

# **Deep 3D Convolutional Neural Network in Early Detection of Lung Cancer**



Inspiring Excellence

**Supervisor:**  
Hossain Arif,  
Assistant Professor,  
Department of Computer Science and Engineering.

## **Submitted By:**

Khandoker Jobayer Hossain-13201007

Md. Jewel Rana-13101203

Abdullah Ali Ajghar-13301057

Nur-a-taj Alam-13301141

**Department of Computer Science and Engineering,  
Submission Date:22-07-2018**

## DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This Thesis, neither in whole or in part, has been previously submitted for any degree.

Signature of Supervisor

Signature of Author

---

Mr. Hossain Arif

---

Khandokar Jobayer Hossain

---

Md.Jewel Rana

---

Abdullah Ali Ajghar

---

Nur-a-Taj Alam

## Abstract

Early detection of lung cancer is essential for the survival of patients. Lung Cancer remains the deadliest cancer in the world. Lung cancer diagnosis is still a time consuming and long awaiting process. The oncologist is facing tremendous workload for the diagnosis of lung cancer. The quality of their work is also deteriorating as work-pressure is increases. On the other hand, the number of cancer affected people is increasing day by day due to environment changes (environmental pollution, increase of UV radiation, increase of radiation substances in the environment, etc.). So, computer-aided-diagnosis systems can play an essential role to detect stage I lung cancer which may force the survival rate of lung cancer from below 20% to more than 70%. In this paper, we propose a model to detect the malignancy from a patient's CT Scans using 3D convolutional neural network. We used lung CT volumes to train our model, which consists of Input layer, several Hidden layers, Fully Connected layers and finally the output layer to show the malignancy. The proposed system runs in a workstation with a moderate configuration and the output comes in a very short time. We do believe that, this application of deep 3D convolutional neural network will have a great impact in the field of medical science and cancer diagnosis.

Keywords: Convolutional Neural Network, CT scan, Lung segmentation.

## **Acknowledgement**

At first all credit goes to ALLAH, who is the creator of the whole world and the owner, the most unique and inherently one, all-merciful and omnipotent who gave us the energy, strength and power to complete the thesis.

We are grateful to our thesis supervisor, Hossain Arif Sir, whose encouragement, guidance, support and kind co-operation in completion of our final thesis project. We are also very thankful to all our faculty members, librarian, other staff members of BRAC University and also all my lovely friends who gave me useful tips and valuable suggestions regarding the thesis project.

Finally, we would like to express our heartfelt thanks to our beloved parents who were very supportive both financially and mentally and for their encouragement, love and care.

## Contents

1. Chapter-1 (Overview)	
1.1 Introduction	1
1.2 Objectives	3
1.3 Report Overview	3
2. Chapter-2 (Literature Review)	
2.1 History of Convolutional Neural Network	4
2.2 Related Work	6
3. Chapter-3 (Deep Convolutional Neural Network Model)	
3.1 Dataset	8
3.2 Preprocessing	8
3.2.1 Loading Dataset and Convert to Hounsfield Unit	9
3.2.2 Resampling	10
3.2.3 Segmentation	11
3.2.4 Down Sampling	13
3.3 Convolutional Neural Network Model	14
3.4 Initialization	17
3.5 Iteration and training	18
4. Chapter-4 (Evaluation and Result)	
4.1 Running Environment	20
4.2 Accuracy Evaluation	21
4.3 Variables Evaluation	23
4.4 Validation Results	25
5. Chapter-5 (Discussion)	
5.1 Accuracy and Prediction	27
5.2 Confusion Matrix	28
5.3 Comparative Study	29
6. Chapter-6 (Conclusions)	
6.1 Conclusion	31
6.2 Future Work	31
7. Reference	33

## List of figures:

Figure 2.1: Alex Net architecture	5
Figure 2.2: Pipeline for commonly used cancer detection technique	6
Figure 3.1: Standard Hounsfield Units for CT	9
Figure 3.2: a) after converting to hounsfield unit	10
b) image representation of the slice	
Figure 3.3: 3D image of a scan after resampling	11
Figure 3.4:(a) Original image	12
(b) Threshold Image	
(c) After erosion and dilation	
(d) Color labels	
(e) Final mask	
(f) Apply mask on original	
Figure 3.5: 3D Segmented Lung After Segmentation	13
Figure 3.6: 3D Deep Convolutional Neural Network Model	14
Figure 3.7: 3D convolution layer filter projection	16
Figure 4.1: TensorFlow Computational Graph	20
Figure 4.2: (a) weights of convolution layer 1	24
(b) weights of convolution layer 4	
(c) weights of fully connected layer 2	
Figure 4.3: (a) biases of convolution layer 1	24
(b) biases of convolution layer 4	
(c) biases of fully connected layer 2	
Figure 4.4: Accuracy after each epoch	25
Figure 4.5: Amount of loss calculated during training	26
Figure 5: Comparative study between our model and different models	30

## List of Tables:

Table 1: Feature sizes after each Convolution Layer and Max pooling Layer	21
Table 2: Number of input neurons and output neurons from each of the Fully Connected Layers	23
Table 3: Prediction of Cancer from our Model	27
Table 4: Confusion Matrix	29
Table 5: Prediction Analysis	29

# Chapter 1

## Overview

### 1.1 Introduction

Lung cancer is one of the most common cases of cancer all over the world [1]. About 14% of the cancer affected people is affected by lung cancer, which accounts for the death of approximately 21% of all cancer deaths [1]. Males are more vulnerable to lung cancer than their counterpart females. Due to environment pollution, number of lung cancer affected people is increasing very rapidly [1]. The 5-year survival rate of lung cancer is approximately 17.7 percent, reported by the world health organization [2]. Although most of the patients with lung cancer are detected in later stages of cancer. Early detection can enhance the survival rate to above 70% [2].

The last few decades computer-aided-diagnosis became one of the fascinating fields of research for the researchers. Computerized image segmentation techniques, machine learning algorithms are useful for feature extraction and detecting malignant cancerous nodules. Traditional cancer detection is performed by oncologist and the first step to diagnosis is detection of cancer from Computed Tomography (CT) images. The more the number of affected people is increasing the more the quality of work is declining. Also, the workload for them is rising. The average time needed for the detection of lung cancer from CT images is more than 30 days for a single patient [3]. And sometimes the process need to be repeated. So, if the patient is diagnosed as cancer a lot of valuable time wastes to detect cancer. So, Computerized diagnosis can help the oncologist to detect cancer more quickly and can help to save people [4].



Machine learning algorithms, especially deep architected learning algorithms have attracted intensive research interests. Convolutional Neural Network is the most commonly used deep architectural model for classification and for image segmentation. The 2-dimensional Convolutional Neural Network features 2D map on a 2-D plain. In 2D Convolutional Neural Network model 2-dimensional images are used. For feature extraction and learning process the images go through several layers with Weights and Biases. The initial layers most often consist of Convolution Layer and Max pooling Layer followed by one or more Fully Connected Layers and finally the Output Layer. Activation function and Dropout is used in each layer. In 2D Convolutional Neural Network, each layer produces 2D images with different color channels and different filter sizes [4]. In 2D CNN model weights with 2-dimensional sizes are used and number of output images are defined by biases. Then in each layer an activation function is used for regularization. There are various types of activation functions such as: linear activation function, non-linear activation function, sigmoid activation functions, rectified linear unit (Relu) are used for regularization. In the Fully Connected Layers activation functions are used and a dropout function is used. Dropout function defines the minimum probability for a class and dropout function can help to reduce overfitting problem.

However, 3D Convolutional Neural Network is built for 3-dimensional images with RGB channels. 3D CNN model projects feature map onto 3D map via a 3D filter [5]. The 3D filter produces 3D images with different color channels. In 3D Convolutional Neural Network 3-Dimensional input images are used. Then several hidden layers comprised with Convolution Layer, Max pooling Layer, Fully Connected Layers generates Different images with different sizes, which are used for learning. In the Convolutional Layers, Weights with 3-dimensional sizes are used. Biases defines the number of output image or output neuron from each layer.

Initial hidden layers which comprised with Convolution layer and max pooling layer generates 3D images with different color channels of the input image or images for the layer [5].

## **1.2 Objectives**

We have developed a deep convolutional neural network for the detection of lung cancer. Our work can be used for:

- Lung cancer detection (Early stage preferably stage-1).
- Improvement of Alex Net architecture for classification with small dataset.
- Aid doctors and oncologist in detection process.

