

# **Evaluation of VGG & ResNet Very Deep Convolutional Neural Networks for Detecting Lung Cancer in CT Scans**

W.P.H.D. Perera

168253V

This dissertation submitted in partial fulfillment of the requirements for the Degree of MSc in  
Computer Science specializing in Data Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

**May 2018**

## DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:.....

Date:.....

Name: W.P.H.D.Perera

The supervisor/s should certify the thesis/dissertation with the following declaration.

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the CS5999 PG Diploma Project.

Signature of the supervisor: .....

Date: .....

Name: Dr. Charith Chitraranjan

## ABSTRACT

Lung cancer is the second most common destructive cancer in the world. It is important to detect Lung cancer at its earliest possible time because this dreadful illness spreads in a rapid pace weakening and killing the entire body of a human. The lung cancer identification process is not easy as its symptoms are visible to outside mostly at its final stage. Lung cancer nodules are detected by radiologists through CT (Computed tomography) scans, but there is a high probability to fail to spot where the actual lung cancer nodule is because, the lung lesions are low in contrast. Therefore, there should be a Computer Aided Diagnosis (CAD) system to assist radiologists in identifying lung nodules efficiently, accurately so the results given by the CAD systems can be taken as a second opinion to detect lung nodules for radiologists. Accurate CAD systems can improve the quality and productivity of radiologists' image interpretation. There are many research subjects ongoing in medical imaging and diagnostic radiology. But it is needed to continuously improve the accuracy and the consistency of radiological diagnoses because still there are high false positive rates associated with CAD system results.

Current CAD systems have been developed using two main different approaches. First is the conventional old framework which detects lung cancer nodules using manual feature extraction and conventional image preprocessing approach. New approach is the Deep Neural Network architecture which automatically and directly uncovers features from the training data. In this approach the three steps, feature extraction, selection and supervised classification have been realized within the optimization of the same deep architecture.

This research evaluates two existing very deep learning architectures, Resnet-50 and VGG19 for learning high-level image representation to achieve high classification accuracy with low variance in medical image binary classification tasks. The classification accuracy was performed with two different datasets, NDSB and LUNA16+LIDC. For NDSB dataset the Restnet-50 model outperformed VGG19 model by giving the accuracy, sensitivity and specificity as 68%, 73%, 65% respectively. And for LUNA16+LIDC dataset 80.3%, 79.8%, 80.6% results were obtained for accuracy sensitivity and specificity out performing again the VGG19 network architecture.

## **ACKNOWLEDGEMENT**

I would like to express profound gratitude to my advisor, Dr. Charith Chitraranjan, for his invaluable support by providing relevant knowledge, materials, advice, supervision and useful suggestions throughout this research work. His expertise and continuous guidance enabled me to complete my work successfully.

Last but not least, I present my appreciation to my family and my friends who were behind me, encouraging and directing me towards the success of my project.

## Table of Content

<b>DECLARATION.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT.....</b>	<b>iii</b>
<b>LIST OF FIGURES.....</b>	<b>vii</b>
<b>LIST OF TABLES.....</b>	<b>x</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1. Chapter Overview .....	1
1.2. Project Background.....	2
1.3. Problem .....	3
1.4. Motivation .....	4
1.5. Objectives.....	4
1.6. Pathogenesis.....	5
1.7. Diagnosis.....	5
1.8. Screening.....	6
1.9. Lung Nodule Detection .....	7
1.10. Detection and classification of lung nodules on digital chest radiographs.....	10
<b>2. Literature Review.....</b>	<b>12</b>
2.1. Nodule location detection.....	12
2.2. Lung Segmentation and Boundary Correction Method .....	17

2.3.	Lung cancer detection using artificial neural networks .....	19
2.4.	Evaluation of existing deep neural network architectures .....	24
2.4.1.	The advantages of deep neural networks over conventional old frameworks .....	25
2.4.2.	LeNet 5.....	25
2.4.3.	Dan Ciresan Net.....	27
2.4.4.	AlexNet.....	27
2.4.5.	Overfeat.....	28
2.4.6.	VGG.....	28
2.4.7.	Network-in-network.....	30
2.4.8.	GoogLeNet and Inception.....	31
2.4.9.	Bottleneck layer .....	32
2.4.10.	Inception V3 (and V2).....	33
2.4.11.	Multi-Column deep neural network .....	35
2.5.	Lung cancer detection using convolutional neural networks .....	38
<b>3.</b>	<b>Methodology .....</b>	<b>43</b>
3.1.	Thoracic CT scan images reading .....	44
3.1.1.	Lung Image Reading.....	45
3.2.	Lung Extraction and Preprocessing.....	45
3.3.	Nodule Candidate / ROI generation .....	46
3.4.	Classifying using CNN.....	46
3.4.1.	Develop model approach .....	48

3.4.2.	Pre-trained model approach .....	49
<b>4.</b>	<b>Implementation .....</b>	<b>54</b>
4.1.	Chapter Overview .....	54
4.2.	Using NDSB 2017 dataset to detect lung cancer .....	54
4.2.1.	Thoracic CT scan images reading .....	54
4.2.2.	Lung segmentation approach .....	58
4.2.3.	U-Net for Lung Nodule Segmentation.....	66
4.2.4.	Malignancy Classification .....	69
4.3.	Using LUNA16 + LIDC dataset to detect lung cancer .....	74
4.3.1.	CT scan image reading.....	76
4.3.2.	Getting the Nodule position of the mhd files.....	76
4.3.3.	Isolation of the Lung ROI to narrow the nodule search.....	79
4.3.4.	Loading the Segmenter .....	83
4.3.5.	Training the Segmenter.....	83
4.3.6.	Malignancy classification .....	83
<b>5.</b>	<b>Evaluation.....</b>	<b>85</b>
5.1.	Chapter Overview .....	85
5.2.	Results and Discussion.....	85
5.2.1.	Using the initial neural network only as the feature extractor .....	85
5.3.	Evaluation Results for NDSB 2017 .....	89
5.3.1.	Training only last few layers.....	91

5.3.2. Freeze, Pre-train and Finetune (FPT).....	93
5.4. Evaluation Results for LUNA16 dataset.....	99
5.5. Performance Evaluation with Non-3D CNN -NDSB 2017 .....	106
5.6. Performance Evaluation with 3D CNN and Other types -NDSB 2017 .....	107
5.7. Performance Evaluation Results for Luna16 .....	109
<b>6. Conclusion .....</b>	<b>114</b>
<b>7. Future works .....</b>	<b>115</b>
<b>8. References.....</b>	<b>116</b>

#### LIST OF FIGURES

Figure 1-Signatures of real nodules from the JSRT database.....	14
Figure 2-Nodule visibility test before(A) and after(B) Applying Wiener Filter .....	15
Figure 3- Result of circular object detection.....	16
Figure 4-Flowchart for the process of lung cancer detection using FCM and HNN.....	19
Figure 5-Binarization check method.....	21
Figure 6-Region of interest .....	23
Figure 7-Seed fill operation .....	23
Figure 8-Input image.....	23
Figure 9-Segmented image .....	23
Figure 10-Classification of cancerous images .....	23



Figure 11-Reporting Top-1 accuracy vs. amount of operations required for a single forward pass in multiple popular neural network architectures .....	25
Figure 12-LeNet5 architecture .....	26
Figure 13-AlexNet architecture .....	27
Figure 14- VGG Network .....	29
Figure 15-Network in Network architecture .....	30
Figure 16-Inception module.....	31
Figure 17-3x3 convolution layer.....	34
Figure 18-New inception model .....	34
Figure 19-New inception model .....	35
Figure 20-(a)-DNN architecture, (b)-Training a DNN, (c)-MCDNN architecture.....	37
Figure 21-Nodule candidates are fed to the network composed of six alternating convolution and max-pooling layers, and two fully connected layers .....	39
Figure 22- CNN architecture from the input image of size 50*50*1 to the final output.....	40
Figure 23- Architecture of CNN for nodule recognition. The number of neurons in layers are set to 6272, 1568, 1600, 250, 150, 100 and 50.....	41
Figure 24: Methodology proposed by the Research .....	44
Figure 25 : A residual block.....	50
Figure 26 : Patient wise Folder Structure .....	56
Figure 27 : Original images in DICOM format .....	57
Figure 28: HU results of the results observed from Wikipedia related to different Substances contained in the CT images.....	58
Figure 29 : View of a Slice .....	59

Figure 30 : View of Vessels and Airways etc.....	60
Figure 31 : Output Image.....	60
Figure 32 : Cylindrical Scanning Bounds .....	60
Figure 33 : Results in Hounsfield units.....	61
Figure 34 : Patient 3D scan with pixels greater than 400HU .....	62
Figure 35 : Patient Bronchiole within lung.....	62
Figure 36 : Patient initial mask with no air.....	63
Figure 37 : Patient final mask when bronchioles are included .....	63
Figure 38 : Original 2D slice of a patient .....	64
Figure 39 : Lung segmentation mask by thresholding of sample patient .....	64
Figure 40 : Final watershed segmentation mask of sample patient .....	65
Figure 41 : Final watershed lung segmentation of a sample patient.....	65
Figure 42 : U-Net Architecture .....	67
Figure 43 : U-Net sample input from LUNA16 validation set which is a segmented lung slice	67
Figure 44 : Figure-U-Net predicted output from LUNA16 validation set.....	68
Figure 45 : U-Net sample labels mask from LUNA16 validation set showing ground truth nodule location.....	68
Figure 46 : Representation of a ResNet CNN with an image from Imagenet .....	72
Figure 47 : Workflow of the Proposed Solution.....	73
Figure 48; Original CT slice on the left and its lung nodule mask on the right .....	78
Figure 49: Masked Slice which highlight the nodule .....	78

Figure 50: Pixel Histogram.....	79
Figure 51 : Pixel value histogram of two random slices.....	80
Figure 52: Segmented lung images through Thresholding.....	81
Figure 53: Final Lung Mask .....	82
Figure 54 : The layers of VGG neural network .....	91
Figure 55 : Training loss vs Validation loss for the 15 layers freezed VGG model.....	93
Figure 56 : Training loss vs Validation loss for FPT approach, Resnet-50 model.....	95
Figure 57 : Training accuracy vs Validation accuracy for FPT approach, Resnet-50 model.....	96
Figure 58 : Training loss vs Validation loss for FPT approach, VGG19 model .....	97
Figure 59 : Training accuracy vs Validation accuracy for FPT approach, VGG19 model .....	98
Figure 60 : Model Accuracy and loss for FPT approach.....	102
Figure 61 : Model Accuracy for VGG19 FPT approach .....	104

## LIST OF TABLES

Table 1 : Confusion Matrix.....	89
Table 2 : Diagnosis Test Results.....	89
Table 3 : Confusion Matrix.....	92
Table 4 : Diagnosis Test Results.....	92
Table 5 : Confusion Matrix.....	94
Table 6 : Diagnosis Test Result .....	95
Table 7 : Confusion Matrix.....	97

Table 8 : Diagnosis Test Result .....	97
Table 1 : Confusion Matrix.....	99
Table 2 : Diagnosis Test Results.....	99
Table 9 : Confusion Matrix.....	100
Table 10 : Diagnosis Test Results.....	100
Table 11 : Confusion Matrix.....	101
Table 12 : Confusion Matrix.....	104
Table 13: Diagnosis Results.....	105
Table 14 : Performance Evaluation with Non-3D CNN.....	106
Table 15 : Performance Evaluation with 3D CNN and Other types.....	108
Table 16 : Performance Evaluation Results for Luna16.....	111
Table 17 : Comparison of VGG19 model with their different transfer learning approaches .....	112
Table 18 : Comparison of ResNet-50 model with their different transfer learning approaches.	112
Table 19 : Comparison of VGG19 model with their different transfer learning approaches .....	113
Table 20 : Comparison of ResNet-50 model with their different transfer learning approaches.	113

# 1. Introduction

## 1.1. Chapter Overview

Lung cancer (both small cell and non-small cell) is the second most common cancer for both men and women. In men, prostate cancer is more common, while in women breast cancer is more common. About 14% of all new cancers are lung cancers. This can be characterized by abnormal cell growth in the lung tissues. Chest radiography (CXR) or a computerized tomography (CT) is typically one of the first diagnostic steps performed by radiologists [1] to identify unnecessary cell growths. Lung nodules grow as small masses of tissue, which appears as white shadows on CXR image or CT scan. There are mainly two types of nodules named Benign which approximately 5-30 millimeters and larger nodules which are likely to be malignant. [2]

Chest radiography is one of the primary diagnostic imaging procedures for evaluating pulmonary diseases such as lung nodules because it is economically easy to use. But this has become a difficult task for radiologists to detect and characterize lesions on a chest radiograph when those lesions are low in contrast and overlap with ribs and large pulmonary vessels.

Advances in imaging modalities and Computer-Aided Diagnosis (CAD) systems have helped radiologists to identify lung nodules more quickly and accurately. They have identified potential locations of nodules automatically by considering masses of expected size, shape and texture of lung nodule. Techniques available for detecting lung nodules generally involve three main stages: lung segmentation, candidate detection and nodule classification.

Apart from that there are many researches that had been conducted to improve the existing lung cancer identification techniques. The main contribution of this work is to evaluate on how Very Deep Convolutional Neural Networks (CNN) for lung nodule classification can be performed since deep learning approaches have been rarely used for nodule classification.

## 1.2. Project Background

Lung cancer is also known as lung carcinoma, is a malignant lung tumor characterized by uncontrolled cell growth in tissues of the lung. If it is left untreated, the growth spreads beyond the lung by the process of metastasis into nearby tissue or other parts of the body. Most cancers that start in lung, known as primary lung cancers also can be named as carcinomas. Mainly there are two types of lung cancers named, small-cell lung carcinoma(SCLC) and non-cell lung carcinoma(NSCLC).

Weight losses, coughing with blood, shortness of breath and chest pain are the most common symptoms of lung cancers. Many of the symptoms of lung cancer are not specific. In many cases, people get detected with lung cancer when the cancer had already been spread beyond the original site by the time they have symptoms and seek medical attention.

Eighty five percent (85%) of clear majority of lung cancers are due to long term tobacco smoking and about 10%-15% cases occur in people who had never smoked.

Genetic factors and exposure to radon gas, asbestos, passive smoking or other form of air pollution are the cases for reasoning 10%-15% cancer occurrence. Lung cancers are detected from chest radiographs and computed tomography (CT). Later the diagnosis is confirmed by a biopsy which performs by bronchoscopy or CT-guidance.

Worldwide in 2012, lung cancer occurred in 1.8 million people and resulted in 1.6 million deaths. This makes it the most common caused cancer related deaths in men and women most common in women after breast cancer.

### **1.3.Problem**

Detecting malignant nodules of the lung is an important task for a CAD system, as the outcome of it helps the oncologists to identify whether the patient is either cancerous or not, the exact locations where the cancer is located at and the severity of the cancer. Deep Convolutional neural networks have been successfully run for scanned medical images aiming many forms of cancers, for example prostate, breast, blood etc.

Deep Learning is a drastically improving area. There are enough spaces for the performance of CAD systems to be improved with these rapidly growing deep neural network architectures. Different deep neural network architectures have shown promising results on identifying lung cancer. However, their effectiveness has not been compared with each other under a controlled experimental setup. VGG and ResNet are two popular Very Deep Convolutional Neural Networks that have shown promising results for object recognition in natural images. However, they have not been evaluated in CT slices for Lung Cancer Detection.

In this research, we evaluate the effectiveness of different deep neural networks, VGG and ResNet on detecting lung cancer in lung CT scans/slices. Hence, we believe that using these well-designed networks will enhance the ability of detecting malignant nodules with a high accuracy, sensitivity and specificity.

## **1.4.Motivation**

Motivation behind this research is to identify lung cancer as early as possible since there is a great chance of surviving from the cancer by going in for treatments early. It is important to invest on CAD systems for early lung cancer detection confirming and assisting Radiologists' opinion.

Traditional process of detecting cancer nodules is not as accurate as detecting nodules using deep neural networks. Hence, need for automating solutions with well performing deep learning techniques, have been an essential factor to improve CAD systems now days. Improving the automated CAD systems will help in different ways for medical field as shown below.

- Improve the quality of cancer detection
- Improve the productivity of radiologist
- Improve the accuracy and the consistency
- Ability to get the output of the improved system as a second opinion

Another major motivation factor for this research is, vast improvement in deep neural networks involved in computer vision. This has helped researchers to carry out their respective work in neural network and perform well. Even though convolutional neural networks have been used in areas other than medicine, in lung nodule detection there are ample space for deep leaning approaches to be evaluated.

## **1.5.Objectives**

- To identify and evaluate deep neural network architectures and learning techniques suitable for detecting lung cancer nodules
- To use different pre-processing techniques for detecting lung cancer nodules



## **1.6.Pathogenesis**

Like other cancers, lung cancer occurs by the activation of oncogenes or inactivation of tumor suppressor genes. Carcinogens cause mutations in these genes which include the development of cancer.

Epigenetic changes such as alteration of DNA methylation, histone tail modification, or microRNA regulation may lead to inactivation of tumor suppressor genes.

The epidermal growth factor (EGFR) regulates cell proliferation, apoptosis, angiogenesis and tumor invasion. Mutation and amplification of EGFR are common in non-small cell cancer and provide the basis for treatment with EGFR-inhibitors.

## **1.7.Diagnosis**

When a person is reported symptoms which supports lung cancer, the first investigation step to be followed is performing chest radiograph.

