output.

7.1.4. USE CASE-4: Viewing CT scans

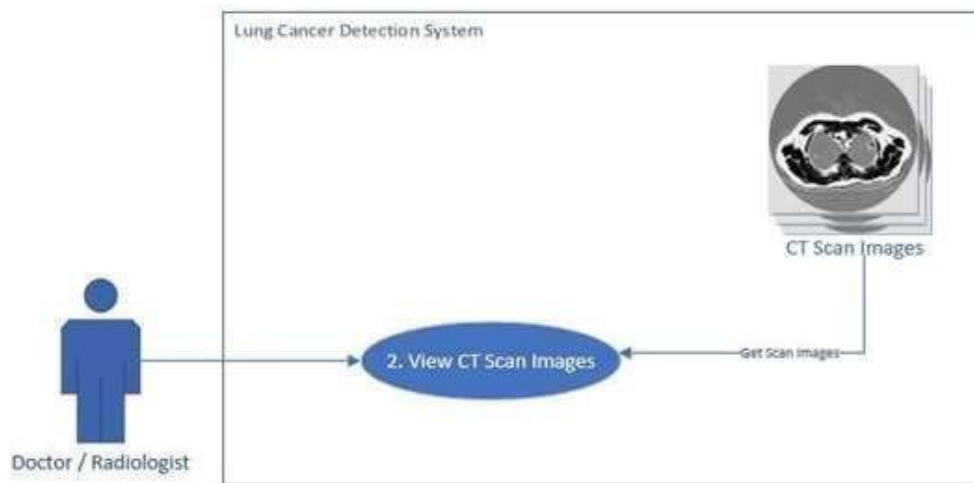


Figure 7.4: Viewing the CT scans

After extracting the image data using OpenCV from the metadata that is fed to the system, the png files of the CT scans can be made available for viewing. This can assist radiologists in determining the tumor location and size in the lungs.

7.1.5. USE CASE-5: Viewing Results

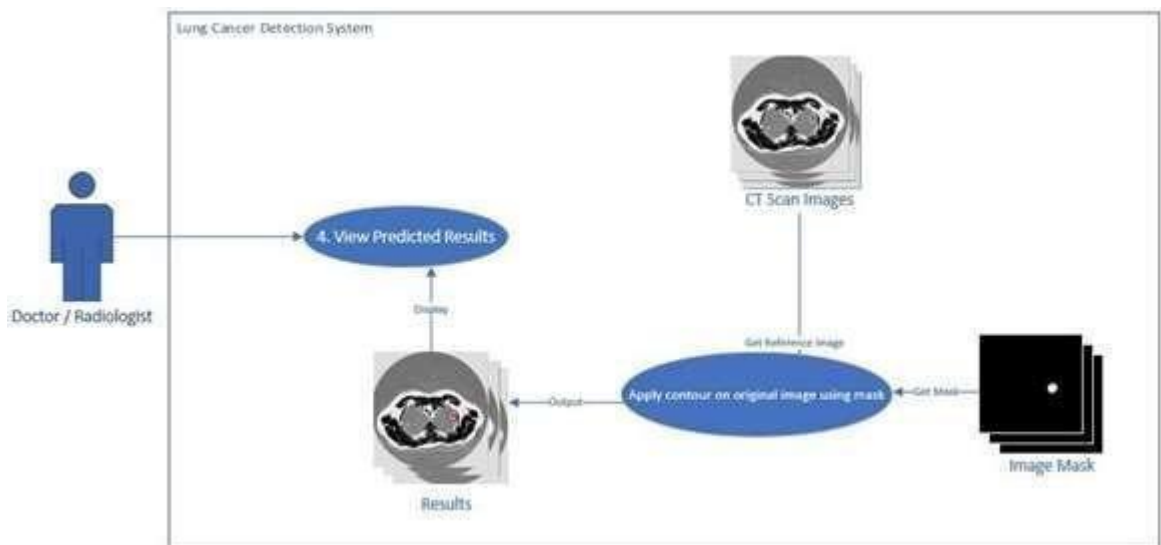


Figure 7.5: Use Case 5: View Predictions

The system considers the original CT scan image as reference and the associated mask and applies an image contour on the original image.

7.2. OVERVIEW

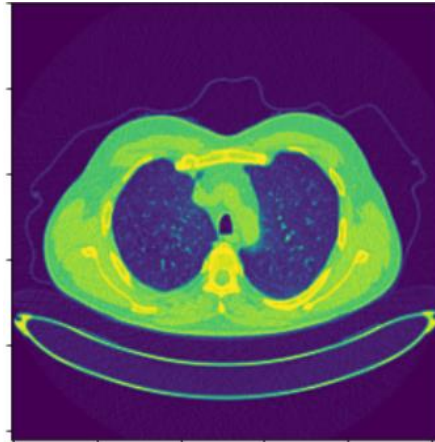


Figure 7.6: Picture depicting malignant lesion

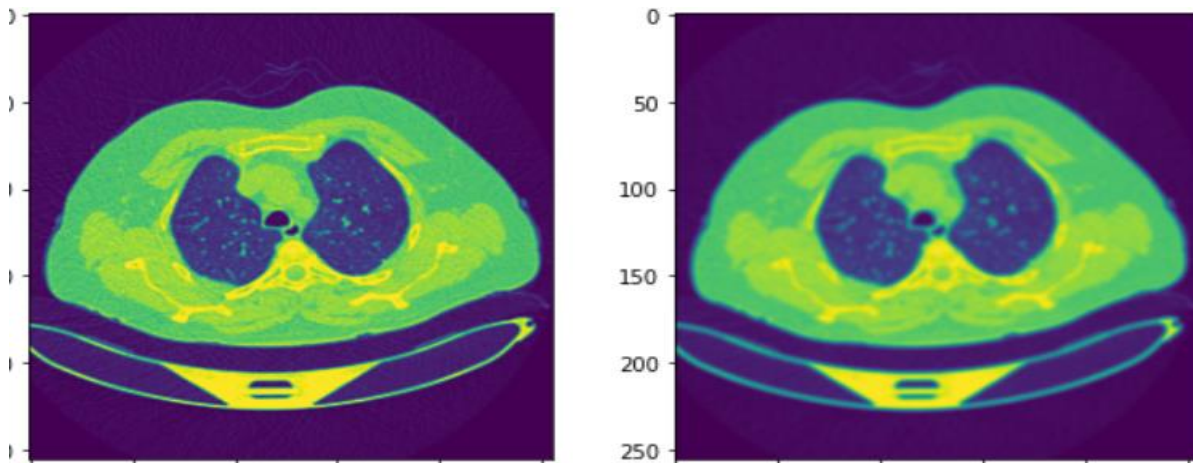


Figure 7.7: The top row images depict the original CT scan images of affected lungs, images in the bottom row depict the sliced pulmonary nodules.