## **1.3 Report Overview**

In this paper, we aim to investigate the performance of our 3D Convolutional Neural Network model by training the model with 3D lung segmented CT images for early detection of lung cancer. Lung CT images are very crucial to work with. If we use 2D images, then important and valuable information can be missed about the nodules. The cancer nodules are very small in size initially. So, to detect them we used 3D Convolutional Neural Network. The whole training and testing procedure is divided into four parts: (a) Loading Dataset, (b) Segmentation and preprocessing the whole dataset in a file to minimize the time for training and testing, (c) Training process, (d) Testing the model with a classification function expertizing on binary classes. The paper is organized as follows: in Chapter 3, we discuss about the dataset, preprocessing techniques and our proposed 3D deep Convolutional Neural Network. Chapter-4 describes the experimental setup and chapter 5 gives the experimental result and the contribution and limitation of the project. Chapter 6 of this paper is the ending remarks and the scope of future work of this work.

## Chapter 2

### Literature Review

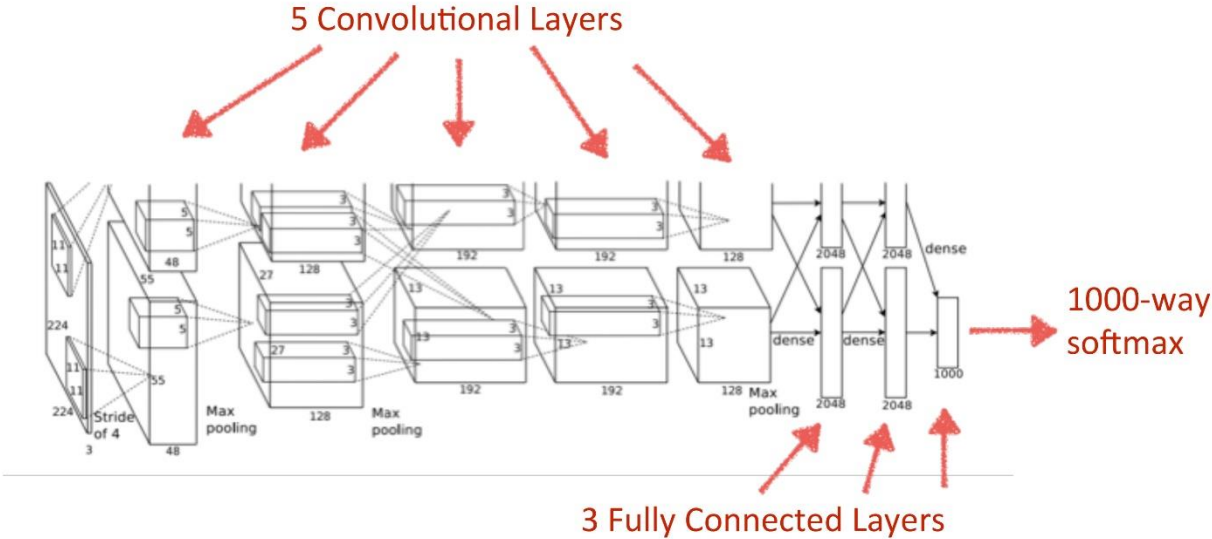
#### 2.1 History of Convolutional Neural Network

In this modern world computers are better than human at crunching series of numbers. But there are many tasks which is more complex. For example how do people teach a computer what an animal look like or how to drive a car? In these cases, computer scientists use neural networks [6][7]. Neural network mainly a subset of algorithm which make around a model of artificial neurons spread across three or more than three layers. There are more machine learning techniques available but these are not relying on neural networks. After that artificial neural network systems are a branch of substantive learning models bottomed by natural neural systems. Neural systems which are already created displayed as frameworks of internally connected neurons. Deep neural networks that have hidden layers which are mainly trained using new methods [8]. Deep learning mainly used for construct particularly perplexing networks with many more layers than normal. What is the advantage of these added layers? This is that the networks are able to develop much greater levels of abstraction. These are necessary for certain complex tasks. For example, image recognition and automatic translation and many more.

In 1943, Warren McCulloch and Walter Pitts build the first model of neural network. Warren McCulloch and Walter Pitts mainly made this model for mathematical and algorithm purpose but could not be tested due to lack of computational resources. After that, in 1958, Frank Rosenblatt give a model that could do pattern recognition but the main model could not be tested. Then 7years later 1965 Alexey Ivakhnenko and Lapa published a neural network model which could be tested and had many layers. After these 1969, 1975, 1992 more scientists

were came and they published more NN models and made neural network learning more efficient [9]. In 2011, deep neural network started incorporating convolutional layers with maximum pooling layers then passed to several fully connected layers which were followed by an output layer. And these are called Convolutional Neural Networks. In current years, deep neural networks have won numerous contests in pattern recognition and machine learning [10].

Traditional Alex Net architecture was first proposed on 2012 [4]. The architecture is comprised with 5 Convolution layers and 3 Fully connected layers. In Alex Net architecture uses relu activation function, which makes the training process 6 times faster than the Sigmoid and Tanh activation functions. Also, dropout of 0.5 is used instead of regularization to deal with overfitting for small datasets.



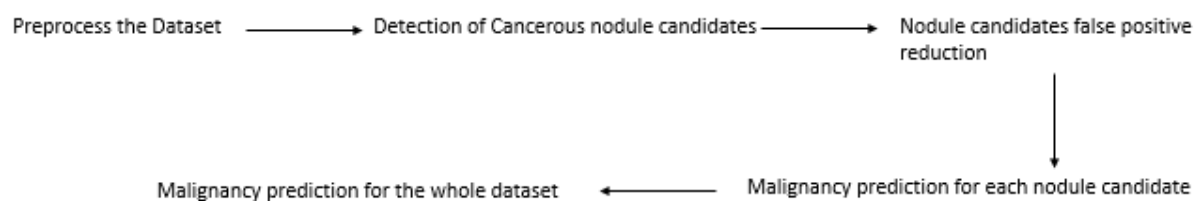
**Figure 2.1: Alex Net architecture**

The input of Alex Net architecture should be  $227 \times 227$  and convolution layer 1 size should be  $11 \times 11$  with 96 filters and stride of 4 . Max pooling is used after each of the convolution layer to keep the important features. In the first 2 Fully connected layers dropout of 0.5 is used to deal with overfitting. Figure 2.1 shows the Alex Net architecture.

## 2.2 Related Work

In the last several years computing power has caught up. Distributed systems like Hadoop's Map Reduce paradigm means you may no longer need a supercomputer to handle the massive calculations neural networks require-you can just spread the job out across clusters of commodity hardware. Neural networks are the cutting edge of machine learning and artificial intelligence which implementing them requires expertise in statistical analysis, distributed systems, big data processing and this related fields. There are a large number of different libraries available that make designing and implementing neural networks relatively easy which are scikit-learn, Theano(It's a Python machine learning library), TensorFlow and Deep learning( It's a Python based library).

Almost every research paper on lung cancer detection follows the pipeline [11]:



**Figure 2.2: Pipeline for commonly used cancer detection technique**

But, this pipeline requires more than one dataset. Cancer dataset are crucial, and they are hard to find. Also, each phase of the above pipeline requires different dataset. For example, false positive reduction phase requires true or false labeled nodule candidates and for prediction of malignancy requires a dataset labeled with malignant nodules. However, dataset labeled with malignant nodule may not exist for some cases. So, the System creates complexity. The maximum accuracy acquired by this System is 0.83[12]. However, in our deep learning model tried to overcome these issues. We also aim to acquire a performance close to the above system.

## Chapter 3

### 3D Deep Convolutional Neural Network Model

#### 3.1 Dataset

The Kaggle Data Science Bowl 2017 dataset [13] is used for our model. The dataset is obtained from US National Cancer Institute which comprised of 1,595 instances of high risk patient's lung CT scans saved in DICOM standard. For each patient's scan the data contains 2D axial slices and the slice size is 512 x 512. The number of slices varies from 128 to 220 depending on machine and patient. The DICOM files have a header that contains important information about the patient's id, as well as the scan parameters like slice thickness, pixel spacing etc. Each instance is labeled with either 0 or 1, where 0 stands for the patient is diagnosed as non-cancer and 1 means the patient is diagnosed as cancer. Within the dataset 1176 instances are labeled as 'non-cancer' and 419 instances are labeled as 'cancer'.

#### 3.2 Preprocessing

The preprocessing purpose of the dataset is to remove the unnecessary things from our dataset. Convolutional Neural Networks performs well if the dataset contains only the necessary information [14]. For our case the input images are lung CT images and the images contain a lot of information which is not necessary for our model. The whole preprocessing is coded in python 3.6. The libraries used for preprocessing are numpy, matplotlib, pydicom, pandas, os, Scikit-learn. The notebook used for coding and testing is PYCHARM. For these purposes we divide our preprocess into several small modules: (a) Loading dataset and convert to HU (Hounsfield Unit) (b) Resampling the images (c) Segmentation (d) Down sampling. The work in [14] also uses almost the identical preprocessing techniques and much

of our preprocessing techniques are developed following their preprocessing method for lung CT image.