This may reveal an obvious mass, widening of the mediastinum (suggestive of spread to lymph nodes there), atelectasis (collapse), consolidation (pneumonia) or pleural effusion. CT imaging is typically used to provide more information about the type and extent of disease. Bronchoscopy or CT-guided biopsy is often used to sample the tumor for histopathology.

Lung cancer often appears as a solitary pulmonary nodule on a chest radiograph. However, the differential diagnosis is wide. Many other diseases can also give this appearance, including metastatic cancer, hamartomas and infectious granulomas such as tuberculosis, histoplasmosis and coccidioidomycosis.

Lung cancer can also be an incidental finding, as a solitary pulmonary nodule on a chest radiograph or CT scan done for an unrelated reason. The definitive diagnosis of lung cancer is based on histological examination of the suspicious tissue in the context of the clinical and radiological features.

Clinical practice guidelines recommend frequencies for pulmonary nodule surveillance. CT imaging should not be used for longer or more frequently than indicated as extended surveillance exposes people to increased radiation.

## **1.8. Screening**

Cancer screening uses medical tests to detect disease in large groups of people who have no symptoms. For individuals with high risk of developing lung cancer, computed tomography (CT) screening can detect cancer and give a person options to respond to it in a way that prolongs life. This form of screening reduces the chance of death from lung cancer by an absolute amount of 0.3% (relative amount of 20%). High risk people are those age 55-74 who have smoked equivalent amount of a pack of cigarettes daily for 30 years including time within the past 15 years.

CT screening is associated with a high rate of falsely positive tests which may result in unneeded treatment. For each true positive scan there are about 19 falsely positives scans. Other concerns include radiation exposure and the cost of testing along with follow up.

Chest radiography is one of the primary diagnostic imaging procedures to evaluate pulmonary diseases such as lung nodules and many other diseases. But the issue is, it is sometimes difficult for radiologists to detect and characterize lesions on a chest when the lesions are low in contrast, overlap with ribs and large pulmonary vessels. To improve the radiologists' performance for evaluating pulmonary diseases in digital chest radiography, many investigators have developed Computer Aided Diagnosis (CAD) schemes over a large period [4].

These systems are used as the second opinion for the diagnosis made by the radiologists. The computer output is obtained from the computerized analysis of digital chest radiographs based on image processing based on image and data processing. Concept of Computer-aided diagnosis (CAD) made by radiologists is different than "automated diagnosis" made by computers. CAD systems include not only the generation of algorithms for detection and classification of lesions on chest photographs, but also the investigation of effects of the computer output on radiologists' diagnoses.

The purpose of CAD is to improve the quality and the productivity of the radiologists' image interpretation. This helps in reducing the time taken to read the image, improving the accuracy when interpreting data.

## **1.9.Lung Nodule Detection**

Computer-aided diagnosis (CAD) has become one of the major research subjects in medical imaging and diagnostic radiology. The number of presented papers on subject related to CAD at Radiological Society of North America (RSNA) in Chicago, which is the main meetings in the field of diagnostic radiology, has been increased by approximately 50% during past years. There are many CAD systems that have been developed for the detection and/ characterization of various lesions in medical imaging, including conventional projection radiography, computed tomography (CT), Magnetic resonance imaging(MRI) and ultrasound.

The organs currently being subjected to research on CAD includes breast, chest, colon, brain, liver, kidney and the vascular and skeleton systems. The basic idea of CAD is to provide a computer output as a “second opinion” to radiologists' image reading [1-7]. Therefore, for the development of a successful CAD scheme is necessary not only to develop computer algorithm, but also to investigate how useful the computer output would be for radiologists in their diagnoses, how to quantify the benefits of the computer output for radiologists, and how to maximize the effect of the computer output on their diagnoses.

The goal of CAD is to improve the quality and the productivity of radiologists' task by improving the accuracy and the consistency of radiological diagnoses and by reducing the image reading time. The general approach of CAD is to find the location of a lesion and to determine an estimate of the probability of the disease; these correspond to CAD for detection of a lesion and CAD for differential diagnosis.

These are the basic technologies involved in CAD schemes.

1. Image processing for detection and extraction of abnormalities
2. Quantization of image features for candidates of abnormalities

3. Data processing for classification image features between normal and abnormal (for benign and malignant)
4. Quantitative evaluation and retrieval of images like those of unknown lesions
5. Observer performance studies using ROC curve.

Because the concept of CAD is broad, CAD can be applied to all imaging modalities including projection radiography, CT, MRI, ultrasound and nuclear medicine imaging used for all body parts such as skull, thorax, heart, abdomen and extremities, and all kind of examinations including skeletal imaging, soft tissue imaging, functional imaging and angiography.

However the majority of CAD schemes developed in the past included the detection of breast lesions on mammograms [7, 21-30], the detection of lung nodules in chest radiographs [31-49] and thoracic CT[50-71], and the detection of polyps in CT colonography[72,82].

Therefore, the current results obtained from basic research and clinical applications of CAD may be considered the tip of an iceberg. Therefore, major impact of CAD on medical imaging and diagnosis radiology may be expected in the future.

One of the important events in the history of CAD is that ‘R2 Technology’ has succeeded in commercialization of the first CAD system for detection of breast lesions in mammography based on licensing of CAD technologies from University of Chicago, and that it obtained US Food and Drug Administration(FDA) approval for the clinical use of their system in 1998. Subsequently clinical uses of the mammographic CAD system have begun at many screening sites for breast cancers in the United States, and more than 1500 CAD systems are currently use in assisting radiologists in the early detection of breast cancer at many hospitals, clinics and screening centers around the world. It also had been reported that CAD has provided a gain of approximately 20% in the early detection of breast cancers on mammograms [29]. In 2001, ‘Dues technologies’ developed another CAD system to detect lung nodules on chest radiographs, and then received FDA approval for its clinical use. In Japan, Mitsubishi Space Software had developed a CAD system with temporal subtraction of sequential chest radiographs and also for detection of pulmonary nodules in thoracic CT have been developed by manufacturers and have being

evaluated at medical centers around the world. Recently, 'R2 Technology' received FDA approval for their CAD system for detection of pulmonary nodules in CT.

Because several commercial CAD systems for detection of breast cancer on mammograms are currently available for clinical use, some recent research on mammographic CAD has been focused on clinical studies evaluating the effect of CAD on radiologists' performance and on clinical outcomes [83-88]; these include prospective studies on the detection rates of breast cancer without and with the use of CAD [29,89] and comparison of double reading of mammograms with single reading by CAD [90]. At present, the results of these studies are mixed and controversial, thus requiring further investigations on the effect of the computer output on clinical outcome based on the prospective studies and/or randomized clinical trials.

The development of multi detector CT (MDCT) has produced many CT images that may require additional time and effort in image interpretation by radiologists. It has expected, that CAD would assist radiologists in reducing the reading time as well in improving the diagnostic accuracy. Many investigators have attempted to develop CAD schemes for detecting pulmonary nodules by MDCT. Because the quality of MDCT images has been improved considerably in terms of three dimensional (3D) images, the performance of CAD schemes in the detection of nodules MDCT images has generally been improved.

Subjects for CAD schemes included in the following sections:

- 1) Detection and classification of the lung nodules on digital chest radiographs
- 2) Detection of lung nodules on digital chest radiographs
- 3) Detection of nodules in low dose CT
- 4) Detection between benign and malignant nodules on high resolution CT
- 5) Usefulness of similar images for distinction between benign and malignant lesions
- 6) Detection of intracranial aneurysms in magnetic resonance angiography

## **1.10. Detection and classification of lung nodules on digital chest radiographs**

Normally radiologists may miss nearly 20% to 50% of detectable lung nodules on chest radiographs. These errors can be resulted in delaying starting the cancer treatments, which may finally affect negatively in patient's survival rate. [5] Therefore the outcome of CAD is to assist radiologists to identify or draw their attention on potential locations or affected areas by nodules in chest radiographs. Although radiologists make their own judgments, these CAD systems help them by depicting the missed potential locations of nodules which are hard to identify.

The computerized scheme for automated detection of nodules was based on a different image technique. [31,32] with which nodules were enhanced and the majority of background normal structures were suppressed. The candidates for nodules were then identified by thresholding of pixel values in the difference images derived from a chest radiograph. A number of image features on nodule candidates were quantified, and some false positives caused by a rule-based method together with the use of artificial neural networks. Finally, the locations of the potential sites for nodules were indicated by markers such as arrows on chest images displayed on a monitor. But that one false positive in the computer output which is pointing to a normal anatomical structure in the mediastinum was included. It is important to reduce the number of such false positives as much as possible in CAD schemes, including that for detection of lung nodules.

The biggest problem in lung cancer detection is that lesions are relatively small and contrast poorly with respect to anatomical structure. In last decade, several CAD systems for chest radiology have been proposed and tested on private image sets. Different approaches based on image processing techniques, conventional pattern recognition techniques and artificial neural networks etc. have been proposed and developed to detect lung nodules successfully.

Lung nodules commonly describes as small round-like objects which have a low contrast. But these nodules generally have a high variable shape and their boundaries are not defined. In addition, normal anatomical structures are responsible for making troubles. Also, nodules are embodied in a complex background where they can be hidden easily. Also, the projection process produces nodule-like shapes on the radiogram because various anatomical structures generate gray

level patterns like nodules do in conventional X-ray imaging. This happens mostly for vessels and overlapping bone structures such as rib cross. Normally low contrasts of lesions are small in their sizes. In such cases anatomical masking affects heavily in detection sensitivity.

It is a localized object which looks like a blob like pattern and it is slightly lighter than the surrounding structures. A simple strategy to achieve high sensitivity in nodule detection is to search the captured image for gray level blobs.

The next section discusses on how different technologies have been involved in detecting cancer nodules and how past research papers had discussed about cancer cell detection.

## 2. Literature Review

### 2.1. Nodule location detection

A paper published in 2003 by G. Coppini, M. Falchini, N. Villari and G. Valli have used Neural Networks for Computer Aided Diagnosis. The main aim of this paper is to detect Lung Nodules in Chest Radiograms [6]. They mention nodules are as localized objects that produce a blob like patterns which are slightly lighter than the surrounding structures. The strategy they had used is, searching the image exhaustively for gray level blobs with the range of 5-20 mm. Blob searching by using Laplacian of Gaussian(LoG) filters in a multiscale setting, which had been first been proposed by Blostein and Ahuja [7]. Further investigations on this area had been reported by Lindeberg [8]. The benefit of the second approach is no prior knowledge is required about the actual shape of the nodules and the structure of the image background, is what needed.

Laplacian filters are derivative filters used to find areas of rapid change (edges) in images. Since derivative filters are very sensitive to noise, it is common to smooth the image(e.g using a Gaussian filter) before using the Laplacian. This two-step process is what called Laplacian of Gaussian(LoG) operation.

The LoG kernel can be written as,

$$\begin{aligned} \text{LoG}_\sigma(x, y) &= w(\sigma) \nabla^2 G_\sigma(x, y) \\ &= -\frac{w(\sigma)}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] \exp\left[ -\frac{x^2 + y^2}{2\sigma^2} \right] \end{aligned}$$

$\sigma$  - Standard deviation of the Gaussian

$w(\sigma)$ - Normalization term depending on  $\sigma$ .

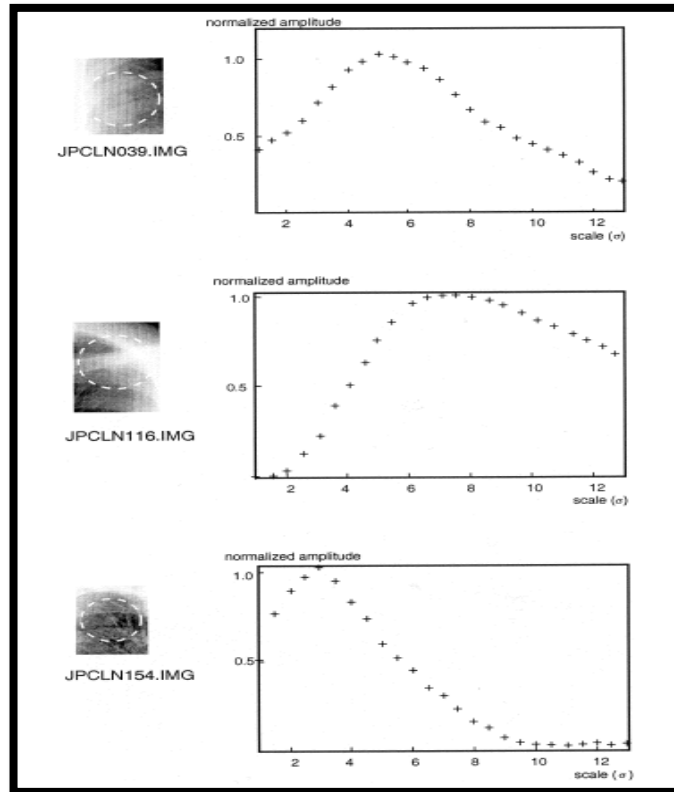


$-w(\sigma) = (e\sigma^2/2)$  has been selected during the paper because, it ensures that the LoG central lobe is positive and has a unit volume. Paper mentions that the LoG-filtered image is not biased by the scale parameter and we can compare images which have been filtered with different  $\sigma$ s. In addition, it is convenient to relate the filter scale to the width  $h$  of its central lobe, where  $h = 2\sqrt{2}\sigma$ . These LoG filters have many important mathematical properties and they can be implemented very efficiently.

According to the theory, of linear scale-space [8], the representation  $L(x, y, \sigma)$  of the image  $I(x, y)$  has been built as,

$$L(x, y, \sigma) = I(x, y) \otimes \text{LoG}_\sigma(x, y)$$

Where  $\otimes$  is the convolution operator. The analysis of the plot (scale-space signature)  $L(x, y, \sigma)$  with respect to the value  $\sigma^*$ , of  $\sigma$  usually suggests the presence of a blob at a scale  $\sigma$ [8]. In Fig. 2,  $L(x, y, \sigma)$  vs.  $\sigma$  for some nodules in JSRT database has been plotted. Images show that though the shape of the signature varies, a well-defined peak is always an evident.



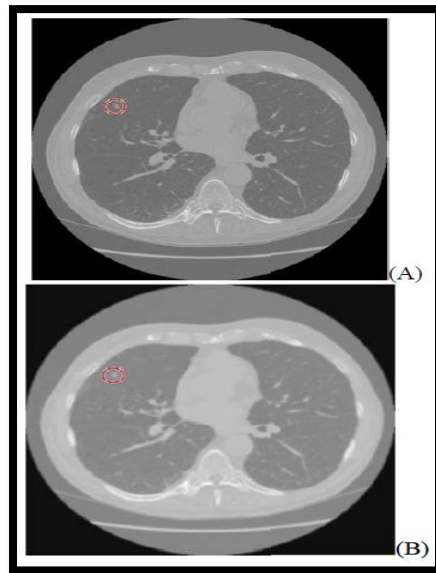
**Figure 1-Signatures of real nodules from the JSRT database.**

**Amplitudes are normalized with respect to the peak-value**

As mentioned in the paper blob searching alone will result in reducing the specificity of blob detection as normal chest anatomy is characterized by blob-like structures. They have used prior knowledge to handle this problem as it leads to disregard most of the detected maxima by eventually providing a small set of region of interest (ROI). Supervised neural network methods have been used in this regard as they discover meaningful relations between image gray level patterns and nodule lesions. More descriptive details on this regard will be discussed later in this chapter.

M. Assefa, I. Faye, A.S. Malik, had presented, a lung nodule detection method using Multi-Resolution Analysis approach using low-dose CT scans [9]. This scheme consists of two stages. In the first stage nodular models are developed and used as a reference and then Multi resolution and intensity based features have been extracted for false positive reduction and improvement of

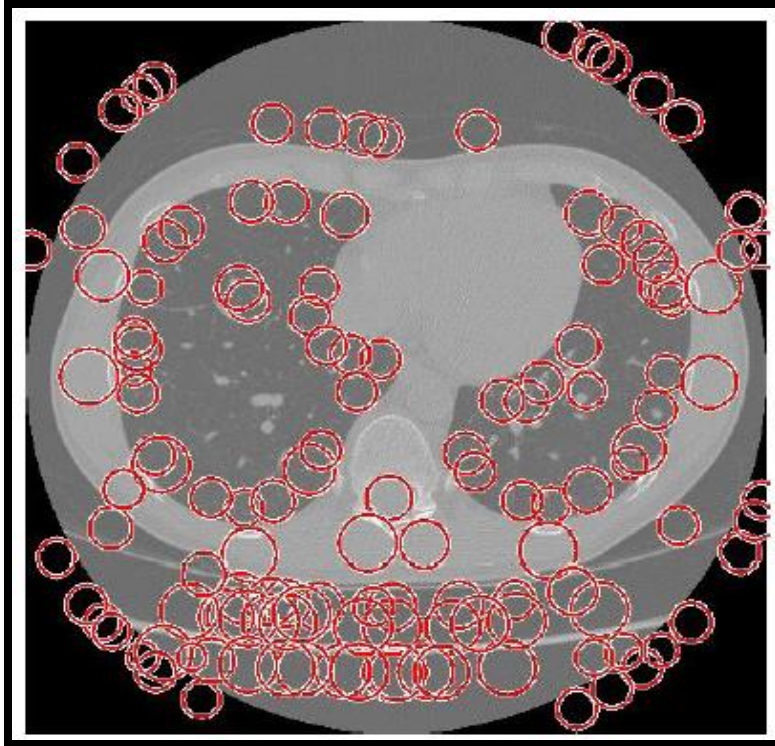
the detection rate. As the first step they have performed a noise removal in the entire dataset. They have developed a Wiener filter with appropriate filter parameters applied them on ground truth dataset with pre-identified nodules (Fig. 2A). An adaptive Wiener filtering on 7\*8 or 9\*9 blocks have given the best performance as shown in the (Fig 2B).