7.3. GLIMPSE OF THE WORKING METHODOLOGY

The working methodology of our proposed system begins with taking the CT scans of various cases from the IQ-OTHNCCD lung cancer dataset. Then we segment the lung parenchyma and slice the radiographs using image pre-processing methods to obtain the sliced image of the lesion which is fed to the convolutional neural network as an input image. The input image then goes through the different layers of the CNNs.

Input -> Convolution -> SoftMax -> Convolution -> SoftMax -> Pooling -> SoftMax -> Pooling -> Fully C connected Layer

The CNN system extracts the potential features while the model is being trained on a set of a variety of images from the IQ-OTHNCCD lung cancer dataset. Later when a new test input is given to the CNN model, it compares learned features with the input data and classifies the inputted sliced tumor to be either normal or benign or malignant.

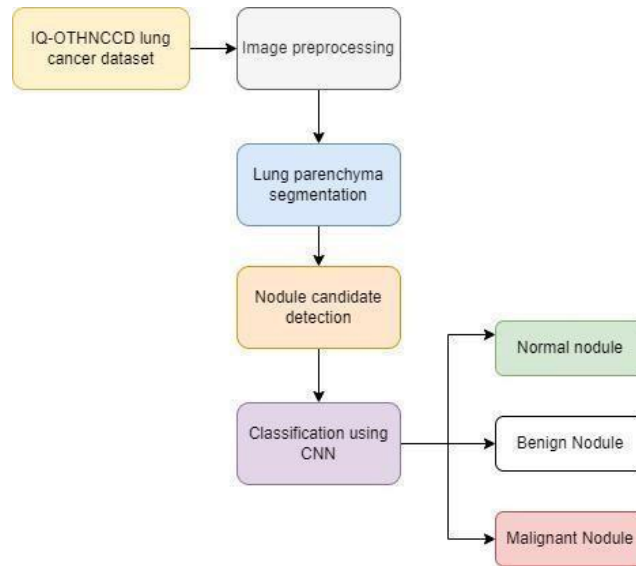


Figure 7.8: The pipeline of working methodology

7.4. PROPOSED SYSTEM ARCHITECTURE

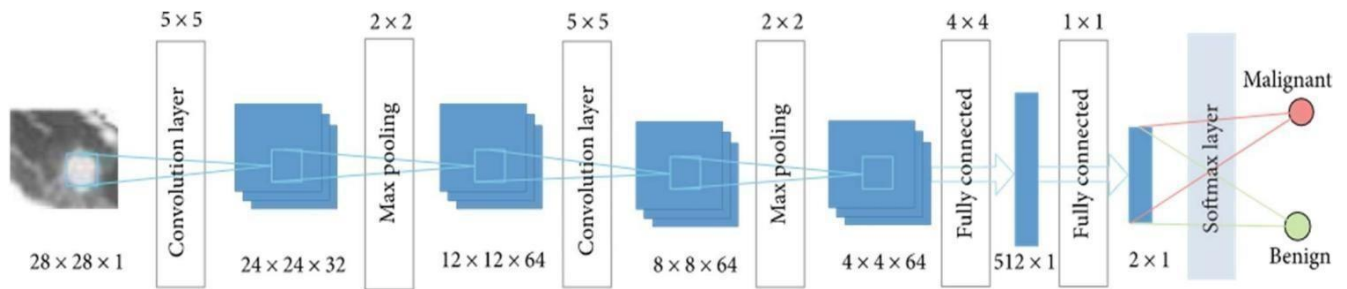


Figure 7.9: The architecture of the CNN to classify sliced CT scan input images as malignant or benign

7.5. PROPOSED SYSTEM METHODOLOGY

Our lung cancer detection system is firstly fed with the IQ-OTHNCCD lung cancer dataset that contains computed tomography scans to facilitate computer- aided systems on the assessment of lung nodule detection, classification and quantification.

We take the radiography images provided in the dataset and perform image processing on it to obtain the sliced image of the pulmonary lung nodule. Image processing is a method to perform operations on an image to extract information from it or enhance it. Here the essential information is the image of the tumor in the lungs which is given as input to our convolutional neural network model.

Alongside performing image processing, we also perform image enhancement where we play suitable filters to the input image to remove unnecessary noises so as to prevent misleading results that may occur in subsequent processes.

Then we apply OpenCV and NumPy functions on the input and separate the .png image files and image data arrays.

The image data arrays obtained are then pre-processed and are made suitable for the use of classification.

The pre-processed data is then fed to the Deep Learning Model.

The deep learning model in our project contains a number of convolutional, RELU and pooling layers.

The sliced image of the pulmonary lung nodule which is fed to our deep learning model goes through many layers of the neural network where convolution takes place.

A convolution is the preliminary process where we apply a suitable filter to an input to produce an activation. When the same filter is repetitively applied to an input, we obtain a feature map, which displays the various supporting and contrasting feature in an input, in our case the feature is the nodule which is either malignant or benign.

The output from the convolutional layers reflects high-level characteristics in the input after going through a sequence of recurrent convolution and pooling layers.

This output is then flattened and a vector matrix is obtained and connected to the output layer by adding a FC layer. Fully Connected layers are the final layers of the network.

The input to the fully connected layer is the output from the final Pooling or Convolutional Layer, which is flattened and then fed into the fully connected layer.

Flattening is the technique of converting an N-dimensional matrix into a vector by unrolling all of the values.

After entering through the FC layers, the final layer uses the soft-max activation function to determine the how likely the input data belongs to either malignant class or benign class (classification).

These results along with the cancerous lesion spots are displayed to clinicians in order to discover malignant cells, which aids in the treatment of patients.

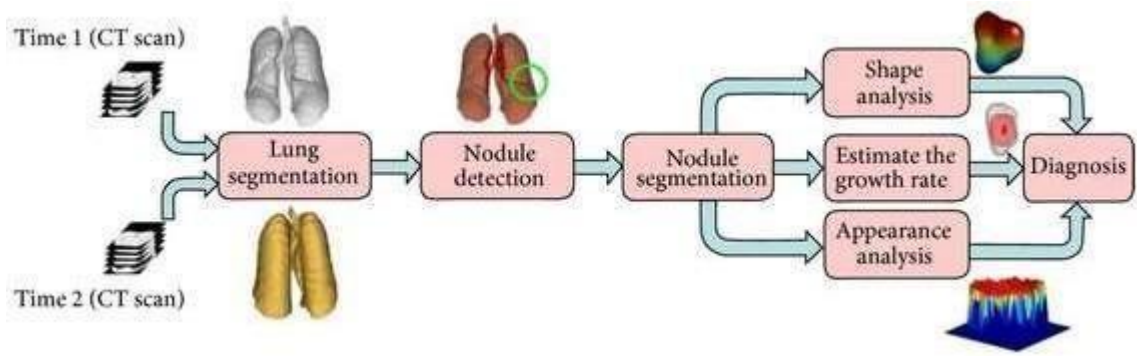


Figure 7.10: Processes that are involved in building a CAD system for lung nodule detection.