### 3.2.1 Loading dataset and Convert to Hounsfield Unit

The first task of our preprocessing is loading the dataset. The whole image data were kept in a directory in the secondary memory. The directory contains the images of the instances used for training and testing. The directory contains the images of each instance are named by the patients id. The images for an instance are first loaded into a List for further manipulation.

The voxel values in the images are raw and they need to be converted to Hounsfield Unit. The standard Hounsfield Unit for lung CT images are shown in Figure-3.1 [15].

<i>TISSUE</i>	<i>HU</i>
Lung	-500
<i>Fat</i>	<i>-100 to -50</i>
<i>CFS</i>	<i>+15</i>
<i>Blood</i>	<i>+30 to +45</i>
<i>Muscle</i>	<i>+10 to +40</i>
<i>Adrenal tumor</i>	<i>less than +10</i>
<i>White matter</i>	<i>+20 to +30</i>
<i>Grey matter</i>	<i>+37 to +45</i>
<i>Liver</i>	<i>+40 to +60</i>
<i>Bone</i>	<i>+700 to + 3000</i>

**Figure 3.1: Standard Hounsfield Units for CT**

The images for an instance are then converted to Hounsfield Unit shown in figure 3.1. After conversion to Hounsfield unit a Histogram with different types of elements in the images are shown in Figure-3.2.a. From the Histogram we can see that, there are a lot of air, there is some lungs, some other muscle mostly liver, soft tissue etc., from the tiny line between 700 to 3000 represents bones.



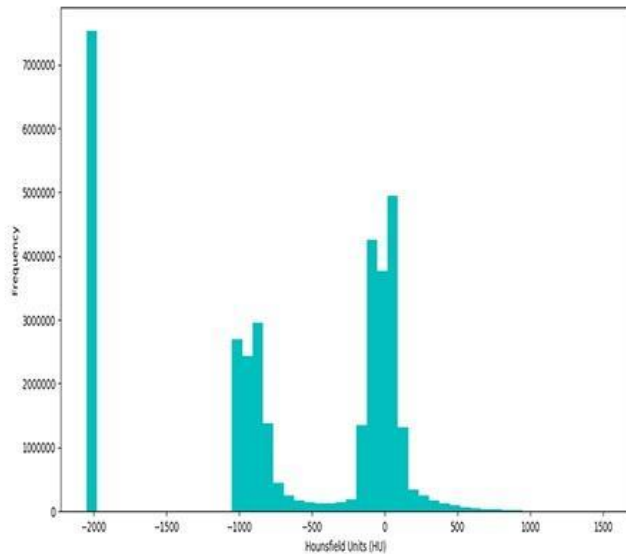


Figure 3(a)

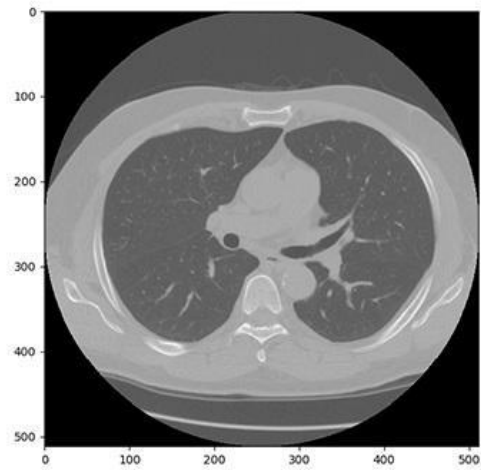
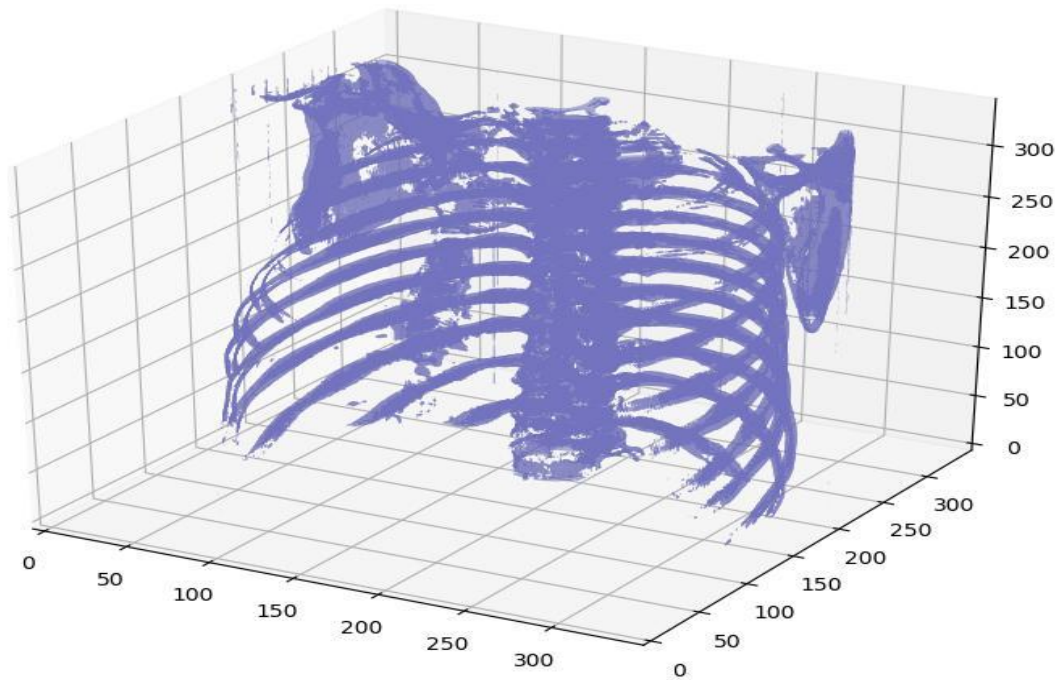


Figure 3(b)

**Figure 3.2: a) after converting to hounsfield unit, b)image representation of the slice**

### 3.2.2 Resampling

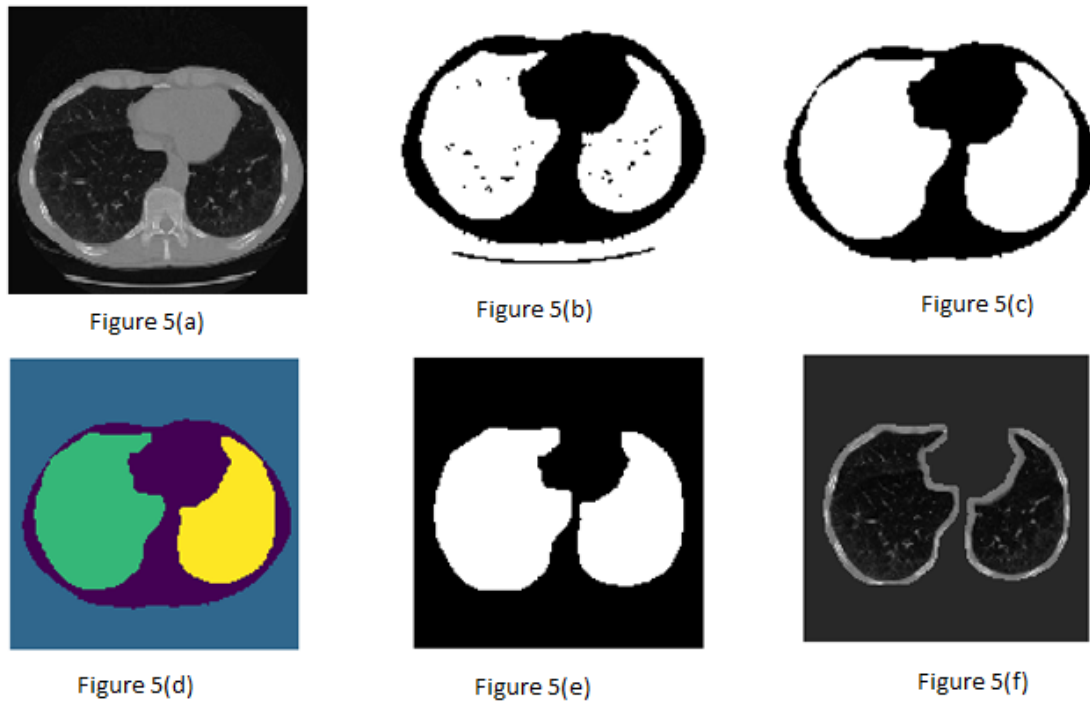
After converting to Hounsfield Unit we have each individual slice and it is not clear how thick each image is. Using the metadata from the DICOM header we can see that the size of each voxel as the slice thickness. For visualization purpose, images then resampled in 1x1x1 mm pixels and slices. Also, different voxel for different images can be problematic for 3D Convolutional Neural Network. The 3D resampled image for an instance is shown in Figure-3.3 [14]. For resampling slice thickness of 0.69 is used and the pixel spacing is also used 0.69. The 512 x 512 images then stacked to form a 3-D image.