**Figure 2-Nodule visibility test before(A) and after(B) Applying Wiener Filter**

Since nodules appear on consecutive slices depending on their sizes to detect pulmonary nodules in 2-D slices circular and semi-circular templates have been developed. The circular templates have been used to detect nodules inside the lung region and the semi-circular templates have been designed to detect nodules at the boundary.

As they mentioned, since pulmonary nodules appear as circular structure on LDCT 2-D scans, a circular structure detection scheme is employed as an initial pre-processing step (Figure 3). This stage helps to optimize the computational time for the template matching process for nodule residing inside the lungs.



**Figure 3- Result of circular object detection**

In the paper they have looked at the intensity HU (Hounsfield units) values in and around the nodules and have observed that the nodules intensity values followed a Gaussian-Like distribution and as a result, templates with Gaussian like intensity distributions have been developed by them. It is important to know how they have removed false positive nodules per scan. They have identified two intensity based and seven multi-resolution based feature for the reduction of false positive nodules. They have used Mean and Standard Deviation to differentiate between the parts of the bone and skin from the true positives.

In the paper published in 2013 (Computerized Detection of Lung Nodules by Means of “Virtual Dual-Energy” Radiography), by S.Chen and K. Suzuki, presents a CAD system with improved sensitivity and specificity by using “virtual dual energy (VDE)” chest radiographs where ribs and clavicles are suppressed with massive-training artificial neural networks (MTANNs). A dual energy subtraction technique is a technique for separating soft tissue from bones in chest radiographs (CXR) by use of two X-ray exposures at two different energy levels. But this

technology is used by lower number of hospitals because a specialized equipment is required and radiation dose can be doubled. To address this issue of the availability of the dual-energy systems Suzuki et al. had developed an image processing technique called virtual dual energy (VDE) radiography to suppress ribs and clavicles in CXRs by means of a multiresolution MTANN. Once the training is done, real dual energy images were not necessary any more.

### **Original Computerized scheme for nodule detection**

There were four major steps involved with the detection of lung nodules in CXRs.

- 1) Segmentation of lung fields based on multi-segment active shape model (M-ASM)
- 2) Two stage nodule enhancement and nodule candidate detection
- 3) Segmentation of nodule candidates by use of clustering watershed algorithm
- 4) Feature analysis and classification of the nodule candidates into nodules or non-nodules by use of nonlinear support vector machine (SVM) classifier.

M-ASM – This is a lung segmentation method which is an improved model of Cootes et.al formulation by fixating selected nodules at specific structural boundaries called transitional landmarks. These landmarks determine multiple segments each of which related to specific boundary types such as heart, aorta, and rib cage.

The node specified active shaped model (ASM) was built using a fixed set of equally spaced feature nodes for each boundary segment. It allows nodes to learn local appearance models for a specific boundary type rather than generalizing over multiple boundary types which resulted in improving the marked boundary accuracy.

## **2.2.Lung Segmentation and Boundary Correction Method**

Segmenting lung region is done normally as a preprocessing step. Normally this task gets complicated when the disease is presented.

‘Computerized detection of Pulmonary nodules on CT scans’ by S.G.Armato and et. al mention an approach to detect pulmonary nodules automatically in CT images. This scheme includes both two and three-dimensional analyses. Within each section, gray- level thresholding methods have been

used to segment the thorax from the background and then the lungs from thorax. An algorithm called rolling ball had been applied to lung segmentation to avoid the loss of juxtapleural nodules. Then multiple gray level thresholds have been applied to lung region to identify nodule candidates. These candidates represent both nodules and normal pulmonary structures. For each candidate, two and three dimensional geometric and gray level features have been computed. These features have been merged with linear discriminant analysis to reduce the number of candidates that correspond to normal structures.

As mentioned in [9] ‘A novel lung segmentation and boundary correction method’ paper by P.H.Jianping and D. Jianping, had proposed a novel method to segment the lung region in CT images. They have proposed a novel lung segmentation algorithm by combining region growing and thresholding to do rough segmentation. Then they have used a boundary correction algorithm based on convex hull operation and ray casting to do boundary correction by minimizing under segmentation and over segmentation relative to the true lung borders. This approach corrects the border in a geometric way. They have extracted lungs from chest CT images by the combination of 2D region growing and gray level thresholding in rough lung segmentation. In the results, the juxta pleural nodules and pulmonary vessels contact to the lung boundary are often lost. Then they have divided the outline of left/right lung into two parts based on geometric characteristic of the outline. To bridge the local concavities (having an outline or surface that curves inwards like the interior of a circle or sphere) formed by improperly excluded juxta pleural nodules, they have reconstructed the shape in the tumor-free state by computing the convex hull of the outer boundary and cut off the nodule along the boundary of the convex hull. To correct the inner boundary near mediastinum and pulmonary vessels, they applied a ray casting approach.

Although density based approaches such as gray-level thresholding can be used to extract the lung parenchyma (Lung parenchyma is that portion of the lung involved in gas transfer) efficiently, paper mentions that such approaches incorrectly exclude some important regions such as juxtapleural nodules from the segmentation, because they are contiguous with chest wall and have similar density. They also mention to segment lung without loss of lung nodules and to ensure smooth appearance of lung boundaries, it is required to correct the lung boundary after lung segmentation.

### 2.3.Lung cancer detection using artificial neural networks

‘Lung cancer detection by using artificial neural network and fuzzy clustering method’ by G. Dheepak, S Premkumar and et.al [13] presents about two segmentation methods, Hopfield Neural Network (HNN) and Fuzzy C-Mean (FCM) clustering algorithm for segmenting sputum color images to detect the lung cancer in its early stages and to determine which is the best. The extreme variation in gray level and the relative contrast among the images make the segmentation results less accurate. Therefore, they had used thresholding techniques as pre-processing steps in all images to extract the nuclei and cytoplasm regions.

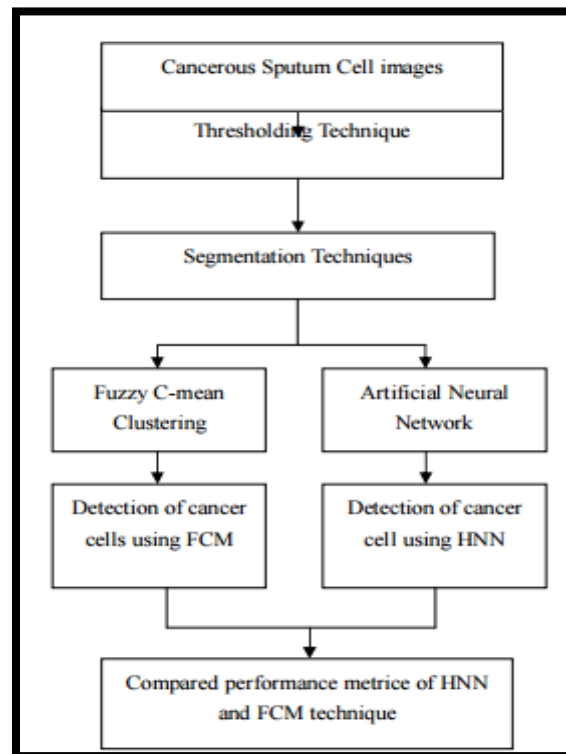


Figure 4-Flowchart for the process of lung cancer detection using FCM and HNN

‘Lung Cancer Detection Using Artificial Neural Network & Fuzzy Clustering’ by Masaood A. Hussain<sup>1</sup> , Tabassum M. Ansari , Prarthana S. Gawas , Nabanita Nath Chowdhury [14] had presented how artificial neural networks can be used for lung cancer detection and following displays the implementation steps they had used in brief.

Their methodology follows up with following main steps.

- 1) Data collection
- 2) Noise removal (Median filter which contains highest signal to noise ratio had been used to remove noise)
- 3) Image enhancement (Increase the intensity of the image and thus can be used for feature extraction)
- 4) Gray scale to Binary conversion (Image binarization is a subclass of image segmentation as it divides an image into segments based on the values of pixels to a threshold value)
- 5) Removal of erroneous part of the image (parts of the image which do not need for further processing is removed in this section)
- 6) Image segmentation. (Image segmentation is obtained by forming a gray level co-occurrence matrix of the image. This matrix is also used for feature extraction. )
- 7) Calculation and feature extraction (Once the segmentation is done on the lung region, the features can be obtained. Ex: Mean, Correlation, Standard deviation, skewness, homogeneity)

This paper had only discussed about how the preprocessing steps need to be done. But they have mentioned those steps clearly so this information can be used for further research studies of this area.

Raviprakash S. Shriwas and Akshay D. Dikondawar in ‘Lung cancer detection and prediction by using neural network’[15] presents that the feature extraction uses two methods mainly called, Binarization and GLCM(Gray Level Co-Occurance Method) is as both methods are based on strong facts that related to lung anatomy and information of lung CT imaging.

GLCM (Gray Level Co-Occurance Method) is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image. They had first created a gray-level co-occurrence matrix from image by using graycomatrix function in MATLAB software. From this they had calculated, texture measures from the GLCM. The features extracted using this method are (contrast, energy, entropy, maximum probability, correlation, cluster shade, cluster prominence, dissimilarity, autocorrelation, sum variance, sum entropy, difference entropy, information measures).



Binarization approach had been used to extract the number of white pixels and check them against some threshold to check the normal and abnormal lungs. They had counted the black pixels for normal and abnormal images to get the average which will be denoted later as threshold. Then each image's black pixels had been compared to this threshold, to check whether it is greater or not. If the number of the white pixels of a new image is less than the threshold, then it indicates that the image is normal. If the number of white pixels is greater than the threshold value, then it shows that the image is abnormal. Combining Binarization and GLCM approaches together had lead them to take a decision whether the case is normal or abnormal.

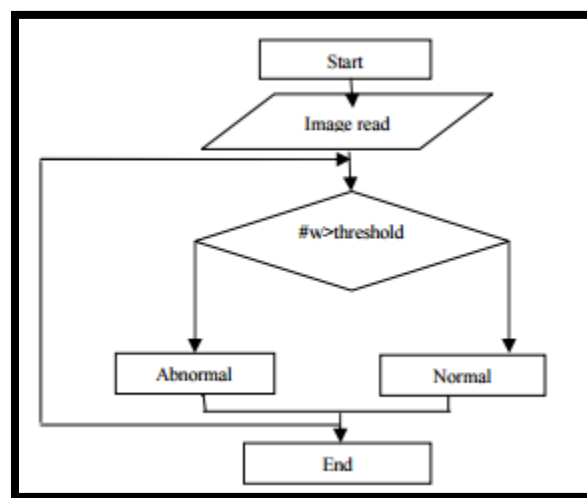


Figure 5-Binarization check method

'Detection and Classification of Lung Cancer Using Artificial Neural Network' by A. Pathan, and B.K.Saptalkar [16], mentions three main pre-processing steps which were followed before classification.

#### 1) Seedfill operation

This is an algorithm that determines the area connected to a given node in a multidimensional array. It performs the operation on background pixels of the binary image starting from the points specified in locations. It fills the holes in the binary image.

## 2) Region of interest (ROI)

A region of interest is a selected subset of samples within a dataset identified for a purpose. The concept of an ROI is commonly used in many application areas. In medical imaging, the boundaries of a tumor may be defined on an image or in a volume, for measuring its size.

## 3) Image segmentation

As mentioned above, segmentation is the process of partitioning an image into disjoint and homogeneous areas. This task can be equivalently achieved by finding the boundaries between the regions. Regions of image segmentation should be uniform and homogeneous with respect to some characteristics such as gray tone or texture. Adjacent regions should have significantly different values with respect to the characteristics which they are uniform. Boundaries of each segment should be simple, not ragged and must be spatially accurate. Therefore, segmentation is an extremely important section in several applications of image processing and computer vision. This segmentation should be happened in a meaningful way. This paper mentions that image segmentation is mostly done based on the color and texture.

## **Classification**

As mentioned in this paper classification is the process of classifying the cancerous images by extracting the features of the given image suffering from the cancer and these features are compared with the features of the given sample image. The above paper 35 samples of images are given for classification and the features of these images are compared with the given image and hence lung cancer is detected.



Figure 8-Input image



Figure 7-Seed fill operation



Figure 6-Region of interest

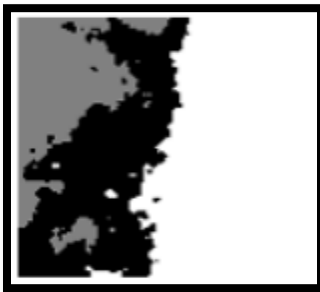


Figure 9-Segmented image

The classifications of cancerous images are shown in below figure.

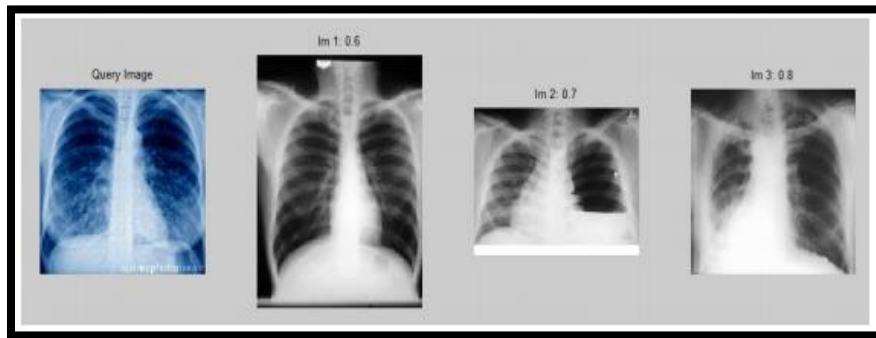
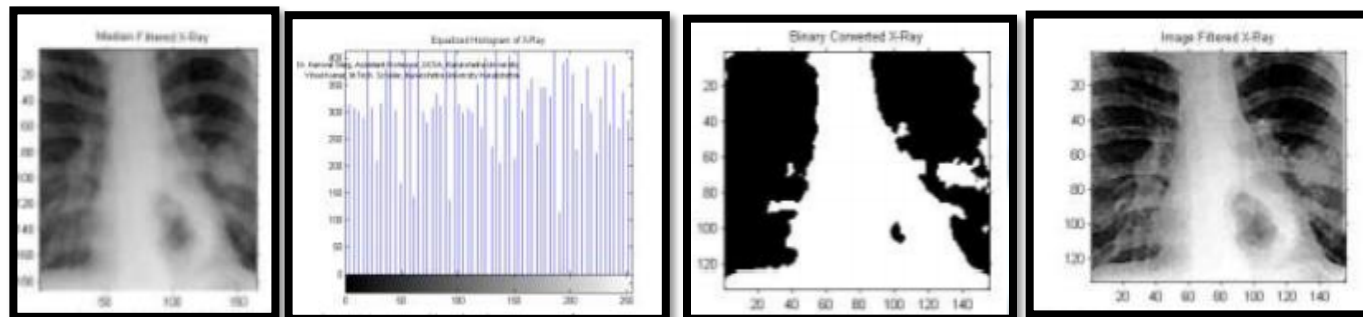


Figure 10-Classification of cancerous images

‘Detection system for lung cancer based on neural network: X-Ray validation performance’ by V. Kumar, A. Saini [17], mentioned some different image processing approaches other than we discussed above. To smooth and enhance the contrast in the captured X-Ray image, they have applied median and Gaussian filters as showing in following figures. The images of lungs have been segmented by using the Otsu’s threshold. Once after the binary conversion is done they have used morphologic technique to extract the features including the parameter, area and shape of the

nodule. Once these denoise processes are followed, to enhance the edge of image structures and to enhance the image contrast histogram equalization approach has been used. As they mentioned, one of the most common problems when detecting lung cancers are white noises. The main suggestion for this that have been suggested is to calculate the pixel weights. As mentioned, the median filter is a nonlinear common enhancement digital filtering technique for removing noise without reducing the sharpness of the image.



The next section explains how deep neural networks have been used in image classification.

## 2.4. Evaluation of existing deep neural network architectures

Artificial neural networks consist of three main layers; input layer, hidden layer, and output layer using the back-propagation algorithm. A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers of units between the input and output layers. Like shallow ANNs, DNNs can model complex non-linear relationships.

The basic idea of this research is to evaluate the performance of lung cancer detection using a deep neural network architecture. Deep neural networks and deep learning are very powerful and popular algorithms. Their success lays in the careful design of the neural network architecture.

Figure 11 shows, Reporting Top-1 accuracy vs. amount of operations required for a single forward pass in multiple popular neural network architectures.