8. OVERVIEW OF TECHNOLOGIES

8.1 Python:

Python is an interpreted high-level general-purpose programming language. With its use of significant indentation, its design philosophy emphasizes code readability. Its language constructs and object-oriented approach are intended to assist programmers in writing clear, logical code for small and large-scale projects.

Python is garbage-collected and dynamically typed. It supports a wide range of programming paradigms, including structured (especially procedural), object-oriented, and functional programming. Because of its extensive standard library, it is frequently referred to as a "batteries included" language.

8.2 Machine learning:

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

8.3. Deep Learning

Deep learning is a sub field of machine learning that deals with artificial neural networks, which are algorithms inspired by the structure and function of the brain.

ML and DL aids AI by providing a set of algorithms and neural networks to solve data driven problems.

8.4. Convolutional Neural Network

A convolutional neural network (CNN or Convent), is a network architecture for deep learning which learns directly from data, and from experiences and does not require any manual feature extraction for the data. CNNs are majorly used for finding patterns in images to recognize objects, faces, and scenes.

Factors for using CNNs for deep learning

- CNNs eliminate the need for manual feature extraction—the features are learned directly by the CNN.

- CNNs produce highly accurate recognition results.

- CNNs can be retrained for new recognition tasks, enabling you to build on pre-existing networks. CNNs provide an optimal architecture for uncovering and learning key features in image and time-series data which is a main reason for which CNNs are considered as a key technology in applications such as **Medical Imaging** as they can examine thousands of pathology reports to visually detect the presence or absence of cancer cells in images.

9. IMPLEMENTATION

9.1. Coding

```
[1] !pip install tensorflow_addons


Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting tensorflow_addons
  Downloading tensorflow_addons-0.18.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
    | 1.1 MB 5.1 MB/s
Requirement already satisfied: typeguard>=2.7 in /usr/local/lib/python3.7/dist-packages (from tensorflow_addons) (2.7.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from tensorflow_addons) (21.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->tensorflow_addons) (3.0.9)
Installing collected packages: tensorflow-addons
Successfully installed tensorflow-addons-0.18.0
```

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
        name=fn, length=len(uploaded[fn])))


# Then move kaggle.json into the folder where the API expects to find it.
!mkdir -p ~/.kaggle/ && mv kaggle.json ~/.kaggle/ && chmod 600 ~/.kaggle/kaggle.json
```

 **kaggle.json**(application/json) - 74 bytes, last modified: n/a - 100% done
Saving kaggle.json to kaggle.json

```
[3] !kaggle datasets download -d adityamahimkar/iqothnccd-lung-cancer-dataset

Downloading iqothnccd-lung-cancer-dataset.zip to /content
 92% 183M/199M [00:01<00:00, 160MB/s]
100% 199M/199M [00:01<00:00, 166MB/s]
```

```
from zipfile import ZipFile
file_name = "/content/iqothnccd-lung-cancer-dataset.zip"
with ZipFile(file_name,'r') as zip:
    zip.extractall()
    print('Done')
```

 Done

```
[5] import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from PIL import Image
import seaborn as sns
import cv2
import random
import os
import imageio
import plotly.graph_objects as go
```

```

import plotly.graph_objects as go
import plotly.express as px
import plotly.figure_factory as ff
from plotly.subplots import make_subplots
from collections import Counter

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.neighbors import LocalOutlierFactor
from sklearn.metrics import accuracy_score, recall_score, precision_score, classification_report, confusion_matrix, plot_confusion_matrix
from sklearn.model_selection import RandomizedSearchCV, cross_val_score, RepeatedStratifiedKFold
from imblearn.over_sampling import SMOTE

import tensorflow as tf
import tensorflow_addons as tfa
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D, GlobalAveragePooling2D, BatchNormalization
from keras.applications import resnet
from tensorflow.keras.applications import EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, EfficientNetB4, EfficientNetB5,
from keras.applications.resnet import ResNet50
from keras_preprocessing.image import ImageDataGenerator, load_img, img_to_array, array_to_img

```

```

[6] directory = r'../content/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung cancer dataset'
categories = ['Benign cases', 'Malignant cases', 'Normal cases']

```

Image Size Variations

```

size_data = {}
for i in categories:
    path = os.path.join(directory, i)
    class_num = categories.index(i)
    temp_dict = {}
    for file in os.listdir(path):
        filepath = os.path.join(path, file)
        height, width, channels = imageio.imread(filepath).shape
        if str(height) + ' x ' + str(width) in temp_dict:
            temp_dict[str(height) + ' x ' + str(width)] += 1
        else:
            temp_dict[str(height) + ' x ' + str(width)] = 1

    size_data[i] = temp_dict

size_data

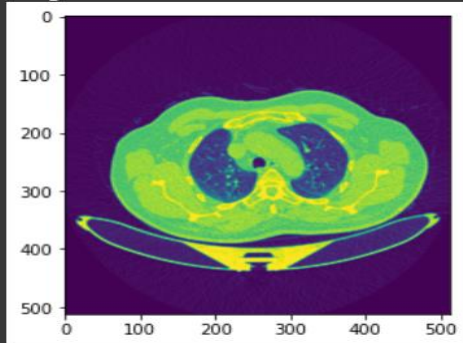
```

```

▶ for i in categories:
    path = os.path.join(directory, i)
    class_num = categories.index(i)
    for file in os.listdir(path):
        filepath = os.path.join(path, file)
        print(i)
        img = cv2.imread(filepath, 0)
        plt.imshow(img)
        plt.show()
        break