**Figure 3.3: 3D image of a scan after resampling**

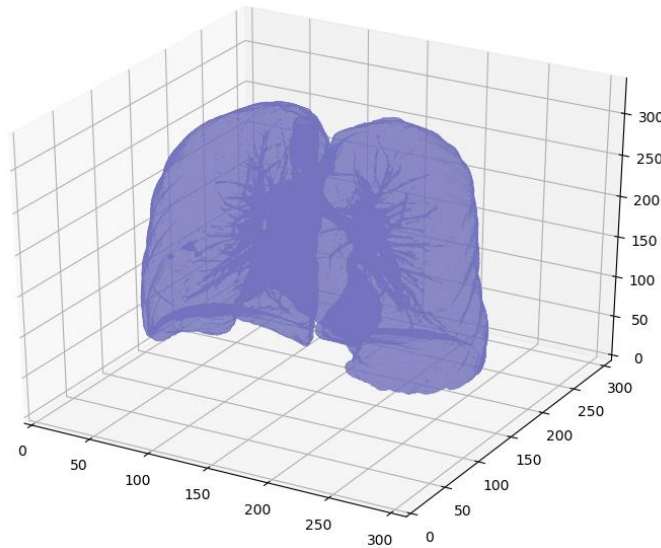
### **3.2.3 Segmentation**

Machine learning algorithms perform better if the input set is narrowly defined to it as what it is looking at. If we can create different model with whole chest region and only lung region, the model with only the lung region will perform better. So, too narrowly define the problem space we perform segmentation [14][18].



**Figure 3.4:(a) Original image, (b) Threshold Image, (c) After erosion and dilation, (d) Color labels, (e) Final mask, (f) Apply mask on original**

Firstly, we subtracted the mean value of the image pixel from their original value and then we divided it by the standard deviation of the pixel value. Then we Created two KMeans clusters to compare between soft tissue vs bone and air vs lung to identify the proper threshold. Then using erosion and dilation we removed the pulmonary vessels and noise. We then identified each region of the image and show them in a different color and label. Furthermore, from the label we generate the mask for the lung and we apply the mask to the original image. All steps in segmentation process are shown in Figure-3.4 [14] [18].



**Figure 3.5: 3D Segmented Lung After Segmentation**

Figure 3.5 shows the 3d segmented lung. The python library matplotlib is used for plotting the 3D image. The segmentation techniques used in our work follows the procedure described in other related works [14].

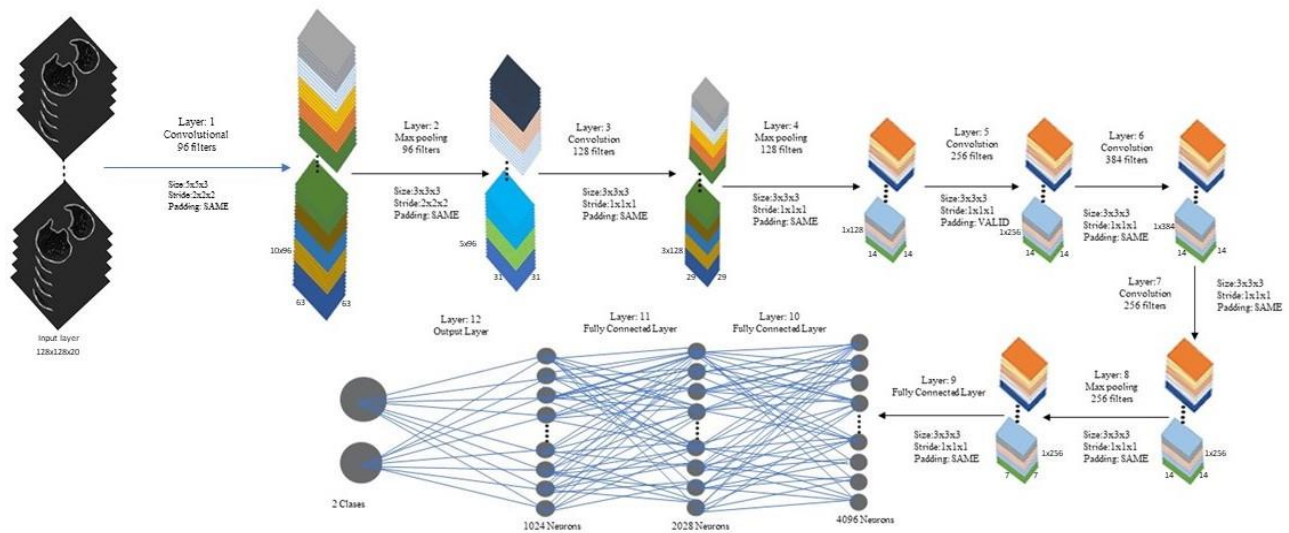
### **3.2.4 Down sampling**

As the images are inconsistent in sizes and to reduce the stress on memory we further down sampled the images [5]. A volume of (128 x 128 x 20) is used to down sample the images. The images are first loaded and then 128, 128, 20 values are divided respectively by X, Y, Z- axis voxel size of the images to calculate the resize factor. Then, using the resize factor images are resized and the new image size after this step is (128 x 128 x 20). The cancer/non-cancer label of the dataset are then attached to the images. After these steps, the images are ready to feed into our 3D Convolutional Neural Network.

### 3.3 3D Convolutional Neural Network Model

The Training and Testing of our model is done using 3D Convolutional Neural Network Architectural model also known as Alex Net [4]. To train and test of our model we have used the open-source library TensorFlow and Tensor board libraries for python 3+, which are introduced by google for machine learning.

Our proposed model is shown in Figure-6. From the Figure we can see that, we have used 13 Layers including Input Layer and Output Layer. Rather than the two layers there are 5 Convolution Layers,3 Max pooling Layers and 4 Fully Connected Layers[5][18][19]. The Input Layer holds the Input image and send the input image to the next Layer. The input image size is (128 x 128 x 20).



**Figure 3.6: 3D Deep Convolutional Neural Network Model**

The Convolution Layer Accepts the Volume size ( $W_1 \times H_1 \times D_1$ ) as input  
 Here,  $W_1$  = Width of the Input image in the Convolution Layer

$H_1$  = Height of the Input image in the Convolution Layer

$D_1$  = Depth of the Input image in the Convolution Layer

Every Convolution Layer requires four hyperparameters:

Number of filters:  $K$

The spatial extent:  $F$

The stride:  $S$

The number of Zero padding:  $P$

The Convolution Layer produces a volume size of  $(W_2 \times H_2 \times D_2)$  as output:

$$\text{Now, } W_2 = (W_1 - F + 2P) / S + 1 \quad (1)$$

$$H_2 = (H_1 - F + 2P) / S + 1 \quad (2)$$

$$D_2 = K \quad (3)$$

The Max Pooling Layer accepts the volume size  $(W_3 \times H_3 \times D_3)$  as input:

Here,  $W_3$  = Width of the input image in the Max Pooling Layer

$H_3$  = Height of the input image in the Max Pooling Layer

$D_3$  = Depth of the input image in the Max Pooling Layer

Every Max Pooling Layer requires two hyperparameters:

The spatial extent:  $F$

The stride:  $S$

The Max pooling Layer produces a volume size of  $(W_4 \times H_4 \times D_4)$  as output:

$$\text{Now, } W_4 = (W_3 - F) / S + 1 \quad (4)$$

$$H_4 = (H_3 - F) / S + 1 \quad (5)$$

$$D_4 = D_3 \quad (6)$$

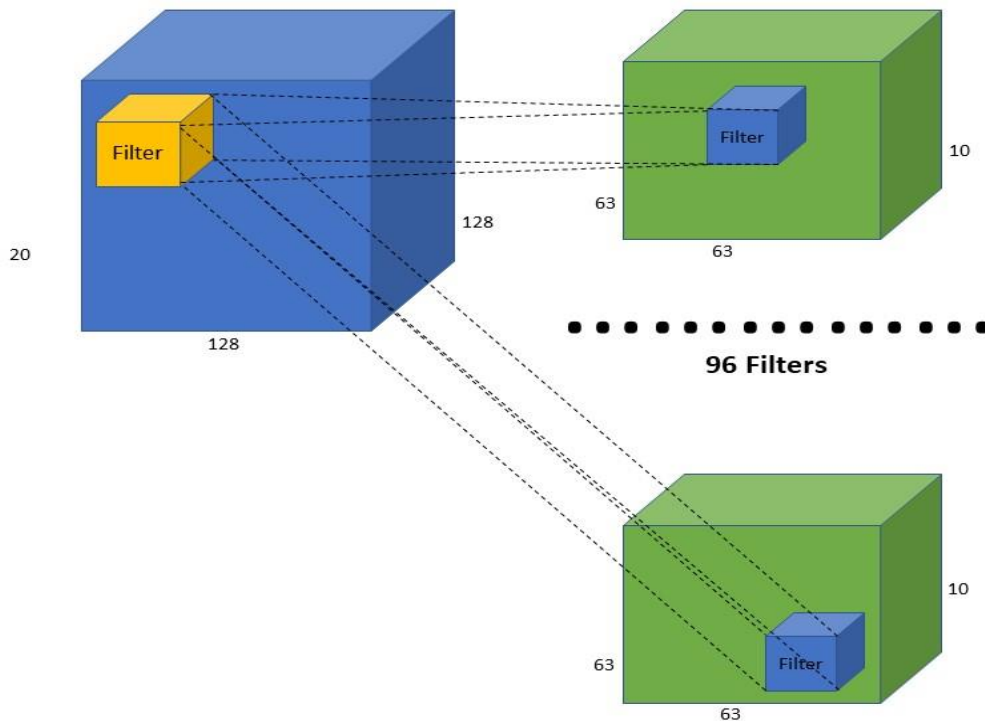
The Layer-1 for the model is Convolution Layer. The kernel size for the layer is (5,5,3) and the Stride is (2,2,2) and the padding is 'SAME'. A rectified linear unit (Relu) activation function is used in the layer 1. The Rectified Linear Unit(Relu) function is as followed:

$$f(x) = \max(0, x) \quad (7)$$

Here,  $f(x)$ = Rectified Linear Unit

$x$ = Input of the function

The above function returns 0 if the input of the function is negative, else it returns the original number. The rectified linear unit function is used for its computational efficiency [16].



**Figure 3.7: 3D convolution layer filter projection**

The Layer-2 is a Max Pooling Layer with kernel size of (3,3,3) and stride of (2,2,2). The padding of the Layer-2 is kept as 'SAME'. The Layer-3 is a Convolution Layer with kernel size of (3,3,3) and stride of (1,1,1). The padding of the layer is 'SAME'. Rectified linear unit(Relu) activation function is used in Layer-3. The Layer-4 is a Max Pooling Layer with the kernel size of (3,3,3) and stride of (2,2,2). The padding of the Layer-4 is 'SAME'. The Layer-5 is a Convolution Layer with kernel size of

(3,3,3) and stride of (1,1,1). The padding of the Layer-5 Convolution Layer is 'VALID'. Here padding 'VALID' means the return output shape is (1,1,1) and no padding will be done on the layer. Rectified Linear Unit (Relu) activation function is used in this Layer. The Layer 6 is a Convolution Layer with kernel size (3,3,3) and stride of (2,2,2). The padding is 'SAME' in this Layer. The reason of using two Convolution Layer without pooling layer is to construct complex features before the destructive Pooling Layer. The Layer-7 of the proposed model is a Convolution Layer with kernel size of (3,3,3) and stride of (2,2,2). Rectified Linear Unit activation function is used in the Layer-5, Layer-6 and Layer-7. The Layer-8 of the model is a Max Pooling Layer. The kernel size is (3,3,3), stride is (2,2,2) and padding is 'SAME'.

The Layer-9, Layer-10, Layer-11 of the proposed model is Fully Connected Layer and Rectified Linear Unit (Relu) activation function is used on each layer. Before the Output Layer which is Layer-12, dropout rate of 0.5 is used. Dropout rate is kept low to minimize overfitting problem [17]. The number of output class is two which is also known as binary classification.

### **3.4 Initialization**

Firstly, the input of the model is initialized as a placeholder. Placeholder is a tensor, which can store some value, and which can be passed between different nodes. The shape of the input tensor is 5-Dimensional. The input tensor value is kept [-1,128,128,20,1]. Here, -1 represents that the tensor can take any number of images. Then, a placeholder for the output of our model is initialized with a 1-dimensional tensor. The placeholder is initialized by ['Float'], which means the placeholder can hold any Float data type values.



After initialization of the placeholder variables, the weights and biases for the model is initialized with random value. At the end, the input image is reshaped to [128,128,20]. The purpose of reshape is to make all input images of same width, height and depth. The maximum number of epoch for the training process is initialized as 50. The learning rate is initialized as .00001. To calculate the cost of our model during training, SoftMax and cross entropy with logits function is used. For the optimization of our model Adam Optimization is used with our initialized learning rate and cost function.

### **3.5 Iteration and Training**

For algorithm for training the model is shown below:

```
full_data=[Di= 1,2,3,.....,N]
```

```
train_data= full_data[0:1550]
```

```
test_data=full_data[-45:]
```

```
learning_rate=.00001
```

```
prediction=Convolution_neural_network()
```

```
cost=tensorflow.reduce_mean(tensorflow.nn.softmax_cross_entropy_with_logits(logits=prediction,labels=y))
```

```
optimizer=tensorflow.train.adamOptimizer(learning_rate).minimize(cost)
```

```
correct=tensorflow.equal(tensorflow.argmax(prediction,1),(y,1))
```

```
accuracy=tensorflow.reduce_mean(correct)
```

with tensorflow.Session() as sess:

```
Max_epochs=50
```

```

Successful_runs=0

Total_runs=0

init= initialize_all_global_variables()

sess.run(init)

for epoch in Max_epochs:

    epoch_loss=0

    for data in train_data:

        X=data[0]

        Y=data[1]

        try:

            loss=sess.run([optimizer,cost],feed_dict={ x:X,y:Y })

            sucessful_runs+=1

            cost+=loss

        Except:

            pass

        total_runs+=1

    accuracy.eval({x: [index[0] for index in test_data],
                    y:[index[1] for index in test_data]})

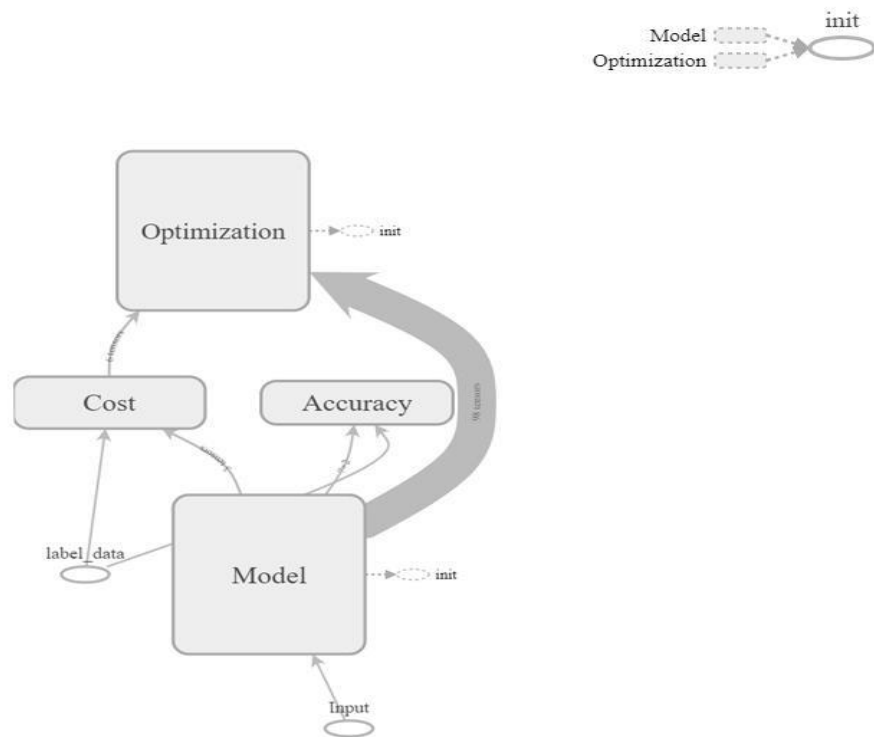
```

# Chapter 4

## Evaluation and Result

### 4.1 Running Environment

The computer used for running and implementation is with Intel Core i5-8500 6 Core CPU processor, 16GB DDR4 Ram with bus speed of 2400Mhz, ADATA NVME 120GB SSD(Boot). Windows 10 64 bit operating system is installed on the computer. The total work is implemented on Python 3.6.1 64 bit and for training and testing TensorFlow 1.8 is used. After running our model in TensorFlow the TensorFlow computational graph for our model is built first. The TensorFlow Computational graph for this model is shown in Figure-4.1.



**Figure 4.1: TensorFlow Computational Graph**

## 4.2 Accuracy Evaluation

The prediction accuracy was calculated using,

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (8)$$

Here, TP and TN are True Positive and True Negative, which denotes the Positive and Negative number of correct classification. On the contrary, FP and FN represents False Positive and False Negative. False positive are the number of Positive wrong classification and false negative are the number of negative wrong classification.