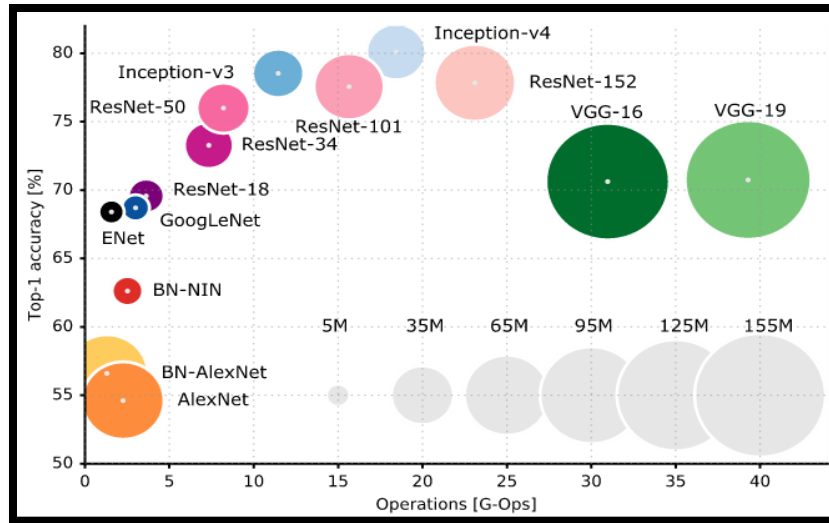


Figure 11-Reporting Top-1 accuracy vs. amount of operations required for a single forward pass in multiple popular neural network architectures

### 2.4.1. The advantages of deep neural networks over conventional old frameworks

- 1) Deep learning can directly uncover features from the training data. Therefore, the effort of explicit elaboration on feature extraction can be significantly reduced.
- 2) Feature interaction and hierarchy can be exploited jointly within the basic deep architecture of a neural network. (This leads to simplify the feature selection process significantly)
- 3) The three steps of feature extraction, selection and supervised classification can be realized within the optimization of the same deep architecture. (With such a design, the performance can be tuned more easily in a systematic fashion)

### 2.4.2. LeNet 5

LeNet 5 is the very first convolutional neural network. And this is the base of the concept of Deep learning. This pioneering work had been performed by Y. LeCun, L. Bottou, Y. Bengio, P. Haffner in paper, 'Gradient-Based Learning Applied to Document Recognition'. [19]

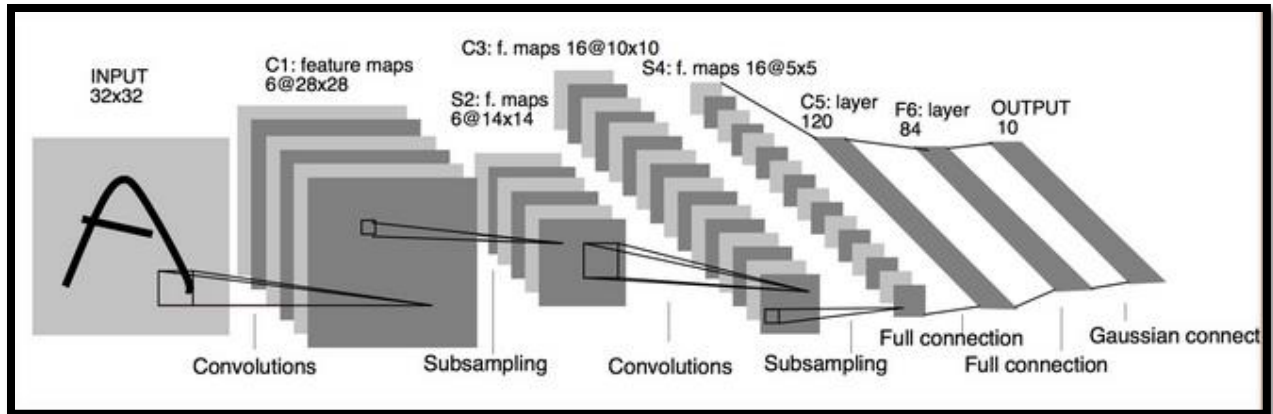


Figure 12-LeNet5 architecture

The LeNet5 architecture was the fundamental architecture which explained that an image features are distributed across the entire image and convolutions with learnable parameters which is an effective way to extract similar features at multiple location with few parameters. This was identified at the time that there were no GPUs to help training, and even CPUs were slow. The ability to save parameters and perform advanced computation was a key advantage of this approach. This contrasts with using each pixel as a separate input of a large multi-layer neural network. LeNet5 explained that those should not be used in the first layer, because images are highly spatially correlated, and using individual pixel of the image as separate input features would not take advantage of these correlations.

LeNet5 features can be summarized as:

convolutional neural network use sequence of 3 layers: convolution, pooling, non-linearity → This may be the key feature of Deep Learning for images since this paper;

- use convolution to extract spatial features
- subsample using spatial average of maps
- non-linearity in the form of tanh or sigmoids
- multi-layer neural network (MLP) as final classifier
- sparse connection matrix between layers to avoid large computational cost
- In overall this network was the origin of much of the recent architectures, and a true inspiration for many people in the field.

In the years from 1998 to 2010 neural networks were in progress. Most people did not notice their increasing power, but many other researchers slowly progressed. Meanwhile, data were exponentially increasing because of the rise of cell-phone cameras, cheap digital cameras and computing power was rising as CPUs were becoming faster and GPUs were becoming a general-purpose computing tool. These two trends made neural network progress, but at a slow rate. Both increasing data amounts and computing power made the tasks that neural networks tackled more interesting.

### 2.4.3. Dan Ciresan Net

In 2010, [20] ‘Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition’ by D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhber, had published one of the very first implementation of GPU Neural nets. This implementation had both forward and backward implemented on a a NVIDIA GTX 280 graphic processor of an up to 9 layers neural network.

### 2.4.4. AlexNet

In 2012, A. Krizhevsky, I. Sutskever, G.E. Hinton released ‘ImageNet Classification with Deep Convolutional Neural Networks’ which was a deeper and much wider version of the LeNet and won by a large margin the ImageNet competition.

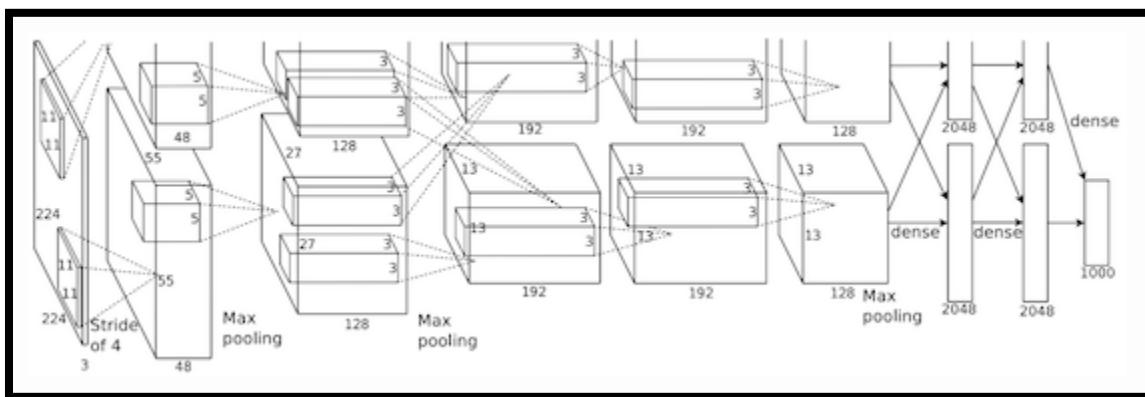


Figure 13-AlexNet architecture

AlexNet scaled the insights of LeNet into a much larger neural network that could be used to learn much more complex objects and object hierarchies.

The contributions of this work were:

- use of rectified linear units (ReLU) as non-linearities
- use of dropout technique to selectively ignore single neurons during training, a way to avoid overfitting of the model
- overlapping max pooling, avoiding the averaging effects of average pooling
- use of GPUs NVIDIA GTX 580 to reduce training time

At this time GPU offered a much larger number of cores than CPUs, and allowed 10x faster training time, which in turn allowed using larger datasets and bigger images.

The success of AlexNet started a small revolution. Convolutional neural network were the workhorse of Deep Learning, which became the new name for “large neural networks that can now solve useful tasks”.

#### **2.4.5. Overfeat**

In December 2013 the NYU lab from Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun came up with Overfeat [22] ‘Integrated Recognition, Localization and Detection using Convolutional Networks’, which is a derivative of AlexNet. I believe it is better to learn to segment objects rather than learn artificial bounding boxes.

#### **2.4.6. VGG**

The VGG networks from Oxford’s, Karen Simonyan and Andrew Zisserman,[23] ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’ were the first to use much smaller 3×3 filters in each convolutional layers and also combined them as a sequence of convolutions.

This seems to be contrary to the principles of LeNet, where large convolutions were used to capture similar features in an image. Instead of the 9×9 or 11×11 filters of AlexNet, filters started to become smaller, too dangerously close to the infamous 1×1 convolutions that LeNet wanted to



avoid, at least on the first layers of the network. But the great advantage of VGG was the insight that multiple  $3 \times 3$  convolutions in sequence can emulate the effect of larger receptive fields, for examples  $5 \times 5$  and  $7 \times 7$ . These ideas will be also used in more recent network architectures as Inception and ResNet.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Figure 14- VGG Network

The VGG network uses multiple  $3 \times 3$  convolutional layers to represent complex features. Notice blocks 3, 4, and 5 of VGG-E:  $256 \times 256$  and  $512 \times 512$   $3 \times 3$  filters are used multiple times in sequence to extract more complex features and the combination of such features. This is effectively like having large  $512 \times 512$  classifiers with 3 layers, which are convolutional! This obviously amounts to a massive number of parameters and learning power. But training all these networks was difficult and had to split into smaller networks with layers added one by one. All these were

because of the lack of strong ways to regularize the model, or to somehow restrict the massive search space promoted by the large number of parameters.

VGG used large feature sizes in many layers and thus inference was quite costly at run-time. Reducing the number of features, as done in Inception bottlenecks, will save some of the computational cost.

#### 2.4.7. Network-in-network

In December 2013, Min Lin, Qiang Chen and Shuicheng Yan's [25] 'Network-in-network' (NiN) had the great and simple insight of using  $1 \times 1$  convolutions to provide more combinational power to the features of a convolutional layers.

The NiN architecture used spatial MLP layers after each convolution to combine features before another layer. Again, one can think the  $1 \times 1$  convolutions are against the original principles of LeNet, but in reality they instead help to combine convolutional features in a better way, which is not possible by simply stacking more convolutional layers. This is different from using raw pixels as input to the next layer. Here  $1 \times 1$  convolution is used to spatially combine features across features maps after convolution, so they effectively use very few parameters, shared across all pixels of these features!

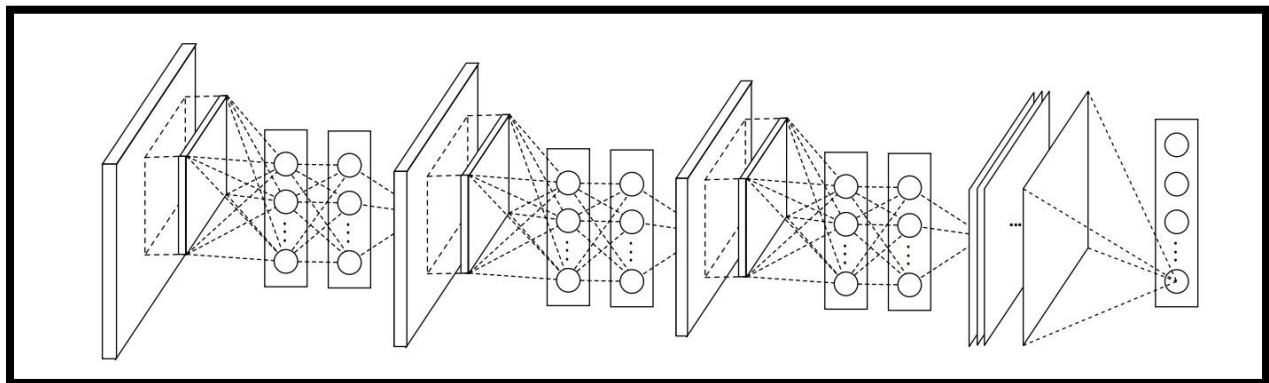


Figure 15-Network in Network architecture

The power of MLP can greatly increase the effectiveness of individual convolutional features by combining them into more complex groups. This idea will be later used in most recent architectures as ResNet and Inception and derivatives.

### 2.4.8. GoogLeNet and Inception

In September 2014, Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich from Google began a quest aimed at reducing the computational burden of deep neural networks, and devised the [26] ‘Going Deeper with Convolutions’ GoogLeNet the first Inception architecture.

By now, Fall 2014, deep learning models were becoming extremely useful in categorizing the content of images and video frames. Most skeptics had given in that Deep Learning and neural nets came back to stay this time. Given the usefulness of these techniques, the internet giants like Google was very interested in efficient and large deployments of architectures on their server farms.

Christian thought a lot about ways to reduce the computational burden of deep neural nets while obtaining state-of-art performance (on ImageNet, for example). Or to keep the computational cost the same, while offering improved performance.

He and his team came up with the Inception module:

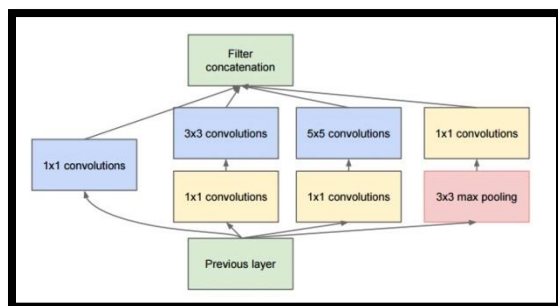


Figure 16-Inception module

Which at a first glance is basically the parallel combination of  $1 \times 1$ ,  $3 \times 3$ , and  $5 \times 5$  convolutional filters. But the great insight of the inception module was the use of  $1 \times 1$  convolutional blocks (NiN)

to reduce the number of features before the expensive parallel blocks. This is commonly referred as “bottleneck”. This deserves its own section to explain: see “bottleneck layer” section below.

GoogLeNet used a stem without inception modules as initial layers, and an average pooling plus softmax classifier like NiN. This classifier also have extremely low number of operations, compared to the ones of AlexNet and VGG. This also contributed to a very efficient network design by Alfredo Canziani, Adam Paszke and Eugenio Culurciello [27] ‘An Analysis of Deep Neural Network Models for Practical Applications’ in May 2016.

#### **2.4.9. Bottleneck layer**

Inspired by NiN, the bottleneck layer of Inception was reducing the number of features, and thus operations, at each layer, so the inference time could be kept low. Before passing data to the expensive convolution modules, the number of features were reduced by, say, 4 times. This led to large savings in computational cost, and the success of this architecture.

Let’s examine this in detail. Let’s say you have 256 features coming in, and 256 coming out, and let’s say the Inception layer only performs 3x3 convolutions. That is 256x256 x 3x3 convolutions that to be performed (589,000s multiply-accumulate, or MAC operations). That may be more than the computational budget to run this layer in 0.5 milli-seconds on a Google Server. Instead of doing this, they have decided to reduce the number of features that will have to be convolved, say to 64 or 256/4. In this case, they have first performed 256 -> 64 1x1 convolutions, then 64 convolutions on all Inception branches, and then we use again a 1x1 convolution from 64 -> 256 features back again. The operations are now:

- **$256 \times 64 \times 1 \times 1 = 16,000s$**
- **$64 \times 64 \times 3 \times 3 = 36,000s$**
- **$64 \times 256 \times 1 \times 1 = 16,000s$**

For a total of about 70,000 versus the almost 600,000 which had before. Almost 10x fewer operations.

And although there are fewer operations, generality is not losing in this layer. In fact, the bottleneck layers have been proven to perform state-of-art on the ImageNet dataset and will be also used in later architectures such as ResNet.

The reason for the success is that the input features are correlated, and thus redundancy can be removed by combining them appropriately with the 1x1 convolutions. Then, after convolution with a smaller number of features, they can be expanded again into meaningful combination for the next layer.

#### **2.4.10. Inception V3 (and V2)**

In February 2015, Sergey Ioffe and Christian Szegedy's [28] 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift' Inception was introduced as Inception V2. Batch-normalization computes the mean and standard-deviation of all feature maps at the output of a layer, and normalizes their responses with these values. This corresponds to "whitening" the data, and thus making all the neural maps have responses in the same range, and with zero mean. This helps training as the next layer does not have to learn offsets in the input data, and can focus on how to best combine features.