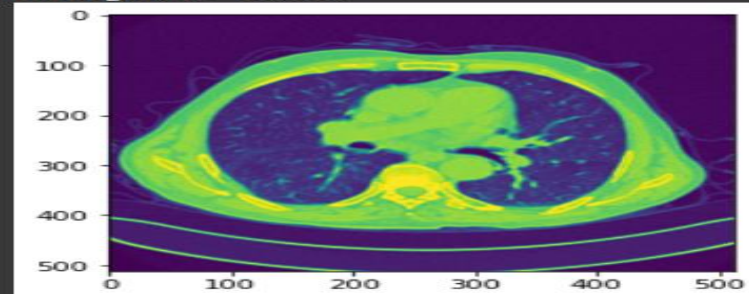
```

↳ Benign cases



Malignant cases

Malignant cases



Normal cases

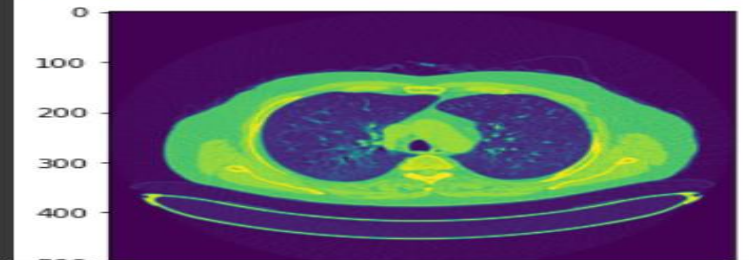


Image Preprocessing and Testing

```
img_size = 256
for i in categories:
    cnt, samples = 0, 3
    fig, ax = plt.subplots(samples, 3, figsize=(15, 15))
    fig.suptitle(i)

    path = os.path.join(directory, i)
    class_num = categories.index(i)
    for curr_cnt, file in enumerate(os.listdir(path)):
        filepath = os.path.join(path, file)
        img = cv2.imread(filepath, 0)

        img0 = cv2.resize(img, (img_size, img_size))

        img1 = cv2.GaussianBlur(img0, (5, 5), 0)

        ax[cnt, 0].imshow(img)
        ax[cnt, 1].imshow(img0)
        ax[cnt, 2].imshow(img1)
        cnt += 1
    if cnt == samples:
        break
```

Preparing Data

```
data = []
img_size = 256

for i in categories:
    path = os.path.join(directory, i)
    class_num = categories.index(i)
    for file in os.listdir(path):
        filepath = os.path.join(path, file)
        img = cv2.imread(filepath, 0)
        # preprocess here
        img = cv2.resize(img, (img_size, img_size))
        data.append([img, class_num])

random.shuffle(data)

X, y = [], []
for feature, label in data:
    X.append(feature)
    y.append(label)

print('X length:', len(X))
print('y counts:', Counter(y))
```

Model-1:

Applying SMOTE to oversample the data

```
print(Counter(y_train), Counter(y_valid))

Counter({1: 420, 2: 312, 0: 90}) Counter({1: 141, 2: 104, 0: 30})

[ ] print(len(X_train), X_train.shape)

X_train = X_train.reshape(X_train.shape[0], img_size*img_size*1)

print(len(X_train), X_train.shape)

822 (822, 256, 256, 1)
822 (822, 65536)

print('Before SMOTE:', Counter(y_train))
smote = SMOTE()
X_train_sampled, y_train_sampled = smote.fit_resample(X_train, y_train)
print('After SMOTE:', Counter(y_train_sampled))

Before SMOTE: Counter({1: 420, 2: 312, 0: 90})
After SMOTE: Counter({2: 420, 1: 420, 0: 420})
```

```
[ ] X_train = X_train.reshape(X_train.shape[0], img_size, img_size, 1)
X_train_sampled = X_train_sampled.reshape(X_train_sampled.shape[0], img_size, img_size, 1)

print(len(X_train), X_train.shape)
print(len(X_train_sampled), X_train_sampled.shape)

822 (822, 256, 256, 1)
1260 (1260, 256, 256, 1)
```

Model Building with SMOTE data

```
model1 = Sequential()

model1.add(Conv2D(64, (3, 3), input_shape=X_train.shape[1:]))
model1.add(Activation('relu'))
model1.add(MaxPooling2D(pool_size=(2, 2)))

model1.add(Conv2D(64, (3, 3), activation='relu'))
model1.add(MaxPooling2D(pool_size=(2, 2)))

model1.add(Flatten())
model1.add(Dense(16))
```

```
model1.add(Dense(3, activation='softmax'))

model1.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 64)	640
activation (Activation)	(None, 254, 254, 64)	0
max_pooling2d (MaxPooling2D)	(None, 127, 127, 64)	0
conv2d_1 (Conv2D)	(None, 125, 125, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 64)	0
flatten (Flatten)	(None, 246016)	0
dense (Dense)	(None, 16)	3936272
dense_1 (Dense)	(None, 3)	51

=====
Total params: 3,973,891
Trainable params: 3,973,891

```
[29] history = model1.fit(X_train_sampled, y_train_sampled, batch_size=8, epochs=10, validation_data=(X_valid, y_valid))
```

Epoch 1/10
158/158 [=====] - 188s 1s/step - loss: 0.5290 - accuracy: 0.8206 - val_loss: 0.0851 - val_accuracy: 0.9709
Epoch 2/10
158/158 [=====] - 172s 1s/step - loss: 0.0278 - accuracy: 0.9937 - val_loss: 0.0285 - val_accuracy: 0.9891
Epoch 3/10
158/158 [=====] - 174s 1s/step - loss: 0.0246 - accuracy: 0.9976 - val_loss: 0.0435 - val_accuracy: 0.9927
Epoch 4/10
158/158 [=====] - 173s 1s/step - loss: 0.0388 - accuracy: 0.9937 - val_loss: 0.0339 - val_accuracy: 0.9927
Epoch 5/10
158/158 [=====] - 173s 1s/step - loss: 0.0222 - accuracy: 0.9952 - val_loss: 0.0262 - val_accuracy: 0.9891
Epoch 6/10
158/158 [=====] - 170s 1s/step - loss: 0.0269 - accuracy: 0.9944 - val_loss: 0.0235 - val_accuracy: 0.9891
Epoch 7/10
158/158 [=====] - 171s 1s/step - loss: 0.0474 - accuracy: 0.9873 - val_loss: 0.0071 - val_accuracy: 1.0000
Epoch 8/10
158/158 [=====] - 170s 1s/step - loss: 0.0141 - accuracy: 0.9952 - val_loss: 0.0220 - val_accuracy: 0.9927
Epoch 9/10
158/158 [=====] - 169s 1s/step - loss: 0.0107 - accuracy: 0.9968 - val_loss: 0.0622 - val_accuracy: 0.9927
Epoch 10/10
158/158 [=====] - 168s 1s/step - loss: 0.0066 - accuracy: 0.9968 - val_loss: 0.0410 - val_accuracy: 0.9855