### 3.3 Feature Sizes and Neuron numbers

the size of the input image is (128 x 128 x 20). Let be the input image size is ( $I_1 \times I_2 \times I_3$ ), filter size is ( $F_1 \times F_2 \times F_3$ ) and stride size is ( $S_1 \times S_2 \times S_3$ ). So, if the output size is represented by ( $O_1 \times O_2 \times O_3$ ). The output feature size is,

$$O_i = \frac{I_i - F_i}{S_i} + 1 \quad (9)$$

According to our 3D Convolutional Neural Network of figure- 3.6 the output feature sizes are shown in Table 1.

<b>Table 1: Feature sizes after each Convolution Layer and Max pooling Layer</b>	
<b>Layer Number</b>	<b>Feature size after each layer</b>
Layer 0 (Input Layer)	128 x 128 x 20

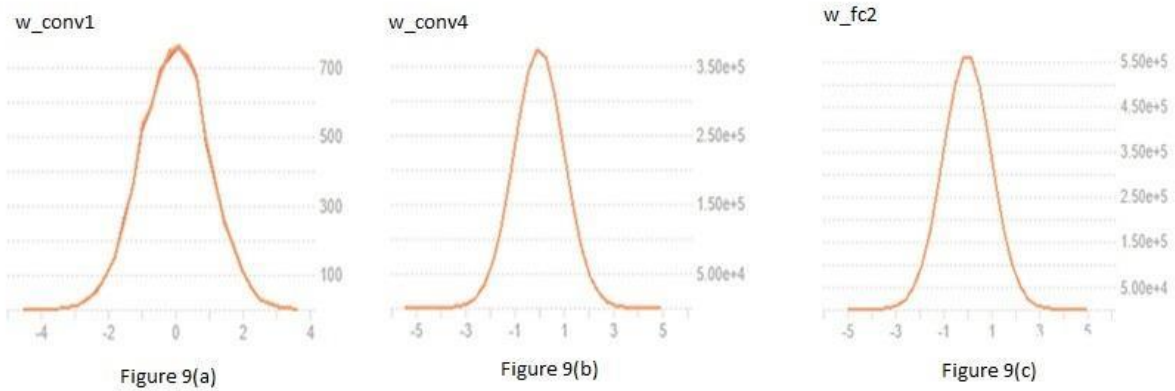
<b>Table 1: Feature sizes after each Convolution Layer and Max pooling Layer</b>	
<b>Layer Number</b>	<b>Feature size after each layer</b>
Layer 1 (Convolution Layer)	63 x 63 x 10 x 96
Layer 2 (Max pooling Layer)	31 x 31 x 5 x 96
Layer 3 (Convolution Layer)	29 x 29 x 3 x 128
Layer 4 (Max Pooling Layer)	14 x 14 x 1 x 128
Layer 5 (Convolution Layer)	14 x 14 x 1 x 256
Layer 6 (Convolution Layer)	14 x 14 x 1 x 384
Layer 7 (Convolution Layer)	14 x 14 x 1 x 256
Layer 8 (Max Pooling Layer)	7 x 7 x 1 x 256

According to our 3D Convolutional Neural Network of Figure- 3.6 the number of input neurons and number of output neurons from each of the Fully Connected Layer are shown in Table-2.

<b>Table 2: Number of input neurons and output neurons from each of the Fully Connected Layers</b>		
<b>Layer Number</b>	<b>Input Neurons</b>	<b>Output Neurons</b>
Layer 9 (Fully Connected Layer)	$7 \times 7 \times 1 \times 256 = 12544$	4096
Layer 10 (Fully Connected Layer)	4096	2048
Layer 11 (Fully Connected layer)	2048	1024
Layer 12 (Output Layer)	1024	2

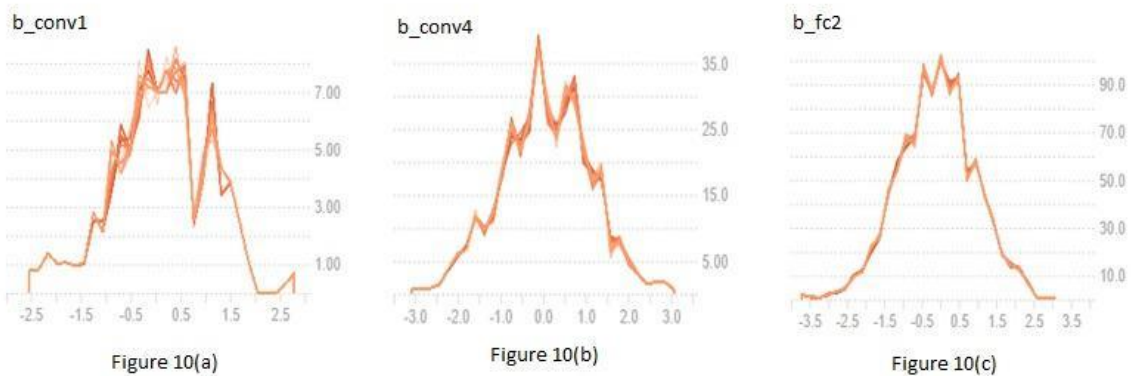
### **4.3 Variables Evaluation**

For visualizing and storing the model weights, biases, activation function we have used Tensor board Histogram. the Figure-4.2 shows the weights of different layers.



**Figure 4.2: (a) weights of convolution layer 1, (b) weights of convolution layer 4, (c) weights of fully connected layer 2**

Model biases are used to add some constant value with the weight. If in a layer the value came as 0, bias will add some value with it. so, the model converges quickly. Figure-4.3 shows the biases of different layers:



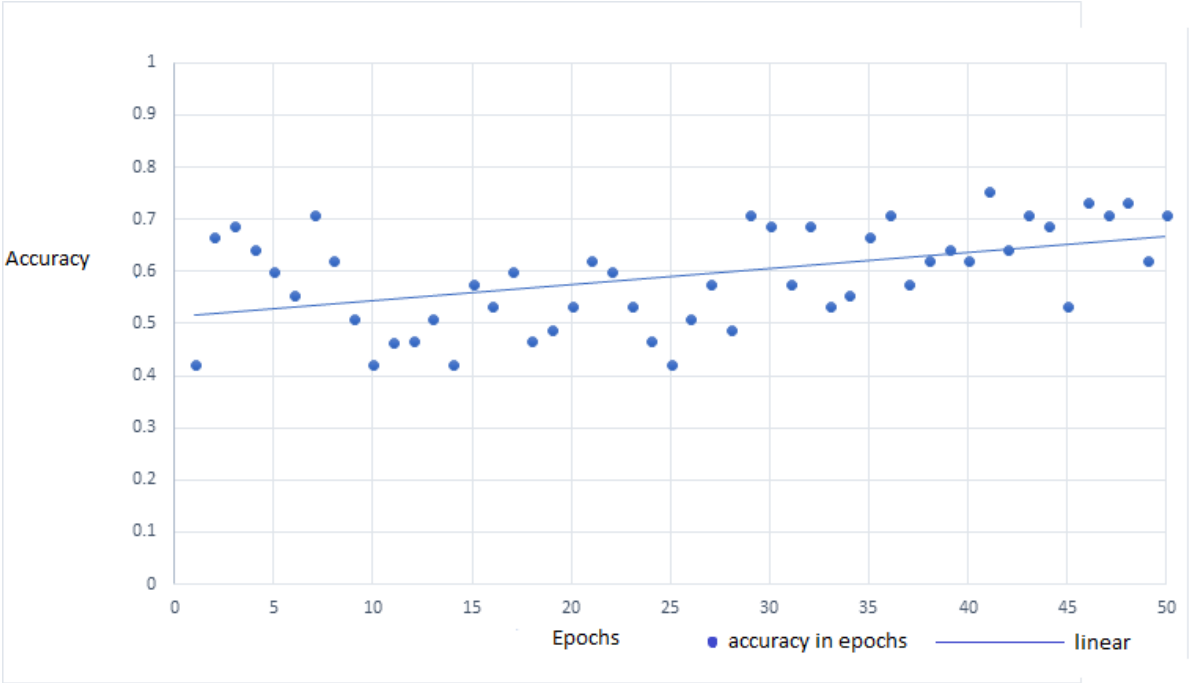
**Figure 4.3: (a) biases of convolution layer 1, (b) biases of convolution layer 4, (c) biases of fully connected layer 2**

From the Figure-4.2 and Figure-4.3 we can see that; the distribution of model weights and biases are between -2 Standard deviation to +2 Standard Deviation.

### 4.4 Validation Results

The network took 192 hours to train and test the deep learning model with the segmented lung CT volume with 1550 as train set and 45 as test set and validation set. the final accuracy is 77.78 percent and final error rate is 22.22 percent.