In December 2015, Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna released a new version of the Inception modules and the corresponding architecture named [29] 'Rethinking the Inception Architecture for Computer Vision' This article better explains the original GoogLeNet architecture, giving a lot more detail on the design choices. A list of the original ideas is,

- Maximize information flow into the network, by carefully constructing networks that balance depth and width. Before each pooling, increase the feature maps.
- when depth is increased, the number of features, or width of the layer is also increased systematically
- Use width increase at each layer to increase the combination of features before next layer

Use only 3x3 convolutions, when possible, given that filter of 5x5 and 7x7 can be decomposed with multiple 3x3. See figure:

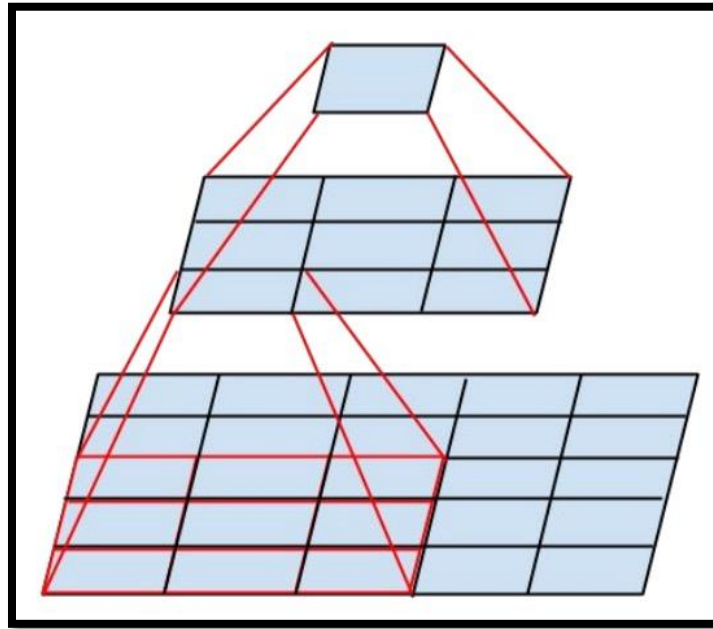


Figure 17-3x3 convolution layer

- The new inception module thus becomes:

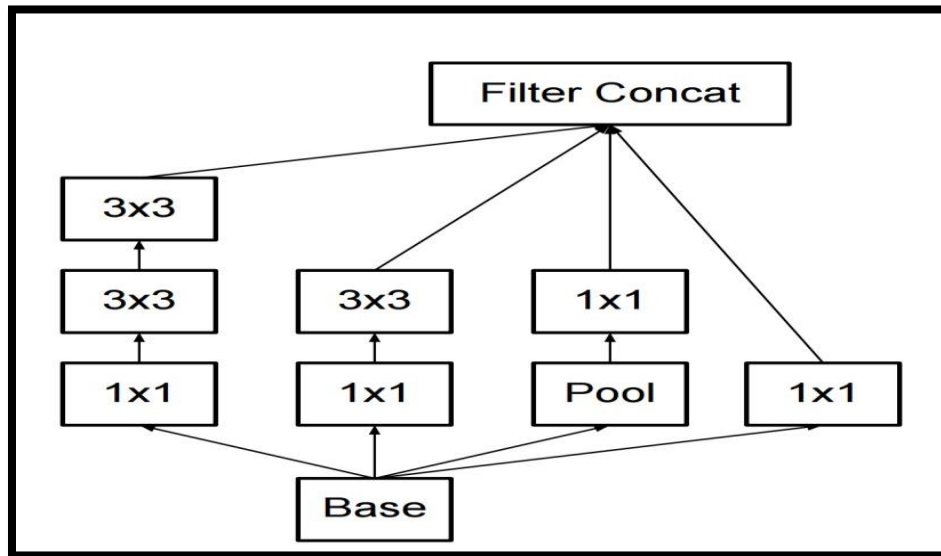


Figure 18-New inception model

- filters can also be decomposed by [30] ‘Flattened Convolutional Neural Networks for Feedforward Acceleration’ introduced by Jonghoon Jin, Aysegul Dundar, Eugenio Culurciello, into more complex modules:

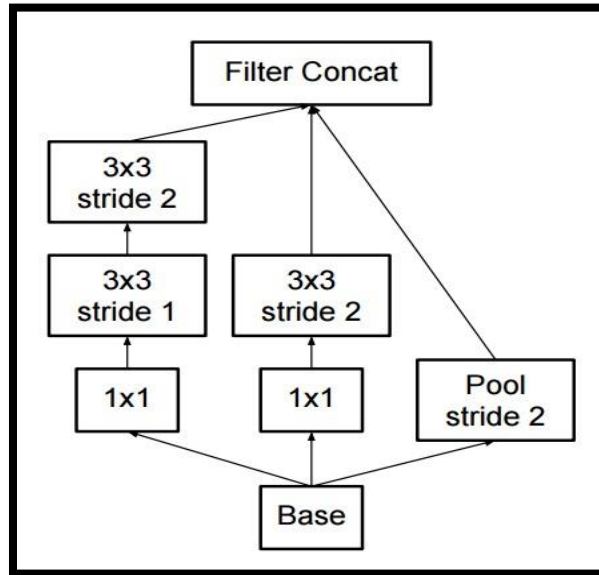


Figure 19-New inception model

- Inceptions modules can also decrease the size of the data by provide pooling while performing the inception computation. This is basically identical to performing a convolution with strides in parallel

#### 2.4.11. Multi-Column deep neural network

‘Multi-Column Deep Neural Network for Traffic Sign Classification’ by Dan Cire, san, Ueli Meier, Jonathan Masci and Jürgen Schmidhuber [18] had used a supervised learning way to classify Traffic Signs. They have combined various DNN trained on differently preprocessed data into a Multi-Column DNN (MCDNN) which further improves performance. As they mention the most hierarchical object recognition systems all extract localized features from input images convolving image patches with filters. Filter responses then repeatedly, resulting in a deep feed-forward network architecture whose output feature vectors are eventually classified. Computational speed is a limiting factor for deep architectures. For the experiments because of this reason they have relied on a fast implementation on Graphics Processing Units (GPUs).

Their proposed method, rely only on the raw pixel images.

Their DNN architecture consists of succession of convolutional and max pooling layers and each layer only receives connections from its previous layer. This is a hierarchical feature extractor that maps raw pixel intensities of the input image into a feature vector to be classified by several usually 2,3 fully connected layer.

### Convolutional layer

Each convolutional layer performs a 2D convolution of its  $M^{(n-1)}$  input maps with a filter of size  $K_x^n \times K_y^n$ . The resulting activations of the  $M^n$  output maps are given by the sum of the  $M^{(n-1)}$  convolutional responses which are passed through a nonlinear activation function:

$$Y_j^n = f\left(\sum_{i=1}^{M^{n-1}} Y_i^{n-1} * W_{ij}^n + b_j^n\right)$$

n- Indicates the layer

Y- map size  $M_x \times M_y$

$W_{ij}$  - filter size  $K_x \times K_y$  connecting input map i with output map j

$b_j^n$  - bias of output map j

$b_j^n$  \*- Valid 2D convolution



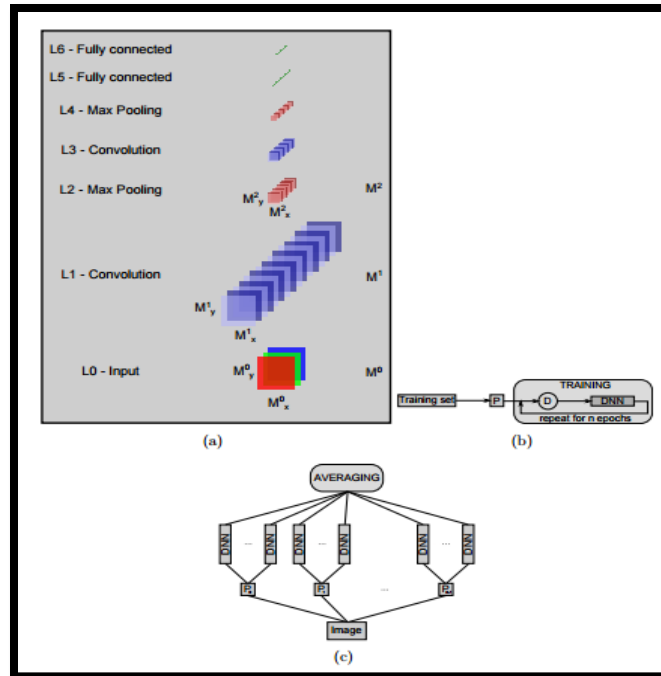


Figure 20-(a)-DNN architecture, (b)-Training a DNN, (c)-MCDNN architecture

- DNN architecture
- Training a DNN: The dataset is preprocessed (P) before training starts. During training all original or preprocessed images are randomly distorted before each epoch (D)
- MCDNN architecture. The input image is preprocessed by  $n$  different preprocessors. An arbitrary number of columns is trained on each preprocessed input. The final predictions are obtained by averaging individual predictions of each DNN.

## Max-pooling layer

The biggest architectural difference between our DNN and the CNN of LeCun et al. (1998) is the use of max-pooling layers (Riesenhuber & Poggio, 1999; Serre et al., 2005; Scherer et al., 2010) instead of sub-sampling layers. The output of a max-pooling layer is given by the maximum activation over non-overlapping rectangular regions of size  $K_x \times K_y$ . Max-pooling creates slight position invariance over larger local regions and down-samples the input image by a factor of  $K_x$  and  $K_y$  along each direction.

## **Classification layer**

Kernel sizes of convolutional filters and max-pooling rectangles are chosen such that either the output maps of the last convolutional layer are downsampled to 1 pixel per map, or a fully connected layer combines the outputs of the last convolutional layer into a 1D feature vector. The last layer is always a fully connected layer with one output unit per class in the recognition task. We use a softmax activation function for the last layer such that each neuron's output activation can be interpreted as the probability of a particular input image belonging to that class

## **Forming the MCDNN**

MCDNN is formed by averaging the output activations of several DNN columns. For a given input pattern, the predictions of all columns are averaged. Before training, the weights of all columns are randomly initialized. Various columns can be trained on the same inputs, or on inputs preprocessed in different ways. If the errors of P different models have zero mean and are uncorrelated, the average error might be reduced by a factor of P simply by averaging the P models.

## **2.5.Lung cancer detection using convolutional neural networks**

Image processing techniques provides good quality tools for improving manual analysis of lung cancer detection. There are many papers that had written based on neural networks.

When talk about the deep convolutional neural networks, although convolutional neural networks had presented well in many tasks, few work had been presented using convolutional neural networks for lung nodules classification.

'Lung Nodule Classification based on deep convolutional neural networks (CNN)' by J.C. Mendoza and H.Pedrini [10] is a paper which presents a method for classifying nodules based on CNN.

Detecting of lung nodules involves with three main stages.

- 1) Lung segmentation
- 2) Candidate detection

### 3) Nodule classification

This paper evaluates CNN for lung nodule classification by performing the candidate analysis directly on the pixels of images using a CNN, instead of extracting features and using classifiers. The CNN had trained through stochastic gradient descent (SGD) algorithm. The largely generated false positive outputs compared to true positives had been addressed in this paper by balancing the mini-batches on each SGD iteration. This approach had outperformed the results obtained with a feature-engineering approach. They have used DeepCNETs which composed with alternating convolutional and max-pooling layers with a linearly increasing number of filters. The evaluation had been done with **3-6 convolutional layers**. The better results have been shown by **6 convolutional layers**.

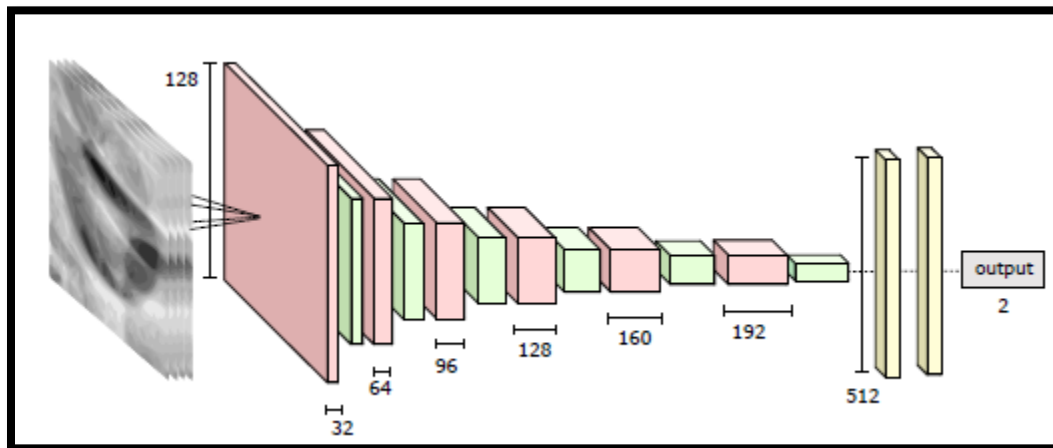


Figure 21-Nodule candidates are fed to the network composed of six alternating convolution and max-pooling layers, and two fully connected layers

The paper ‘Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans’ by J.Cheng, D.Ni et.al[31] has used the ‘Stacked Denoising Auto-Encoder (SDAE)’ architecture on two CADx applications to differentiate breast ultrasound lesions and lung CT nodules. The SDAE is mentioned as a well-equipped technique for automatic feature exploration mechanism and noise tolerance. They have outperformed two conventional CADx algorithms and shown a significant performance boost.

‘Deep Learning for the Classification of Lung Nodules’ by H. Yang & H. Yu[32] had used Deep Convolutional neural network Neural Network to thoracic CT images for the classification of lung nodules. They have used the first layer of the CNN as a convolution layer with filter of size  $7*7*1*20$ , stride of size 1 and no padding. There was a max pooling layer of size  $2*2$  with stride of size of 2. The third layer has been also a convolutional layer, with filter size  $7*7*20*50$  and the same stride as layer-1. This CNN architecture has not been unique. For the datasets they used, each input image of size  $50*50*1$  leads to an output of size  $1*1*2$  after forward propagation of the 9 layers.

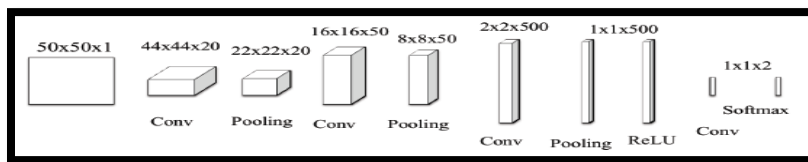
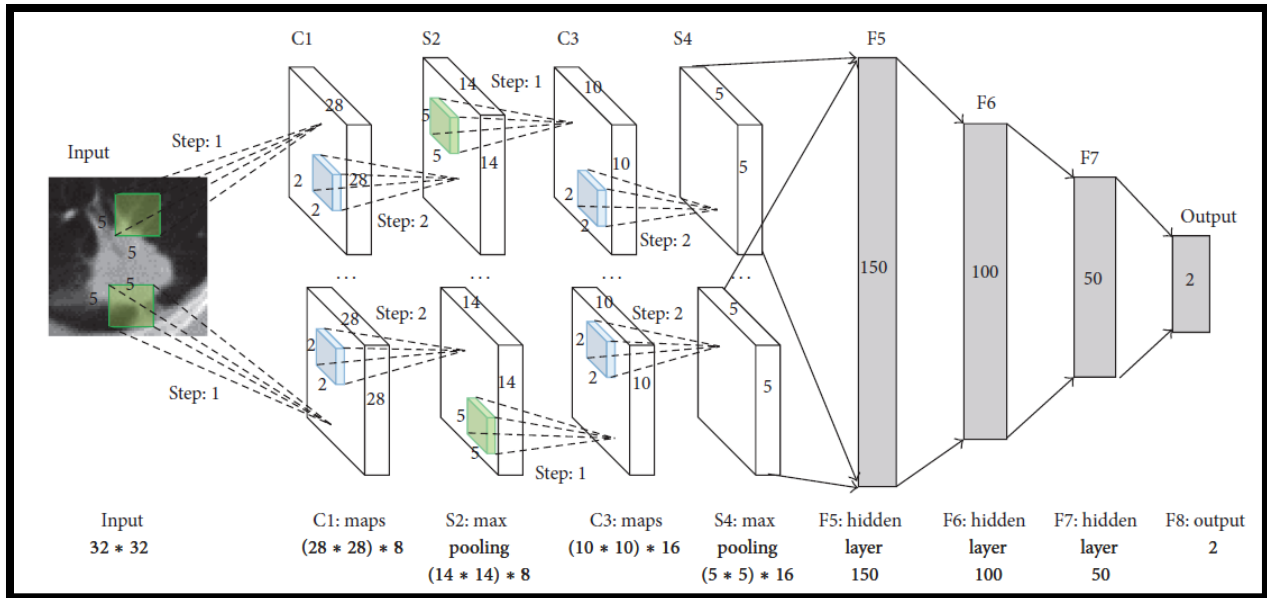


Figure 22- CNN architecture from the input image of size  $50*50*1$  to the final output.

Paper ‘Pulmonary Nodule Classification with Convolutional Neural Networks’[33] by S. Ramaswamy and Karen Truong mentions that they had used, two different CNN architectures that proved successful in ImageNet challenge. ‘AlexNet’ and ‘GoogLeNET’ were the two convolution neural networks that had used. Instead of taking them individually they had followed an ensemble approach of architecture.

‘Pulmonary Nodule Classification with Deep Convolutional Neural Networks on Computed Tomography Images’ by Wei Li, Peng Cao, Dazhe Zhao and Junbo Wang had constructed a CNN on  $32*32$  ROI data as shown below.



**Figure 23- Architecture of CNN for nodule recognition. The number of neurons in layers are set to 6272, 1568, 1600, 250, 150, 100 and 50.**

These convolutional neural networks have two convolutional layers and there is a down sampling layer behind the convolutional layer. Fully connected layers are appended to the last down sampling layer. The first convolutional layer contains 8 feature maps, and the second has 16 ones. The kernel size is 5\*5 in all convolutional layers and the step of kernel is 1. The kernel size is 2\*2 for all the downsampling layers and the step is 2. The first fully connected layer contains 150 nodes and there are 100 nodes in the second fully connected layer. There are 50 nodes in the third fully connected layer and the last layer only has two nodes which were presented as output probabilities of nodule and non-nodule.

Through the training of the dataset, more general characteristics of pulmonary nodules and higher accuracy could have been obtained by their suggested approach. They have proposed that the algorithm can be accelerated by GPU computing for convolution operation.

Now let's have a look on current development of very deep convolutional networks for large scale image processing.