Results

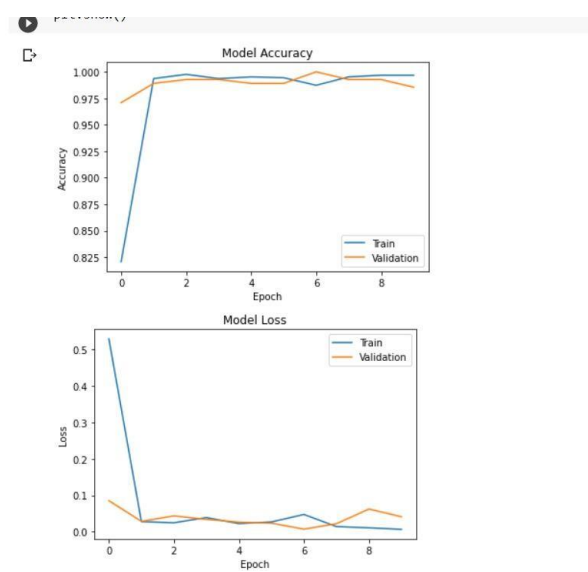
```
y_pred = model1.predict(X_valid, verbose=1)
y_pred_bool = np.argmax(y_pred, axis=1)

print(classification_report(y_valid, y_pred_bool))

print(confusion_matrix(y_true=y_valid, y_pred=y_pred_bool))
```

9/9 [=====] - 10s 1s/step

	precision	recall	f1-score	support
0	0.96	0.90	0.93	30
1	0.98	0.99	0.98	141
2	0.95	0.96	0.96	104
accuracy			0.97	275



```
path = "/content/Test cases/000020_03_01_212.png"
img = load_img(path, target_size = (224,224))
input_arr = img_to_array(img)/255

plt.imshow(input_arr)
plt.show()

input_arr.shape

input_arr = np.expand_dims(input_arr, axis = 0)

#pred = model1.predict_classes(input_arr)[0][0][0]
pred = model1.predict
np.argmax(categories[0])
if np.argmax(categories[0]) == 0:
    print("The following CT Image is Benign")
    print("It's symptoms are:\n1)Mild Cough\n2)Shortness of breath\n3)Coughing up blood")
elif np.argmax(categories[0]) == 1:
    print("The following CT Image is Malignant")
    print("It's symptoms are:\n1)Severe Cough\n2)Shortness of breath\n3)Coughing up blood\n4)Chest pain\n5)Hoarseness\n6)Losing much weight\n7)Bone pain\n8)Continu
else:
    print("The following CT Image is Normal")
    print("You have no lung cancer, but its good to take precautions like\n1)Quit smoking (if you smoke)\n2)Eat healthy food in a balanced diet\n3)Exercise regularl
```

Result:



Model 2: Model Building with Class Weighted Approach

```
model2 = Sequential()

model2.add(Conv2D(64, (3, 3), input_shape=X_train.shape[1:]))
model2.add(Activation('relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Conv2D(64, (3, 3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))

model2.add(Flatten())
model2.add(Dense(16))
model2.add(Dense(3, activation='softmax'))

model2.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 254, 254, 64)	640
activation_1 (Activation)	(None, 254, 254, 64)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 127, 127, 64)	0
conv2d_3 (Conv2D)	(None, 125, 125, 64)	36928

```
conv2d_3 (Conv2D) (None, 125, 125, 64) 36928
```

max_pooling2d_3 (MaxPooling 2D) (None, 62, 62, 64) 0

```
flatten_1 (Flatten) (None, 246016) 0
```

```
dense_2 (Dense) (None, 16) 3936272
```

```
dense_3 (Dense) (None, 3) 51
```

```
=====  
Total params: 3,973,891  
Trainable params: 3,973,891  
Non-trainable params: 0  
=====  
[ ] model2.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
new_weights = {  
    0: X_train.shape[0]/(3*Counter(y_train)[0]),  
    1: X_train.shape[0]/(3*Counter(y_train)[1]),  
    2: X_train.shape[0]/(3*Counter(y_train)[2]),  
}  
  
# new_weights[0] = 0.5
```

```

history = model2.fit(X_train, y_train, batch_size=8, epochs=10, validation_data=(X_valid, y_valid), class_weight=new_weights)

Epoch 1/10
103/103 [=====] - 115s 1s/step - loss: 0.8706 - accuracy: 0.6667 - val_loss: 0.2133 - val_accuracy: 0.9345
Epoch 2/10
103/103 [=====] - 118s 1s/step - loss: 0.1616 - accuracy: 0.9586 - val_loss: 0.1422 - val_accuracy: 0.9818
Epoch 3/10
103/103 [=====] - 115s 1s/step - loss: 0.0383 - accuracy: 0.9939 - val_loss: 0.0165 - val_accuracy: 0.9927
Epoch 4/10
103/103 [=====] - 118s 1s/step - loss: 0.0763 - accuracy: 0.9866 - val_loss: 0.0838 - val_accuracy: 0.9855
Epoch 5/10
103/103 [=====] - 114s 1s/step - loss: 0.0250 - accuracy: 0.9951 - val_loss: 0.0941 - val_accuracy: 0.9891
Epoch 6/10
103/103 [=====] - 116s 1s/step - loss: 0.1228 - accuracy: 0.9818 - val_loss: 0.1075 - val_accuracy: 0.9782
Epoch 7/10
103/103 [=====] - 113s 1s/step - loss: 0.0394 - accuracy: 0.9927 - val_loss: 0.0264 - val_accuracy: 0.9927
Epoch 8/10
103/103 [=====] - 116s 1s/step - loss: 0.0129 - accuracy: 0.9988 - val_loss: 0.0408 - val_accuracy: 0.9927
Epoch 9/10
103/103 [=====] - 113s 1s/step - loss: 0.0209 - accuracy: 0.9976 - val_loss: 0.0804 - val_accuracy: 0.9927
Epoch 10/10
103/103 [=====] - 113s 1s/step - loss: 0.0343 - accuracy: 0.9927 - val_loss: 0.0500 - val_accuracy: 0.9855

```

Results

```

y_pred = model2.predict(X_valid, verbose=1)
y_pred_bool = np.argmax(y_pred, axis=1)

print(classification_report(y_valid, y_pred_bool))

print(confusion_matrix(y_true=y_valid, y_pred=y_pred_bool))

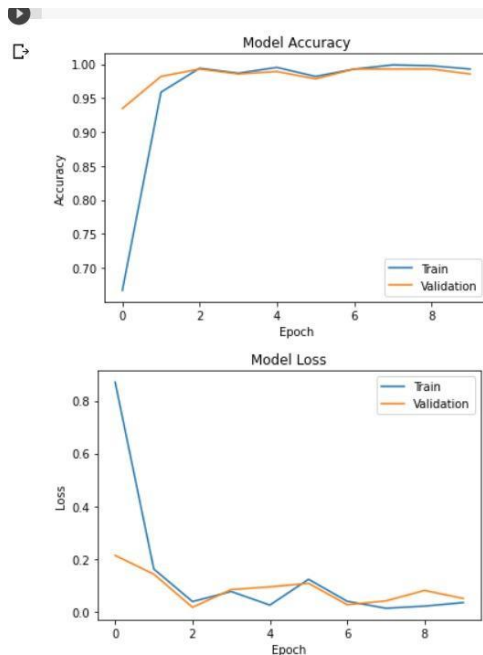
```

```

9/9 [=====] - 10s 1s/step
          precision    recall  f1-score   support

     0       0.95      0.67      0.78        30
     1       0.94      0.99      0.97       141

```



Testing