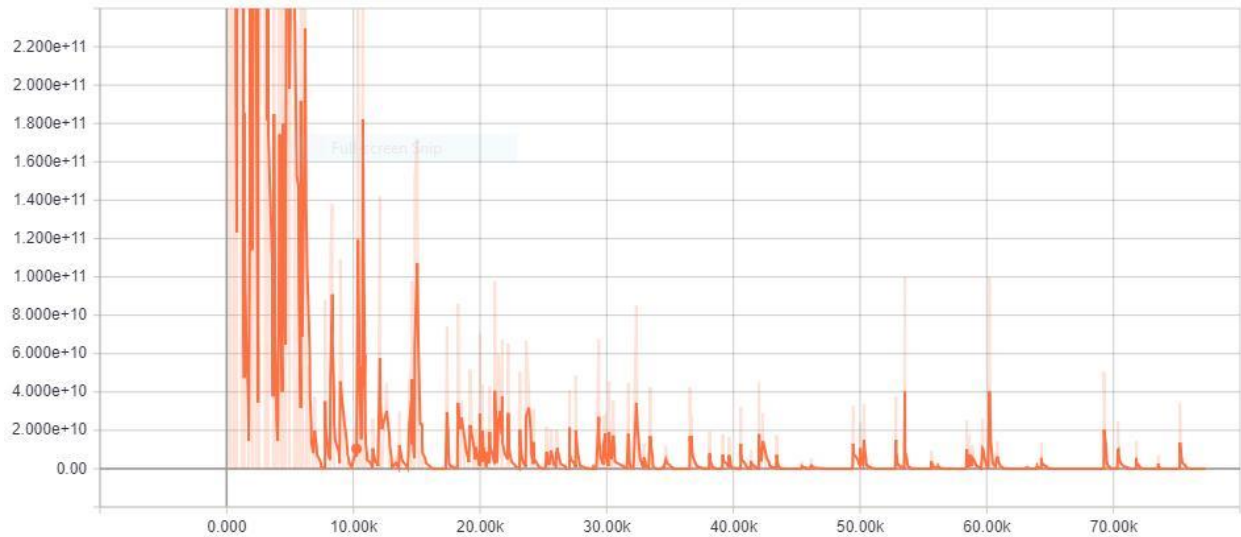
The accuracy after each epoch is shown on Figure-4.4.



**Figure 4.4: Accuracy after each epoch**



Figure-4.5 shows the amount of loss calculated by our cost function.



**Figure 4.5: Amount of loss calculated during training**

From the figure we can see that, the later runs has fewer cost than the starting cost. The figure shows the relative cost of all other runs in respect to the minimum cost.

## Chapter 5

### Discussions

#### 5.1 Accuracy and prediction

we only used the segmented lung CT volume and the ‘cancer/non-cancer’ label of the scans. So, there is no information regarding the malignant nodules. our deep learning model extracts features of malignant nodules from the segmented CT volumes and predicted the scan as ‘cancer/noncancer’. Out of 45 cases our model predicted 35 correctly. but the model failed to predict 10 cases correctly. The prediction from our model is shown in table 3.

Patient id	Original Label	Prediction	Error type
026470d51482c93efc18b9803159c960	1	1	
031b7ec4fe96a3b035a8196264a8c8c3	1	1	
03bd22ed5858039af223c04993e9eb22	0	0	
06a90409e4fcea3e634748b967993531	0	0	
07b1defcfae5873ee1f03c90255eb170	1	0	False Negative
0b20184e0cd497028bdd155d9fb42dc9	0	0	
12db1ea8336eafaf7f9e3eda2b4e4fef	0	0	
159bc8821a2dc39a1e770cb3559e098d	0	1	False Positive
174c5f7c33ca31443208ef873b9477e5	0	0	
1753250dab5fc81bab8280df13309733	0	0	
1cf8e778167d20bf769669b4be96592b	0	0	
1e62be2c3b6430b78ce31a8f023531ac	1	1	
1f6333bc3599f683403d6f0884aefe00	0	0	
1fdbc07019192de4a114e090389c8330	1	0	False Negative
2004b3f761c3f5dfffb02204f1247b211	0	0	
202898fa97c5949fbdc07ae7ff1cd9f0	0	0	
21b73c938fd7d346ee77a60bd60aaeac	0	0	
243038f7bb7787497c59bc17f04c6ed9	0	0	
26142353f46d20c9fdded93f01e2bff4	0	0	
263a1c3bfa43556623e75ed901e3fd8f	1	0	False Negative
2703df8c469906a06a45c0d7ff501199	1	1	
2a3e6ecf9499607ef4fd14b436136b0c	0	0	

Table 3: Prediction of Cancer from our Model			
Patient id	Original Label	Prediction	Error type
2d596b6ead89ab35577fe625a9a17cbb	1	1	
2eb92d17ca91b393765e8acf069763a6	1	0	False Negative
2f77fd993fbd858dec3c085b9ff1a3a2	1	1	
3295cec04482210dc6f78c2b4a1d287b	0	0	
33387bea2cacf6127035cc7033036a02	0	0	
34037914ceeec5605fc890159dd425c5	1	1	
38bf066bba822584e14c0af65d4bb5e9	0	1	False Positive
3ee1fd6a0f3f108c3558e6699fb011f2	0	0	
42b2161e43b4dd0ea94604485976c59c	0	0	
4434e19303b62ebaecf2596583ff351	0	0	
4575fe61bf3f536ce6cf26fcc2893c	1	1	
48ab0b98fc7789304c21430978624f32	0	0	
49433c1588cc078b825a0eff1dc2e816	0	0	
49c88f7cc77341c9ae4e64243f9912fc	0	1	False Positive
4b28f147cb82baba3edcdbc34ca19085	0	1	False Positive
505405b3e70fb24b92e6a8a5b7ed339c	0	0	
50cdacec399071cf70d8badd2511d0b3	0	0	
519ad4ead3e61d2d71088ac8e46f25b6	0	0	
52f6d741e674f62fbcf73e6ec4f6a472	0	0	
538543b57d0c8fa0b2b6bb7c84df3f33	0	1	False Positive
5451203688c930484ba1f3c7f1378847	0	0	
55b06d60e7c0329787f81d1b7cbf9aa0	0	1	False Positive
567547810a1795b9c8e11c15dfd32c34	1	1	

From the table-3, there are 10 errors during our test case of our model. Out of 10 there are 6 False Positive and 4 False Negative Error. The final accuracy during testing of our model is 77.78%.

## 5.2 Confusion Matrix

A confusion matrix is used to describe the classification performance and accuracy of a model on a test set. In a binary confusion matrix, the predicted true positive and true negative defines the correct prediction for a positive class and negative class. On the other hand, False positive and false negative defines the false prediction of a

negative class and a positive class. The confusion matrix of our model is shown in Table 4.

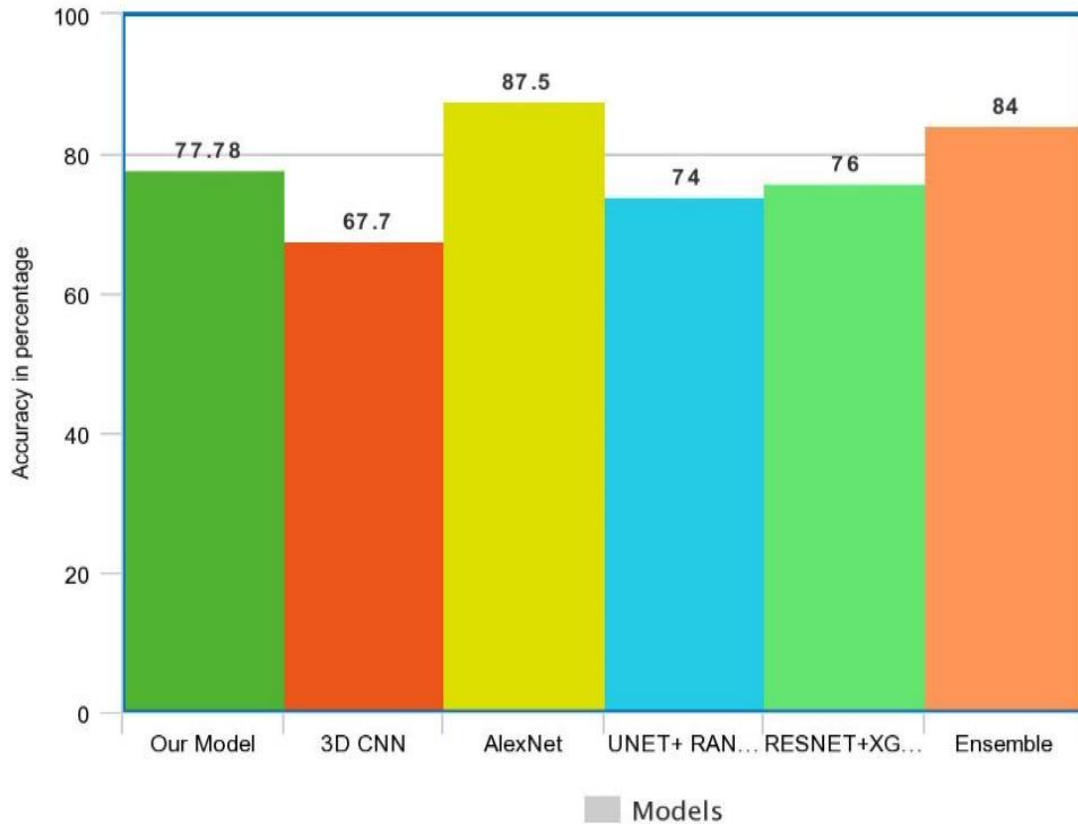
<b>Table 4: Confusion Matrix</b>			
<b>Actual Class</b> <b>Cancer = Yes</b> <b>Non- Cancer= No</b>	<b>Predicted Class</b> <b>Cancer= Yes, Non-Cancer= No</b>		
	<b>Test Set</b> <b>N=45</b>	<b>Yes</b>	<b>No</b>
	<b>Yes</b>	9	4
	<b>No</b>	6	26

From the table-4, the number of true positive class is 9 and false positive is 6. Also, the number of true negative is 26 and false negative is 4. So, the probability of true positive is 0.69 and the probability of false positive is 0.1875. The probability of true negative is 0.8125.