The paper, 'Very deep convolutional networks for large scale image recognition' by K. Simonyan and A. Zisserman [11], had investigated on how the effect of convolutional network works on the

accuracy of large-scale image recognition. Their main contribution is an evaluation of using very small (3\*3) convolution filters by pushing the depth to 16-19 weighted layers. They had fixed the parameters of ConvNets architecture, and steadily increased the depth of the network by adding more convolutional layers, which is feasible due to the use of very small (3\*3) convolutional filters in all layers. As a result, they could have come up with significantly more accurate ConvNet architectures, which not only achieve the state-of-the-art accuracy on ILSVRC classification and localization tasks, but also applicable to other image recognition datasets.

### **Architecture**

- 1) Inputs – 224\*224 RGB images.
- 2) Preprocessing step – Subtracting the RGB value computed on the training set from each pixel.
- 3) Image passes through a number of convolutional (conv.) layers, which had used very small receptive field: 3\*3.
- 4) The convolutional stride – 1 pixel
- 5) Padding – 1 pixel for 3\*3 convolutional layers.
- 6) Number of Max pool layers – 5 (Performed over 2\*2 pixel window, with stride 2)

A stack of convolutional layer had been followed by three Fully-Connected (FC) layers: the first two have 4096 channels each, the third performs 1000 classification and thus contains 1000 channels. The final layer has been a softmax layer. The configuration of the fully connected layers had been the same in all networks.

### 3. Methodology

Many papers had been written on lung cancer detection using deep neural networks. Since deep neural network is a rapidly improving area, there are still many un touched areas of deep neural networks that can be used in lung cancer nodules detection.

Deep neural networks compose of number of processing layers. They involve in deep as well as nonlinear operations. These methods are now dominant approaches for feature extraction. The sections above mainly discussed on how deep neural networks have been used in lung cancer nodule detection.

By taking a step ahead, this chapter discusses on how ‘Very deep Neural Networks’ will be useful in lung cancer nodule detection.

The Methodology of this research mainly consists of five sections as shown in figure 24. This research mainly consists of two sections.

- 1) Process NDSB 2017 dataset to detect lung cancer with the help of LUNA16 dataset.
- 2) Process the annotated LUNA16 dataset alone to detect lung cancer using its annotated slices

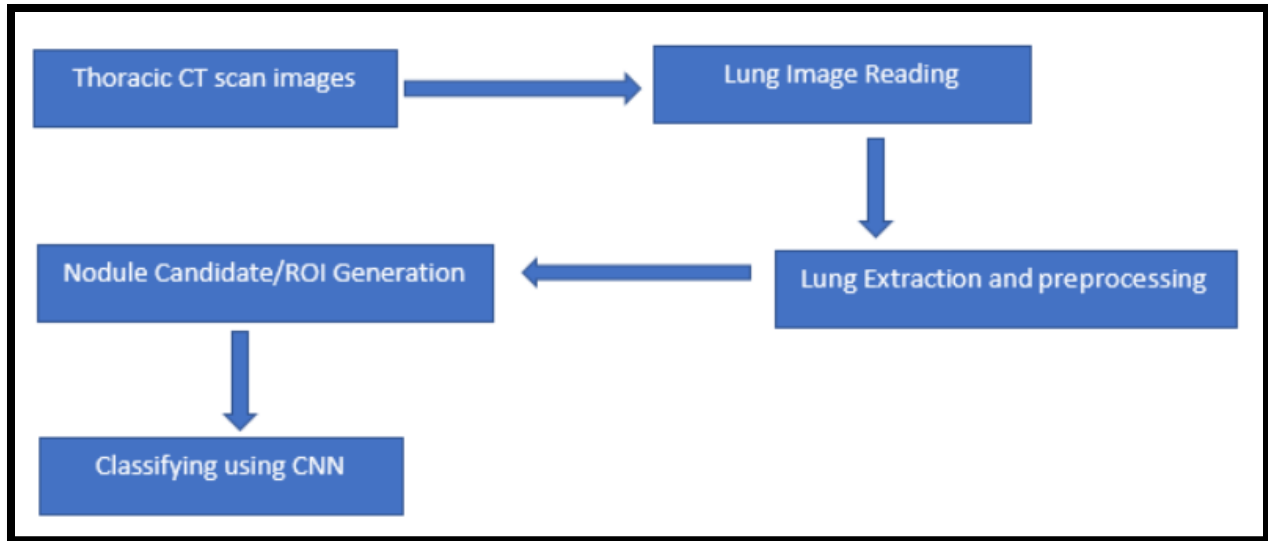


Figure 24: Methodology proposed by the Research

### 3.1. Thoracic CT scan images reading

The first step of the methodology is to read Thoracic CT scan images in order to have a good understanding about our dataset. The initial part of this chapter will discuss on how the data have varied and how it is formatted. Mainly there are two publicly available lung cancer related datasets, one with annotated Thoracic Lung CT images and other with non-annotated but labeled Thoracic Lung CT scan images.

The first dataset is provided by the Lung Image Database Consortium (LIDC-IDRI), which consists of lung cancer screening CT scans with marked annotated lesions. Slice thickness smaller than 2.5mm have been considered for this dataset. The LUNA16 challenge is a computer vision challenge essentially with the goal of finding ‘nodules’ in CT scans. It contains about 900 CT scans. The LIDC/IDRI dataset also contains annotations which are collected during a two-phase annotation process using 4 experienced radiologists. Each radiologist marked lesions they identified as non-nodules where the nodule is less than 3mm. This data set contains some essential information, including the location of 2669 nodules (At least one radiologist assigned a nodule larger than 3mm mark). The LUNA dataset also contains radiologist descriptions of each nodule,



such as nodule malignancy, nodule diameter, nodule speculation (how ‘stringy’ a nodule is-more worse) etc.

The second dataset of this research was observed by the ‘National Data Science Bowl 2017’(NDSB) competition which held in early 2017 to convene data science and medical communities to develop lung cancer detection algorithms [35]. The dataset contains 1500 scans of cancerous as well as of non-cancerous patients. Those data are in the sizes of 512 x 512 pixels. Each patient is entitled for a folder and it contains slices of CT images provided in DICOM format. DICOM (Digital Imaging and Communications in Medicine) is standard for handling, storing, printing, and transmitting information in medical imaging.

Since this NDSB dataset is lack information about nodule, it is inadequate to accurately classify the validation set. Hence, we were planned to use a lung CT scan dataset with labeled nodules from the Lung Nodule Analysis 2016 (LUNA16) Challenge [36] to train a U-Net for Region of Interest segmentation.

### **3.1.1. Lung Image Reading**

Lung images in the datasets are in DICOM (NDSB) and mhd (LUNA16) format. There are inbuilt methods in Python to read DICOM MHD images and to save the images inside Python lists. The unit of measurement of CT scans is Hounsfield Unit (HU), which is a quantitative scale for Radio Density. CT scanners are calibrated to accurately measure these values.

How these Hounsfield units can be used for lung segmentation and extraction is explained in the next section. When Hounsfield units of the CT scan images are considered, we can extract the Lung from other non-related parts of the CT scans.

## **3.2.Lung Extraction and Preprocessing**

Segmentation of lung is challenging because of differences in pulmonary (relating to lung) inflation with an elastic chest wall (Denotes the rebound of the lungs) after having been stretched by inhalation. With inhalation, the pressure with the pleural cavity of lungs decreases and can

create large variability in volumes and margins when attempting to automate the segmentation of lungs.

The first step of preprocessing is the lung structure segmentation, because, the region of interest (RIO) lies inside the lungs. The lungs are the darker regions in the CT scans and the bright area inside the lungs are the blood vessels or the air. A threshold of 604(-400HU) is used at all the places because, in experiments it has found that this value works fine. The lung segmentation was done using each slice of the CT scan. The region of interest attached to lung wall should be protected as there are many nodules attached to the wall [40].

Once the lung segmentation is done from the CT image, the target should be to find the candidate region with nodules as the search space is very large. Also, the whole image cannot be classified using CNN due to limitations in computation. Therefore, it is required to find possible regions of cancer and then trying to classify them. The past research shows that all ROI have intensity  $>604$  (-400HU). Therefore, this approach was used to filter the darker regions. This reduced the number of candidates by a large number and preserved all important regions with high recall.

### **3.3.Nodule Candidate / ROI generation**

Feeding the segmented images directly to the neural network to classify the images either as cancerous or not is not considered as a good approach as the ROI space is too large in the CT scans to be classified by the neural network directly. Therefore, LUNA16 dataset can be used to train the 2D neural network, UNet, to segment the candidates from CT scans, so later the trained UNet can be used to separate the ROI space of CT scans in Kaggle dataset prior sending them to final deep neural networks to classify.

### **3.4.Classifying using CNN**

The main purpose of this research is to use ‘Very Deep Convolutional Neural Network’ approach to detect pulmonary lung nodules using Lung CT Scan images. There is enough space to enhance cancer nodule detection approaches when Deep Neural Network is concerned.

In chapter 2, section 2.4 has discussed about well performing ‘Very deep neural network architectures’ that has been used for computer vision during past years.

Residual neural networks (ResNets) [41] is such an exciting neural network for deep learning research, introduced by Microsoft in 2015. ResNets allow very deep neural networks to be effectively trained. ResNets has won the ImageNet competition in December with 3.57% error score. Recently, researchers have published several papers augmenting the ResNet model with many interesting improvements. ResNet uses the Mathematical formula for ‘Very deep neural networks’. They tweak the layers’ equations to introduce identity connections between them. The identity function is simply  $\text{id}(x)=x$ ; given an input  $x$  which returns the same value  $x$  as output. A layer in a traditional neural network learns to calculate a function  $y=f(x)$ . But the residual neural network layer approximately calculates  $y=f(x)+\text{id}(x) = f(x)+x$ . These identity connections enable the layers to learn incremental or residual representations. The layers can be started as the identity function but can gradually transform to be more complex. Such evolution occurs if parameters for the  $f(x)$  part begin at or near zero.

The residual neural network layer approximately calculates,

$$Y = f(x) + \text{id}(x) = f(x) + x$$

The ResNet technique has shown that deeper neural models can train effectively and successfully. The model which won the ImageNet 2015 has had 152 layers. This is significantly the deepest model compared to other winners in previous years. When models get deeper they hit obstacles during the training process. The gradient signal vanishes with increasing network depth or in other words when the network depth increases, accuracy gets saturated and degrades rapidly. This degradation doesn’t happen from overfitting. Adding more layers to a suitably deep model leads to higher training error as reported in [42,43]. But the specialty of ResNet is, it propagates the gradient throughout the model. In paper [41], the degradation problem has been handled by introducing ‘deep residual learning’ framework. Instead of hoping each few stacked layers directly fit a desired mapping, this model explicitly let these layers fit a residual mapping. Also, this ResNets deep neural network can easily enjoy accuracy gains from greatly increased depth,

producing results substantially better than previous networks. 152-layer residual net is the deepest network ever presented on ImageNet, while still having lower complexity than VGG nets. Their ensemble has 3.57% top-5 error on the ImageNet test set and won the first place in ILSVRC 2015 classification competition. Also, it says that the extremely deep representations also have excellent generalization performance on other recognition tasks. Therefore, it is obvious that the residual learning principle is generic, which can be applicable in other vision and non-vision problems as well.

Transfer Learning is a machine learning method where a model developed for a task can be reused as the starting point for a model on a second task. This approach is popular in deep and very deep learning where pre-trained models can be used as the starting point specially on computer vision and natural language processing tasks.

This form of transfer learning used in deep learning is called inductive transfer. This is where the scope of possible models (model basis) is narrowed in a beneficial way by using a model on a different but on a related task.

Now let's see how to use the Transfer Learning concept.

There are two common approaches that can be followed when using Transfer Learning.

- 1) Develop model approach
- 2) Pre-trained model Approach

### **3.4.1. Develop model approach**

- 1) Select Source Task

You must select a related predictive modeling problem with an abundance of data where there is some relationship in the input data, output data, and/or concepts learned during the mapping from input to output data.

- 2) Develop source model

Next must develop e a skillful model for this first task. The model must be better than a naïve model to ensure that some feature learning has been performed.

3) Reuse model

The model fit on the source task can then be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used.

4) Tune model

The model may need to be adapted or refined on the input-output pair available for the task of interest

### 3.4.2. Pre-trained model approach

1) Select Source Model

A pre-trained model is chosen from available models. Many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose from

2) Reuse model

The model pre-trained can then be used as the starting point for a model on the second task of interest. This may involve using all parts of the model, depending on the modeling technique used.

3) Tune model

The model may be needing to be adapted or refined on the input-output pair data available for the task of interest.

Many research papers [47] have used transfer learning approach for cancer nodule detection. Transfer learning usually takes very deep neural networks to see if a model originally trained on a much more extensive (through qualitatively very different) dataset might allow to produce a model that would go beyond other strategies of predicting majority classes.

Therefore, we selected the ResNet-50 model [41] and VGG model which are deeper neural networks to train, test and validate our dataset. The residual network is considered easier to optimize and can gain accuracy from considerably increased depth. ResNet is many times deeper than the VGG networks but still considered as having a lower complexity than VGG model. ResNet model has won the first place on the ILSVRC 2015 classification task. The depth of representations is highly important for visual recognition tasks.

The core idea of ResNet is introducing a “identity shortcut connection” that skips one or more layers as shown in the following figure.

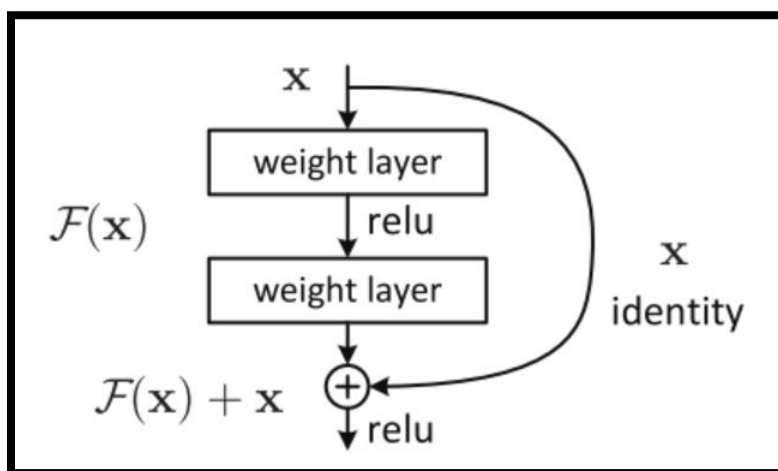


Figure 25 : A residual block

As authors of [41] said, the stacking layers do not reduce the network performance because it could simply stack identity mappings (layer that do not do anything) upon the current network, and the resulting architecture would perform the same. This shows that the deeper models should not produce a training error higher than its shallower counterparts.

In practice, it is hard to practice the entire convolutional neural network from the scratch because it is relatively hard to have a dataset with a sufficient size. It is common to pretrain a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest.

The three major Transfer Learning scenarios look like following.

1) Convnet as fixed feature extractor

In this step, a ConvNet pretrained on ImageNet is selected (ResNet-50 layer network), and the last fully connected layer was removed. Last output layers consist of 1000 classes, but scores for a different task such as ImageNet. Then the rest of the ConvNet after removing the last layer is selected as a fixed feature extractor for the new dataset.

2) Fine-tuning the ConvNet

The second strategy that can be applied is not only to replace and retain the classifier on top of the ConvNet on the new dataset, but also to fine-tune the weights of the pretrained network by continuing the backpropagation. It is possible to fine-tune all the layers in the network. This is motivated by the observation that the earlier features of the ConvNet contain more generic features (e.g. edge detectors or color blob detectors) that should be useful for many tasks, but later layers of the Convnet becomes progressively more specific to the details of the classes contained in the original dataset. In case of ImageNet for example, which contains many dog breeds, a significant portion of the representational power of the ConvNet may be devoted to features that are specific to differentiation between dog breeds.

3) Pretrained models

Since modern ConvNets take 2-3 weeks to train across multiple GPUs on ImageNet, people release their final Convnet checkpoints for the benefit of others who can use the network for fine-tuning.

Following explains what type of transfer learning that we should perform on a new dataset

This is a function of several factors but consists with two main points.

- 1) The size of the new dataset (small or large)
- 2) The original dataset

The ConvNet features are more generic in early layers and more original-dataset-specific in later layers. Following displays the two-main rule of thumb for navigating the four major scenarios [48]

- 1) New dataset is small and like original dataset

Since the dataset is small, it is not good to fine-tune the ConvNet due to overfitting concerns. Since the dataset is similar to the original dataset, we can expect higher-level features in the Convnet to be relevant to this dataset as well. Hence, the best idea is to train a linear classifier on the CNN codes.

- 2) New dataset is large and similar to the original dataset

Since we have more data, we can have more confidence that overfitting will be not happened if we try to fine-tune through the full network

- 3) New dataset is small but very different from the original dataset

Since the dataset is small, it is likely best to only train a linear classifier. Since the dataset is very different it might not be best to train the classifier from the top of the network, which contains more data specific features. Instead it might work better to train the SVM classifier from activations somewhere earlier in the network

- 4) New dataset is large and very different from the original dataset

Since the dataset is very large, we may expect that we can afford to train a ConvNet from scratch. However, in practice it is often still beneficial to initialize with weights from pretrained model. In this approach, there are enough data and confidence to fine-tune through the entire

For classification, transfer learning with a 50-layer residual network (ResNet) was selected to go ahead with this research. ResNet models have been designed to ease backwards propagation of gradients through the network and to improve training. As explained above, the ResNet has generated based on small stacks of convolutional layers. The input for the network is summed with



the output of the layers to create skip connections. In the 50-layer ResNet, the small layer stack between skip connections consists of three convolution layers (1x1 filter, 3x3 filter, 1x1 filter) and is called a ‘bottleneck’ as the 1x1 filters decrease and then restore dimensions to speed computation [44]. The ResNet used in this study was implemented in Caffe [45] and pre-trained on a subset of the ImageNet dataset with about 1.3 million labels color images of common animals and objects in 1000 classes.