```
path = "/content/Test cases/000019_03_01_025.png"
img = load_img(path, target_size = (224,224))
input_arr = img_to_array(img)/255

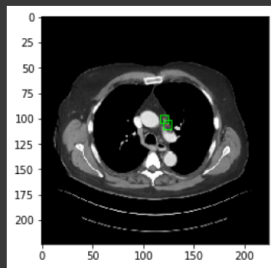
plt.imshow(input_arr)
plt.show()

input_arr.shape

input_arr = np.expand_dims(input_arr, axis = 0)

#pred = model2.predict_classes(input_arr)[0][0][0]
pred = model2.predict
np.argmax(categories)
if np.argmax(categories) == 0:
    print("The following CT Image is Benign")
    print("It's symptoms are:\n1)Mild Cough\n2)Shortness of breath\n3)Coughing up blood")
elif np.argmax(categories) == 1:
    print("The following CT Image is Malignant")
    print("It's symptoms are:\n1)Severe Cough\n2)Shortness of breath\n3)Coughing up blood\n4)Chest pain\n5)Hoarseness\n6)Losing much weight\n7)Bone pain\n8)Continuous")
else:
    print("The following CT Image is Normal")
    print("You have no lung cancer, but its good to take precautions like\n1)Quit smoking (if you smoke)\n2)Eat healthy food in a balanced diet\n3)Exercise regularly\n4)Get regular medical checkups")
```

Result:



The following CT Image is Normal
You have no lung cancer, but its good to take precautions like
1)Quit smoking (if you smoke)
2)Eat healthy food in a balanced diet
3)Exercise regularly
4)Get regular medical checkups

Model 3: Data Augmentation

```
[ ] train_datagen = ImageDataGenerator(horizontal_flip=True, vertical_flip=True)
    val_datagen = ImageDataGenerator()
```

```
[ ] train_generator = train_datagen.flow(X_train, y_train, batch_size=8)
    val_generator = val_datagen.flow(X_valid, y_valid, batch_size=8)
```

```
model3 = Sequential()

model3.add(Conv2D(64, (3, 3), input_shape=X_train.shape[1:]))
model3.add(Activation('relu'))
model3.add(MaxPooling2D(pool_size=(2, 2)))

model3.add(Conv2D(64, (3, 3), activation='relu'))
model3.add(MaxPooling2D(pool_size=(2, 2)))

model3.add(Flatten())
model3.add(Dense(16))
model3.add(Dense(3, activation='softmax'))
```

```
model3.summary()
```

Model: "sequential_2"

```

history = model3.fit_generator(train_generator, epochs=10, validation_data=val_generator, class_weight=new_weights))

... Epoch 1/10
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future
    """Entry point for launching an IPython kernel.
103/103 [=====] - 118s 1s/step - loss: 1.2938 - accuracy: 0.5122 - val_loss: 0.8444 - val_accuracy: 0.6073
Epoch 2/10
103/103 [=====] - 114s 1s/step - loss: 0.7468 - accuracy: 0.6582 - val_loss: 0.6310 - val_accuracy: 0.7309
Epoch 3/10
103/103 [=====] - 117s 1s/step - loss: 0.5001 - accuracy: 0.8163 - val_loss: 0.4961 - val_accuracy: 0.7745
Epoch 4/10
103/103 [=====] - 115s 1s/step - loss: 0.3314 - accuracy: 0.8881 - val_loss: 0.2716 - val_accuracy: 0.8545
Epoch 5/10
103/103 [=====] - 118s 1s/step - loss: 0.1571 - accuracy: 0.9404 - val_loss: 0.1500 - val_accuracy: 0.9491
Epoch 6/10
103/103 [=====] - 115s 1s/step - loss: 0.1847 - accuracy: 0.9453 - val_loss: 0.3225 - val_accuracy: 0.8582
Epoch 7/10
103/103 [=====] - 115s 1s/step - loss: 0.1486 - accuracy: 0.9623 - val_loss: 0.1278 - val_accuracy: 0.9673
Epoch 8/10
103/103 [=====] - 115s 1s/step - loss: 0.0859 - accuracy: 0.9842 - val_loss: 0.1193 - val_accuracy: 0.9455
Epoch 9/10
103/103 [=====] - 118s 1s/step - loss: 0.1322 - accuracy: 0.9757 - val_loss: 0.0446 - val_accuracy: 0.9927
Epoch 10/10
103/103 [=====] - 115s 1s/step - loss: 0.0448 - accuracy: 0.9915 - val_loss: 0.0235 - val_accuracy: 0.9964

```

```

[ ] y_pred = model3.predict(X_valid, verbose=1)
y_pred_bool = np.argmax(y_pred, axis=1)

print(classification_report(y_valid, y_pred_bool))

print(confusion_matrix(y_true=y_valid, y_pred=y_pred_bool))

```

```

9/9 [=====] - 10s 1s/step

```

	precision	recall	f1-score	support
0	0.50	0.60	0.55	30
1	0.88	0.79	0.83	141
2	0.71	0.76	0.73	104
accuracy			0.76	275
macro avg	0.70	0.72	0.70	275
weighted avg	0.77	0.76	0.76	275

```

path = "/content/Test cases/000019_03_01_025.png"
img = load_img(path, target_size = (224,224))
input_arr = img_to_array(img)/255

plt.imshow(input_arr)
plt.show()