<b>Table 5: Prediction Analysis</b>				
<b>Deep 3D</b> <b>CNN Model</b>	<b>Sensitivity</b>	<b>Specificity</b>	<b>PPV</b>	<b>False Positive</b> <b>Rate</b>
	0.69	0.8125	0.6	0.1875

### 5.3 Comparative Study

The comparison between the related works in the field of cancer detection using Convolutional Neural Networks described in chapter-2 and our model is shown in the Histogram Figure 5.1. The accuracy acquired by a 3D convolutional neural network is 67.7% [21].



**Figure 5.1: Comparative study between our model and different models**

Accuracy acquired by a 3D Alex Net architecture is 87.5% [5]. The work from [22], the accuracy acquired by the Unet and Random forest architecture is 74%. The ResNet and XGBoost architecture accuracy is 76% and the accuracy from an Ensemble architecture is 84% [22]. The histogram of Figure 5.1 shows the comparative study between our model and different models [5] [21] [22].

# Chapter 6

## 6.1 Conclusion

Computer-aided-diagnosis has become an important research area as the number of cancer-affected people is increasing very rapidly. so, computer-aided-diagnosis can help the oncologist and medical personals by detecting early stage lung cancer and other cancers as well. The work in this paper, summarizes the processing techniques of lung CT images and 3D deep Convolutional Neural Network for the detection of lung cancer. Machine Learning in Computer-aided-diagnosis is a hot topic in the field of Computer Science and Artificial Intelligence since the last decade. in this work, an attempt was made to contribute in the field of Computer-aided-diagnosis systems. A 3D deep Convolutional Neural Network has been constructed in this work and processing techniques of CT image data. we hope that, this work can be used to detect early stage lung cancer. The accuracy acquired by our work is 77.78 percent. Machine Learning and Deep Neural Network is emerging day-by-day. So, our work can be used in Other cancer detection like Breast Cancer, Prostate Cancer in the near future. The 3D deep convolutional neural network used in the work is an improvement of the tradition 2D Convolutional Neural Network. We hope that our work can be used in lung cancer detection as well as other types of cancer detection.

## 6.2 Future work

These Paper describes the processing techniques of lung CT images and deep 3D convolutional neural network. However, there is scope for future work on this topic. In our work, we detected lung cancer, but we haven't show the location of the cancerous nodule. In future, our concern will be detecting the cancerous nodules and also the size of the cancerous nodule. The Dataset used for this work is small. But we believe that, by collecting more dataset can improve the performance of our

model. Also, our deep 3D convolutional neural network can be used for other cancer detection model, we want to detect other types of cancer. We also want to study the features used for cancer detection by TensorFlow. In addition, with that we want to study the exact feature by which our model detected cancer. At least we want to run our model for different sizes of train data and test data and differentiate the performance of our model and also improve the accuracy of cancer detection.

## References

- [1] J. Alford. "World Cancer Day 2017." *Global lung cancer rates reveal tobaccos dark shadow is looming.* Available: <http://scienceblog.cancerresearchuk.org/2017/02/06/world-cancer-day-2017-global-lung-cancer-rates-reveal-tobaccos-dark-shadow-is-looming/> [Feb.6, 2017]
- [2] A. H. S. A. S. Islam, "Early Lung Cancer Detection, Mucosal, and Alveolar Imaging." *Interventional pulmonology*, vol. 22, no. 3, pp. 271-280, 2016.
- [3] A.Verma; A.Y.H.Lim and S.K.Goh, "Medicine (Baltimore)", Timeliness of Diagnosing Lung Cancer: Number of Procedures and Time Needed to Establish Diagnosis, vol.104, pp. 1791-1797, Oct.21 2017.
- [4] A. Krizhevsky; I. Sutskever and G. E. Hinton, "Image Net Classification with Deep Convolutional Neural Networks", pp. 1-9, 2010.
- [5] T. Jin; H. Cui; S. Zeng and X. Wang, "Learning deep spatial lung features by 3D convolutional neural network for early cancer detection", *Lung Cancer prediction*, pp. 1-6, 2017.
- [6] M.Nielsen,Y.Bengio and A.Couville, *Deep Learning*. USA: Princeton University Press, Dec. 2017.
- [7] K.Kavukcuoglu and A.Graves, "International journal of science", *Human-level control through deep reinforcement learning*, p.529-533, 2015[Online]Available: <https://www.nature.com/articles/nature14236/> [Accessed Feb.26, 2015]
- [8] G.Hinton, L.Deng and D.Yu, "IEEE signal processing Magazine", *Deep Neural Networks for Acoustic Modeling in Speech Recognition*, vol.29, p.82-97, Oct.18 2012.Available: <https://ieeexplore.ieee.org/abstract/document/6296526/>
- [9] S.Saha, "Towards Data Science", *Neural Networks: All you need to know*, Apr.07, 2015.Available: <https://towardsdatascience.com/nns-aynk-c34efe37f15a>
- [10] J.Schmidhuber, "Neural Networks", *Deep learning in neural networks: An overview*, vol.61, p.85-117,



2014[Online]<https://doi.org/10.1016/j.neunet.2014.09.003> [Accessed Oct.13, 2014]

[11] M. Firmino et al., “Computer-aided detection system for lung cancer in computed tomography scans: Review and future prospects,” *Biomedical Engineering On-line*, vol. 13, pp. 41, 2014.

[12] S. Hawkins et al., “Predicting malignant nodules from screening CT scans,” *Journal of Thoracic Oncology*, vol. 11, pp. 2120-2128, 2016.

[13] J. Mulholland; A. Sander and B. A. Hamilton, “Data Science Bowl 2017”, *U-Net Segmentation Approach to Cancer Diagnosis*, Available: <https://www.kaggle.com/c/data-science-bowl-2017/data>, [Feb. 1, 2017]

[14] H.Chen, “Radiology Data Quest”, *Dicom Processing and segmentation in Python*, Available: <https://www.raddq.com/dicom-processing-segmentation-visualization-in-python/>, [Jan.28, 2017]

[15] S. Khalid; A. C. Shaukat; A. Jameel and I. Fareed, "Segmentation of Lung Nodules in CT scan Data: A Review," in *Medical Imaging: Concepts, Methodologies, Tools, and Applications: IGI Global*, pp. 601-613, 2017.

[16] V. Nair and G.E. Hinton, “Restricted Boltzmann Machine”, *Rectified Linear Units Improve Restricted Boltzmann Machines*, pp.1930-1931, 2006.

[17] N. Srivastava; G. E. Hinton; A. Krizhevsky; I.Sutskever and R. Salakhutdinov, "Dropout: a simple Way to prevent neural networks from over fitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[18] A.Chon; N.Balachandar and P.Lu, “CAD System for lung cancer classification of CT scans with unmarked nodules”, *Deep Convolutional Neural Networks for Lung Cancer Detection*, pp.1-9, 2017.

[19] Z.Wang; H.Xu and M.Sun, “10<sup>th</sup> International Symposium on Computational Intelligence and Design”, *Deep Learning Based Nodule Detection from Pulmonary CT Images*, pp.1-4, 2017.

[20] P.Rao; N.A.Pereira and R.Srinivasan, “2<sup>nd</sup> International Conference on Contemporary Computing and Informatics (ic3i)”, *Convolutional Neural Networks for Lung Cancer Screening in Computed Tomography (CT) Scans*, pp.489-493, 2016.

[21] L.Fan, Z.Xia and X.Zhang, “Lung Nodule Detection Based on 3D Convolutional Neural Networks,” Northwestern Polytechnical University, Shannxi, China,2014.

[22] S.Bhatia, Y.Sinha, L.Goel, “Lung Cancer Detection: A Deep Learning Approach,” BITS, Pilani,India,2017.