During the implementation, CT scanned images will be feed forward through the ResNet with fixed pre-trained weights, and features will be extracted from different layers. Earlier Convolutional Neural Network layers generally pick out edges and generic visual features that may be more transferable to new classification tasks, while later may be more specific to the initial ImageNet classes. On the other hand, deeper networks generally perform better in image classification as they allow for more non-linear relationships between pixels and output classes.

Therefore, in this paper, we explore extracting features from multiple layers spread throughout the network to determine which features are best for the nodule classification task. How the methodology was implemented will be explained in the next chapter

## **4. Implementation**

### **4.1. Chapter Overview**

This section will discuss on how NDSB 2017 dataset was used to classify its CT scans as cancerous or not.

### **4.2. Using NDSB 2017 dataset to detect lung cancer**

#### **4.2.1. Thoracic CT scan images reading**

Preprocessing is required before the images are ready for consumption by suggested convolution Neural network. The primary dataset used during the research is the Kaggle Data Science Bowl (DSB). This dataset contains 1397 patients, where each CT scan is labeled as 0 or 1. (0 for no cancer, 1 for cancer). This Kaggle data set does not have labeled nodules. Around 70% of the provided labels are 0. Hence, weighted loss function has been used in malignancy classifier to address this imbalance.

Kaggle dataset alone is inadequate to accurately classify the validation set. Therefore, we used the Lung Nodule Analysis 2016 (LUNA16) to train a U-Net for Lung nodule detection. This dataset contains labeled data for 888 patients. For each patient, the data consists of CT scan data and a nodule label. (List of Nodule center coordinates and diameter). In this dataset as well, the CT scan data consists of variable number of slices of 512x512 pixels.

LUNA 16 dataset was used to train a U-Net for nodule detection, which is a main part in classification pipeline. The main aim of this research is to accurately predict a patient's label (cancer or not-cancer) based on the patients Kaggle lung CT scan. Accuracy, Sensitivity, Specify, and F1-score will be used to evaluate our CAD system's performance on Kaggle dataset.

## ***Loading the DICOM files and adding missing data***

The main programming language used for processing the data is Python 3.5 (64-bit version). Python is a powerful, open-source language which includes Pandas, Git , Matplotlib and many more to manipulate , analyze and visualize complex datasets.

Following were the basic libraries in Python used for data manipulation

### 1) Numpy + MKL

Numpy is the fundamental package for scientific computing with python. It contains powerful N-dimensional array object, sophisticated (broadcasting) functions, tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform and random capabilities. It is also an efficient multi-dimensional container of generic data. Arbitrary data types-can be defined.

### 2) Pandas

This provides high-performance, easy-to-use data structures and data analysis tools for Python programming. It aims to be the fundamental high-level building block for doing practical, real world data analysis of python. Pandas is built on top of Numpy and intended to integrate will within a scientific computing environment with many other third-party libraries

### 3) DICOM

DICOM is a worldwide standard in health IT and provided by National Electrical Manufactures Association (NEMA). DICOM stands for Digital Imaging and Communications in Medicine. The standard specifies the way medical images and meta data like study or patient related data is stored and communicated over different digital medias. Python provides a library to read these data.

### 4) SCIPY

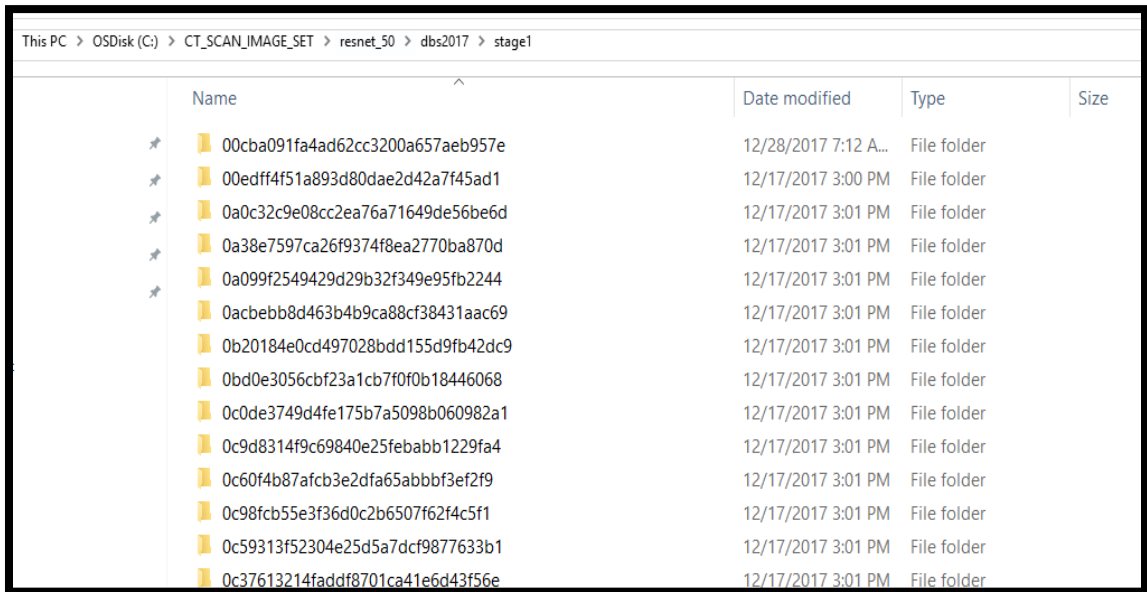
Scipy is open-source software for Mathematics, Science and Engineering. The Scipy library depends on Numpy, which provides convenient and fast N-dimensional array manipulation. The Scipy library is built towork with Numpy arraysand provides many user

friendly and efficient numerical routines such as routines for numerical integration and optimization.

## 5) MATPLOTLIB

Matplotlib is a plotting library for Python programming language and it is an extension of Numpy. It is possible to generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots etc.

As explained earlier, a scan of a single patient has number of slices of the lung. The first step was to load the patients with its slices to a Numpy array.



Name	Date modified	Type	Size
00cba091fa4ad62cc3200a657aeb957e	12/28/2017 7:12 A...	File folder	
00edff4f51a893d80dae2d42a7f45ad1	12/17/2017 3:00 PM	File folder	
0a0c32c9e08cc2ea76a71649de56be6d	12/17/2017 3:01 PM	File folder	
0a38e7597ca26f9374f8ea2770ba870d	12/17/2017 3:01 PM	File folder	
0a099f2549429d29b32f349e95fb2244	12/17/2017 3:01 PM	File folder	
0acbebb8d463b4b9ca88cf38431aac69	12/17/2017 3:01 PM	File folder	
0b20184e0cd497028bdd155d9fb42dc9	12/17/2017 3:01 PM	File folder	
0bd0e3056cbf23a1cb7f0f0b18446068	12/17/2017 3:01 PM	File folder	
0c0de3749d4fe175b7a5098b060982a1	12/17/2017 3:01 PM	File folder	
0c9d8314f9c69840e25febabb1229fa4	12/17/2017 3:01 PM	File folder	
0c60f4b87afcb3e2dfa65abbbf3ef2f9	12/17/2017 3:01 PM	File folder	
0c98fcb55e3f36d0c2b6507f62f4c5f1	12/17/2017 3:01 PM	File folder	
0c59313f52304e25d5a7dcf9877633b1	12/17/2017 3:01 PM	File folder	
0c37613214faddf8701ca41e6d43f56e	12/17/2017 3:01 PM	File folder	

**Figure 26 : Patient wise Folder Structure**

The above shows how the patients' folders will be displayed in the stage1 folder. Each folder is owned by a patient and it is called a scan. Inside each of these folders there are dicom files with the extension, dcm.

Name	Date modified	Type	Size
0a291d1b12b86213d813e3796f14b329.dcm	1/12/2017 4:20 AM	DCM File	514 KB
0af07f64d14d6b0451036cc742a96fbf.dcm	1/12/2017 4:20 AM	DCM File	514 KB
0b8fdd1863b4362453001a94123ab40f.dcm	1/12/2017 4:20 AM	DCM File	514 KB
0ba4f3cf525a9da7e02e181957ca2bd8.dcm	1/12/2017 4:20 AM	DCM File	514 KB
0cd80c17d64e56cc3b9f5d7229880a14.dcm	1/12/2017 4:20 AM	DCM File	514 KB
0d546a5ae1473f772e6a8f271a049535.dcm	1/12/2017 4:20 AM	DCM File	514 KB
1aeb4bab6fc11af5fa8923cdcdec4c5.dcm	1/12/2017 4:20 AM	DCM File	514 KB
1d07f66753a81c3c6a4a1b3b55df860d.dcm	1/12/2017 4:20 AM	DCM File	514 KB
1daaf62f759aa843ee298851e3d87722.dcm	1/12/2017 4:20 AM	DCM File	514 KB
1e1f60ffa2fa56e5f25665ba732f3fee.dcm	1/12/2017 4:20 AM	DCM File	514 KB
1fb3cbe71ea9e0bfb08b0c68a20abe0b.dcm	1/12/2017 4:20 AM	DCM File	514 KB
2cca7587b340477df0e01b9a36679b6b.dcm	1/12/2017 4:20 AM	DCM File	514 KB
2e56c90ec40b3ea7e4da9fb626609317.dcm	1/12/2017 4:20 AM	DCM File	514 KB
3ba0cad671e5514564aaaa939bbc8221.dcm	1/12/2017 4:20 AM	DCM File	514 KB

**Figure 27 : Original images in DICOM format**

These images are called slices of a patient. These slices are taken from different angles to produce cross-sectional images of specific area of the scanned lung. These images allow the user to see inside the object without cutting. The combination of all these .dcm files compose a cancerous or the non-cancerous 3D lung image of the patient.

When the path of the patient folder is given, the slices of the patient were loaded to a python list as following.

```
slices = [dicom.read_file(path + '/' + s) for s in os.listdir(path)]
```

The slice thickness values are determined by the operator in accordance with clinical examination requirements; its values usually fall from 1 to 10 mm. Image noise is inversely relational with the variation square root in the beam thickness (slice thickness). Thinner slice thickness increases image noise [45]. Despite thinner slices increasing the image noise. The isomorphic resampling will be done later to reduce this issue.

The ‘Implementation’ chapter mainly consists of three steps

- 1) Lung Segmentation approach

- 2) Nodule Segmentation approach
- 3) Nodule Classification approach

Following sections will discuss more on these three areas in detail.

#### 4.2.2. Lung segmentation approach

The lung segmentation was done mainly by using following methods.

##### *Thresholding method*

The radio density of a CT scan is measured by the Hounsfield Unit(HU). CT scanners are carefully practiced to accurately measure these values. Following HU results are the results observed from Wikipedia related to different Substances contained in the CT images.

Substance	HU
Air	-1000
Lung	-500
Fat	-100 to -50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

Figure 28: HU results of the results observed from Wikipedia related to different Substances contained in the CT images

By default, the slices are not in HU so a conversion need to be done before using it further. Following approaches were used to convert the slices into images.

```
image = np.stack([s.pixel_array for s in slices])
```

‘slices’ was a numpy array which stores the dicom files related to a patient (scan). It contains number of slices and the number of slices of a patient will be varied from patient to patient. The above code creates a numpy stack which contains the pixel array of each slice. Then it is stored in a variable named as image. Now this ‘image’ variable contains a full stack of pixel arrays related to all the slices of a patient.

This is how a slice of a randomly selected scan will be looked when it is displayed.

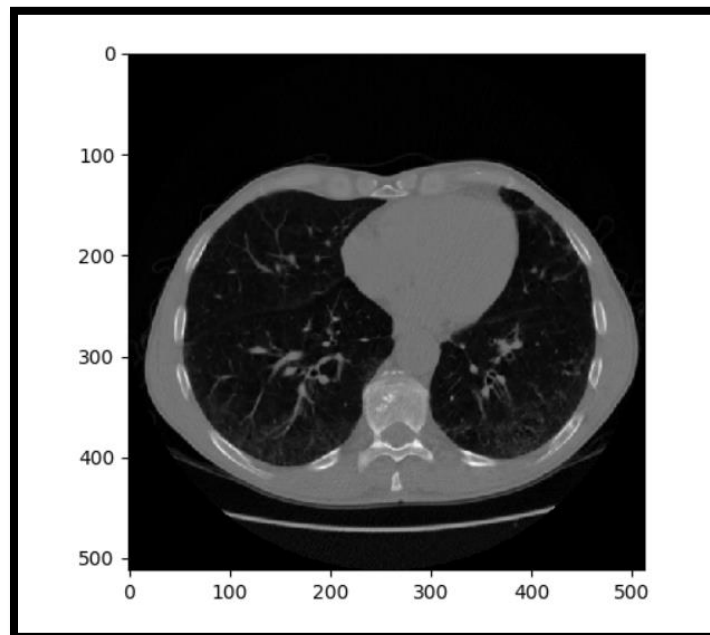


Figure 29 : View of a Slice

The lung includes vessels, airways, major fissures, bronchi, apex, carina etc. as showing in following figures.

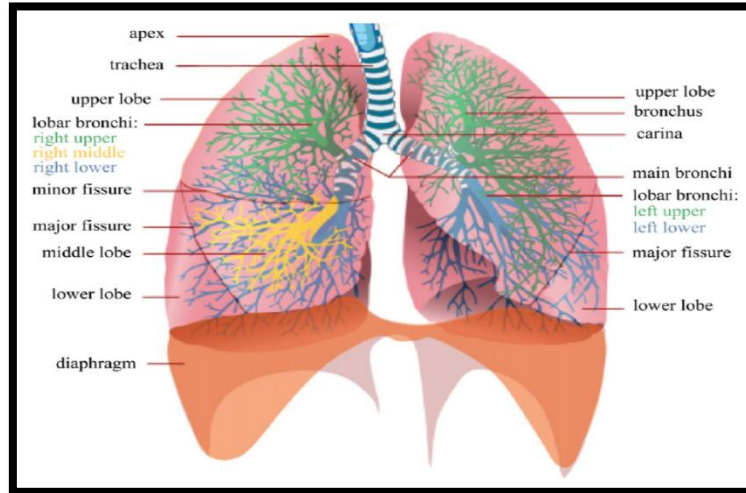


Figure 30 : View of Vessels and Airways etc.

Some scanners have cylindrical scanning bounds as shown below, but the output image is square.

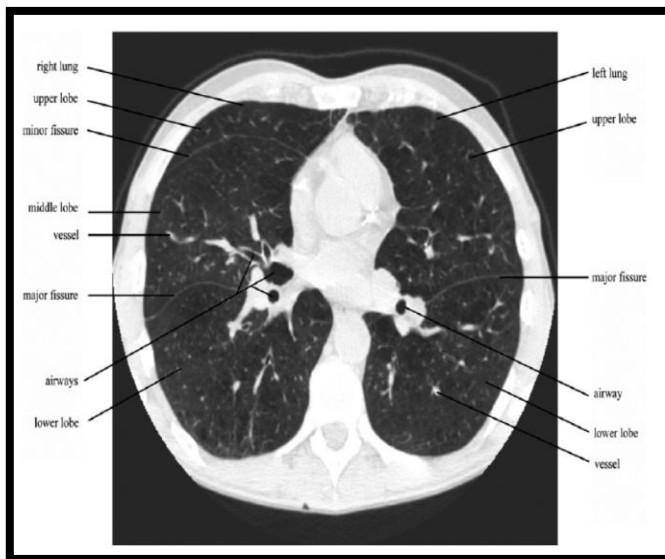


Figure 32 : Cylindrical Scanning Bounds



Figure 31 : Output Image

The pixels that fall outside of these bounds get a fixed HU value -2000. The first step should assign these HU values to 0 as it is corresponded to the air. Converting the slices to HU units can be done



by multiplying with the rescale slope and adding the intercept. (These values are stored as the metadata of DICOM files) The rescale slope and rescale intercept allow to transform the pixel values to HU or other units. For CT images, the unit should be HU and the default value is indeed HU when the tag 0028, 1054 is not present. The rescale slope and intercept are usually defined by the manufacturer of the hardware.

Once the image converted into Hounsfield units, this is how the HU will be displayed for a slice in a randomly selected scan.