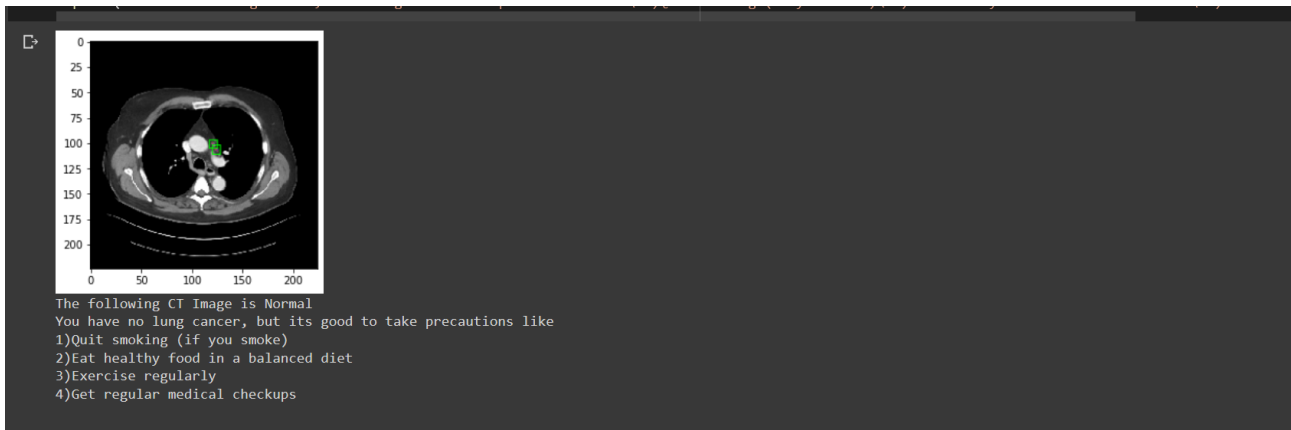
input_arr.shape

input_arr = np.expand_dims(input_arr, axis = 0)

#pred = model3.predict_classes(input_arr)[0][0][0]
pred = model3.predict
np.argmax(categories)
if np.argmax(categories) == 0:
    print("The following CT Image is Benign")
    print("It's symptoms are:\n1)Mild Cough\n2)Shortness of breath\n3)Coughing up blood")
elif np.argmax(categories) == 1:
    print("The following CT Image is Malignant")
    print("It's symptoms are:\n1)Severe Cough\n2)Shortness of breath\n3)Coughing up blood\n4)Chest pain\n5)Hoarseness\n6)Losing much weight\n7)Bone pain\n8)Continuous H
else:
    print("The following CT Image is Normal")
    print("You have no lung cancer, but its good to take precautions like\n1)Quit smoking (if you smoke)\n2)Eat healthy food in a balanced diet\n3)Exercise regularly\n

```

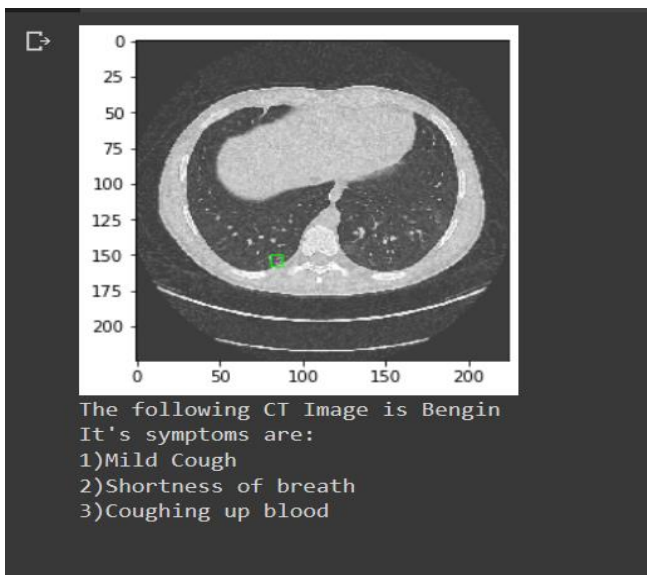
Result:



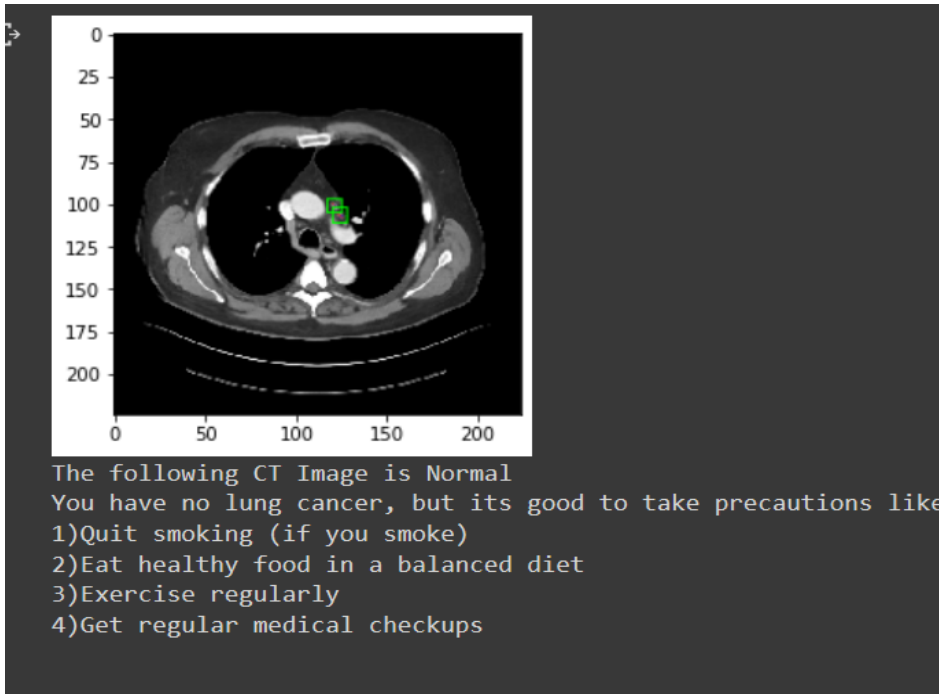
Result and Discussion:

Model 1- Model building with SMOTE data

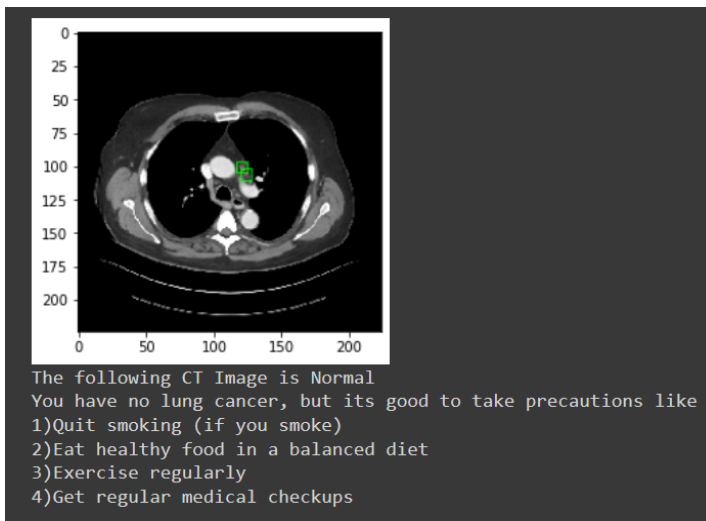
SMOTE is the Synthetic Minority Oversampling Technique used to balance the dataset when we are supposed to take a huge dataset.



Model 2: Model Building with class weighted Approach



Model-3:



Accuracy:

Model 1:

Model Building with smote -98.55

Model 2:

Model building with classes weighted Approach -98.55

Model 3:

Data agumentation-99.64

CONCLUSION AND FUTURE SCOPE

1.1. Conclusion

In this project, we study the use of image processing and deep learning techniques to predict lung cancer nodules in vulnerable patients. With the help of the research that we did, deep analysis and by gaining in depth knowledge of the scenario and its significance in the real world, we were able to develop a CNN model for lung cancer detection. As a part of building our model we used various approaches and carried out image processing and classification, which leaded us to coming up with a novel system that detects lung cancer nodules with high accuracy. We were able to develop a full model that runs with more than 95% accuracy on test data. Given the difficult nature of the problem, diverseness of the data and computing difficulties we faced various challenges throughout the process of pre-processing the data so that the features in the images can be detected well and also working with the imbalanced data was a crucial step. The CT scans being in hundreds of images had a memory constraint while processing and also was a time-consuming process.

1.2. Future Scope

In this project we have built and trained CNN models employing various techniques like SMOTE and Class Weighted Approach to detect lung cancer nodules. In addition to this there is a huge scope to use pre- trained models like VGG16, Resnet50 etc. which can be used for feature extraction and obtain high accuracy while predicting the cancerous cells. Image classification being one of the most complicated and crucial stage of our project, to process the images rightly we can use wide range of deep learning technologies and figure out which one obtains more accuracy. At each stage of the project, there a possibility that we can use different techniques like making use of pre-trained models like Exception, Google net, etc, instead of using traditional CNNs for feature extraction and other modules for deep learning could be combined with deferent loss function, layers and optimization technique which would overall lead to a better model. In this project we

have used the publicly available IQ-OTHNCCD lung cancer dataset which has 1190 CT scan images. There is also a scope of using different dataset which contains wide varieties of CT scans for the models to be trained on to accurately understand the nature of CT scans that are cancerous. Furthermore, we can work on the LIDC-IDRI cancer dataset, which is genuinely available from the cancer imaging archive and contains DICOM files containing collective information about the patients as well as multiple CT scan slices for a single patient, which will be extremely useful in detecting cancer much more precisely.

REFERENCES

- [1] L. A. Torre, F. Bray, R. L. Siegel, J. Fer lay, J. Lortet-Tieulent, and A. Jemal, “Global cancer statistics, 2012,” *CA, Cancer J. Clin.*, vol. 65, no. 2, pp. 87–108, 2015.
- [2] T. Missay, R. C. Hardie, and S. K. Rogers, “A new computationally efficient CAD system for pulmonary nodule detection in CT imagery,” *Med. Image Anal.*, vol. 14, no. 3, pp. 390–406, Jun. 2010.
- [3] D. E. Midturn, “Early diagnosis of lung cancer,” *Dept. Pulmonary Crit. Care Med.*, Mayo Clinic, Rochester, MN, USA, F1000Prime Rep., 2013, vol. 5, p. 12.
- [4] S. Blandine Knight, P. A. Crosbie, H. Balata, J. Chizik, T. Hassell, and C. Dive, “Progress and prospects of early detection in lung cancer,” *Open Biol.*, vol. 7, no. 9, Sep. 2017, Art. no. 170070.
- [5] L. A. Torre, F. Bray, R. L. Siegel, J. Fer lay, J. Lortet-Tieulent, and A. Jemal, “Global cancer statistics, 2012,” *CA, Cancer J. Clin.*, vol. 65, no. 2, pp. 87–108, 2015.
- [6] T. Missay, R. C. Hardie, and S. K. Rogers, “A new computationally efficient CAD system for pulmonary nodule detection in CT imagery,” *Med. Image Anal.*, vol. 14, no. 3, pp. 390–406, Jun. 2010.
- [7] D. E. Midturn, “Early diagnosis of lung cancer,” *Dept. Pulmonary Crit. Care Med.*, Mayo Clinic, Rochester, MN, USA, F1000Prime Rep., 2013, vol. 5, p. 12.
- [8] S. Blandine Knight, P. A. Crosbie, H. Balata, J. Chizik, T. Hassell, and C. Dive, “Progress and prospects of early detection in lung cancer,” *Open Biol.*, vol. 7, no. 9, Sep. 2017, Art. no. 170070.
- [9] K. Awai et al., “Pulmonary nodules at chest CT: Effect of computer aided diagnosis on radiologists’ detection performance,” *Radiology*, vol. 230, no. 2, pp. 347–352, Feb. 2004.
- [10] D. Ost, A. M. Fein, and S. H. Fein silver, “The solitary pulmonary nodule,” *New England J. Med.*, vol. 348, no. 25, pp. 2535–2542, Jul. 2003.
- [11] L. A. Torre, F. Bray, R. L. Siegel, J. Fer lay, J. Lortet-Tieulent, and A. Jemal, “Global cancer statistics, 2012,” *CA, Cancer J. Clin.*, vol. 65, no. 2, pp. 87–108, 2015.
- [12] T. Missay, R. C. Hardie, and S. K. Rogers, “A new computationally efficient CAD system for

pulmonary nodule detection in CT imagery,” *Med. Image Anal.*, vol. 14, no. 3, pp. 390–406, Jun. 2010.

[13] D. E. Midturn, “Early diagnosis of lung cancer,” *Dept. Pulmonary Crit. Care Med.*, Mayo Clinic, Rochester, MN, USA, F1000Prime Rep., 2013, vol. 5, p. 12.

[14] S. Blandine Knight, P. A. Crosbie, H. Balata, J. Chizik, T. Hassell, and C. Dive, “Progress and prospects of early detection in lung cancer,” *Open Biol.*, vol. 7, no. 9, Sep. 2017, Art. no. 170070.

[15] K. Awai et al., “Pulmonary nodules at chest CT: Effect of computer aided diagnosis on radiologists’ detection performance,” *Radiology*, vol. 230, no. 2, pp. 347–352, Feb. 2004.

[16] D. Ost, A. M. Fein, and S. H. Fein silver, “The solitary pulmonary nodule,” *New England J. Med.*, vol. 348, no. 25, pp. 2535–2542, Jul. 2003.

[17] M. N. Gurcan et al., “Lung nodule detection on thoracic computed tomography images: Preliminary evaluation of a computer-aided diagnosis system,” *Med. Phys.*, vol. 29, no. 11, pp. 2552–2558, Nov. 2002.

[18] A. A. A. Setio et al., “Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge,” *Med. ImageAnal.*, vol. 42, pp. 1–13, Dec. 201