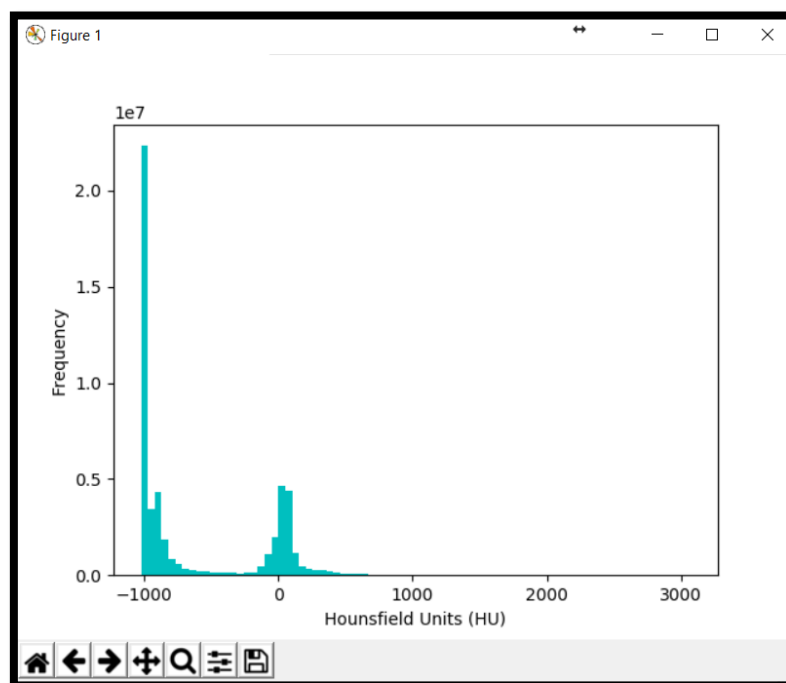
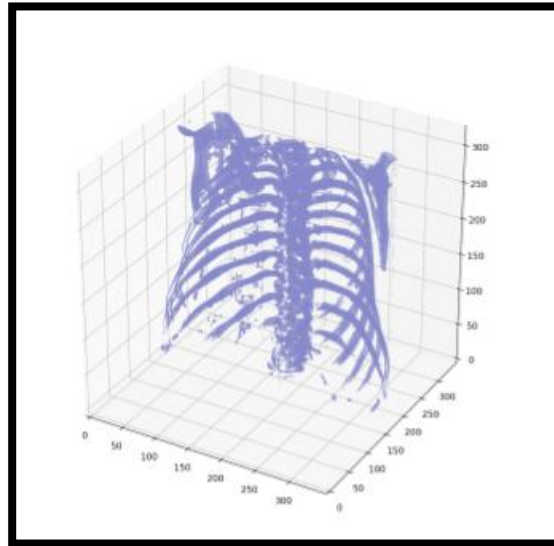


Figure 33 : Results in Hounsfield units

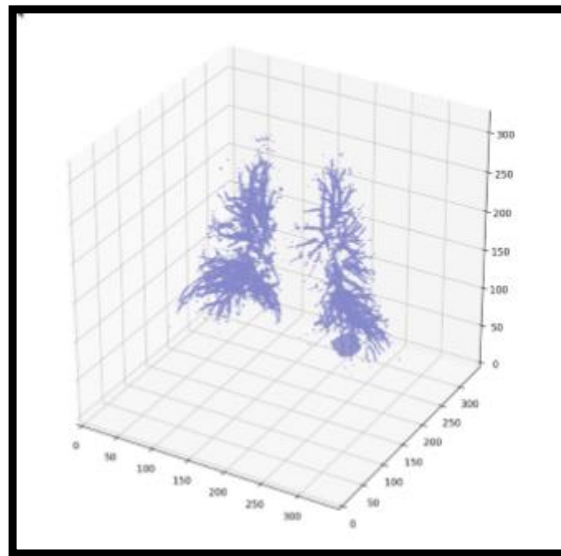
The HU value related to the Air is -1000, water and blood is 0, bone is 700 and lung tissue is -500. In Figure 33, which pixels are related to air and which pixels are related to tissue is displaying. This concept will be used for lung segmentation.

A randomly selected patient's 3D scan with pixels values greater than 400 HU tends to display the bone structure and it is displayed in Figure 34.



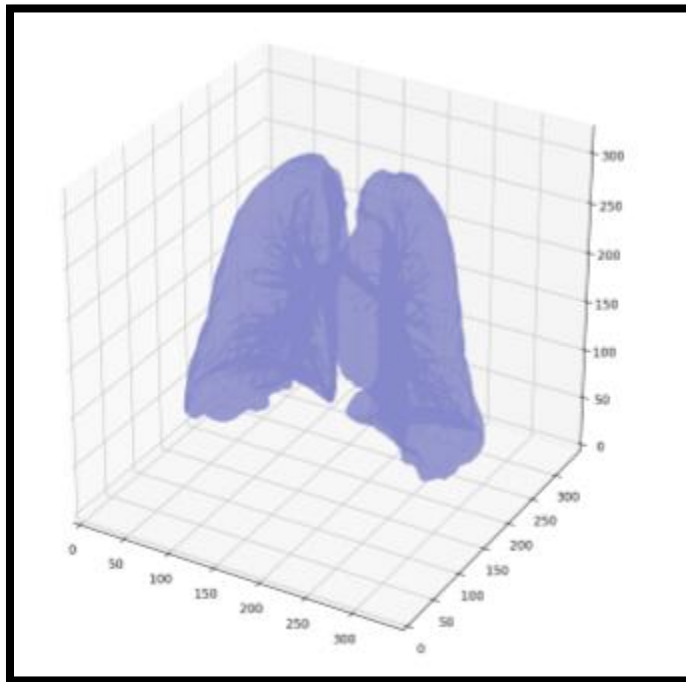
**Figure 34 : Patient 3D scan with pixels greater than 400HU**

Sample patient's bronchioles within lung



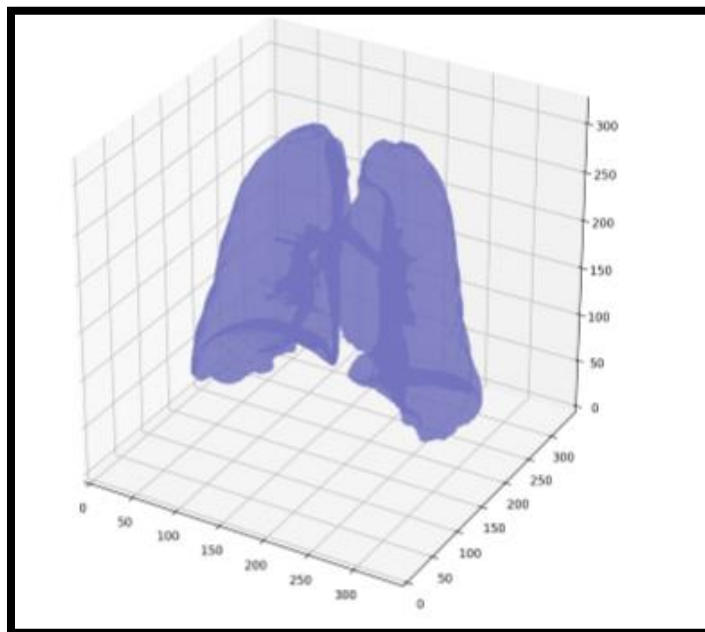
**Figure 35 : Patient Bronchiole within lung**

Sample patient's initial mask with no air



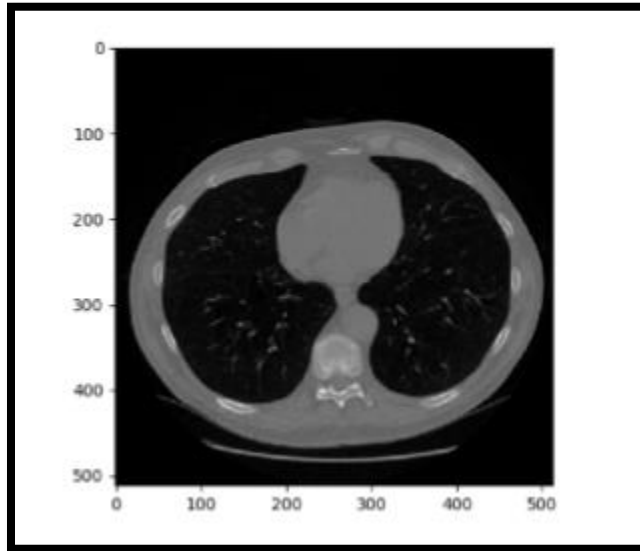
**Figure 36 : Patient initial mask with no air**

Sample patient final mask when bronchioles were included

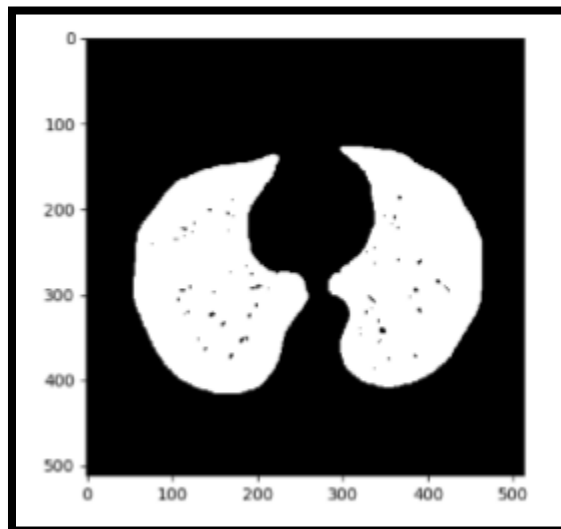


**Figure 37 : Patient final mask when bronchioles are included**

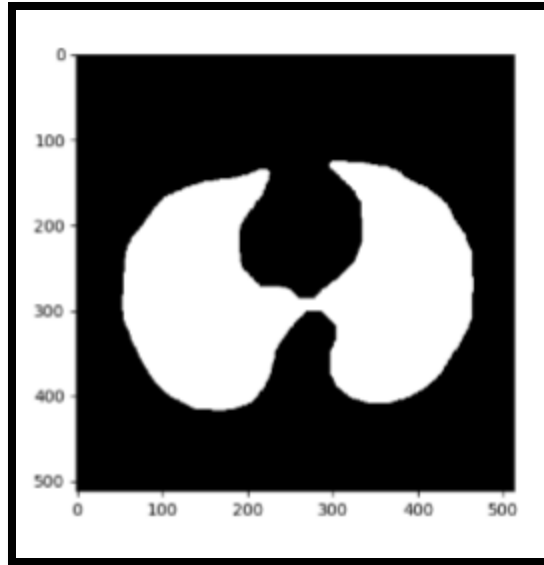
Sometimes cancer can be grown in bronchioles (air pathways) inside the lung. Therefore, to create the final mask, the air pathways were considered.



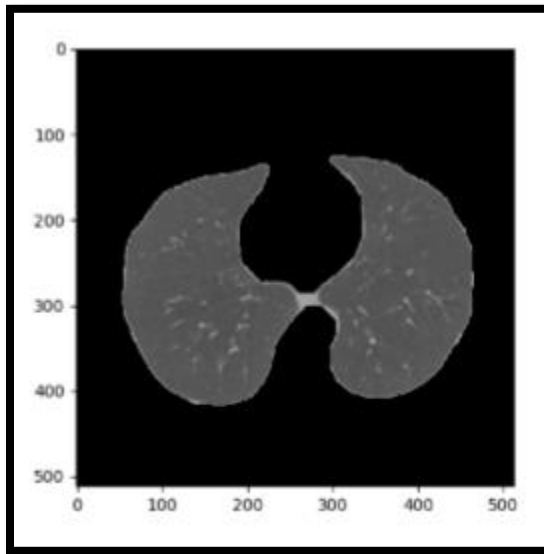
**Figure 38 : Original 2D slice of a patient**



**Figure 39 : Lung segmentation mask by thresholding of sample patient**



**Figure 40 : Final watershed segmentation mask of sample patient**



**Figure 41 : Final watershed lung segmentation of a sample patient**

Figure 38 shows an original image of a slice of a cancer patient. A resulting slice after thresholding has been shown in Figure 39. Figure 41 shows the final watershed lung segmentation of a sample patient. Qualitatively, this output in Figure 41, gives much better segmentation than only thresholding. In Figure 39, many of the voxels (black dots) are missing. But when the Watershed

approach is done, the missed dots could be successfully reincluded. But this approach is very much inefficient. And took quite a long time to complete the preprocessing part.

Once the preprocessing is done, the next step was to segment the suspected nodule areas from each slice of a given CT scan of a patient. Next section discusses how the Lung Nodule Segmentation was performed using the U-Net neural network.

### **4.2.3. U-Net for Lung Nodule Segmentation**

If the entire segmented lungs were feed to the malignancy classifier directly, it showed that the results made were poor due to the large searching space of the entire image. Therefore, feeding smaller ROI (Region of Interest) instead of the entire segmented lung is more appropriate. This was achieved by selecting small boxes containing top cancerous nodule candidates. To identify the top nodule candidates, U-Net neural network was trained as described by Ronneberger et al. [49]. U-Net is a popular, 2D Neural Network architecture which performs well in biomedical image segmentation. The visualization of the U-Net architecture is included in Figure 42.

During the training process, the U-Net takes input images with size 256x256 2D CT slices along with their labels. (256x256 mask where nodule pixels are 1, rest are 0).

The model has been trained to output images of shape 256x256 where each pixels of the output has a value in between 0 and 1 indicating the probability of the pixel belongs either to the non-nodule or nodule classes respectively. This is done by taking the slice corresponding to label one of the softmax of the final U-Net layer.

Corresponding U-Net inputs, labels, and predictions on a patient from the LUNA 16 validation set is shown in the following Figure 42.

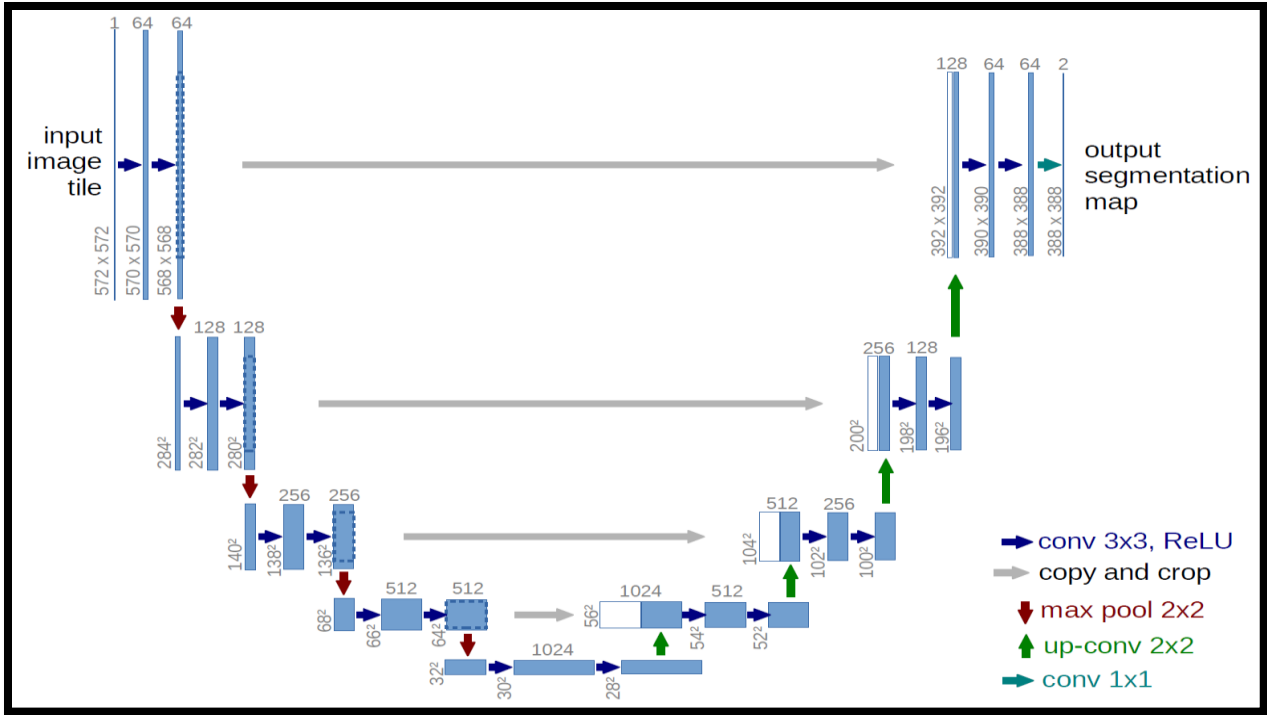


Figure 42 : U-Net Architecture

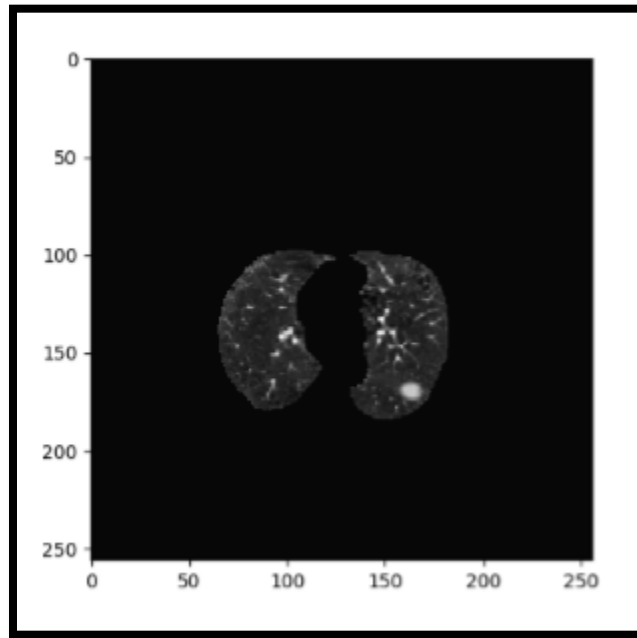


Figure 43 : U-Net sample input from LUNA16 validation set which is a segmented lung slice

Note that the above image has the largest nodule from the LUNA16 validation set, which we chose for clarity most nodules are significantly smaller than the largest one in this image.

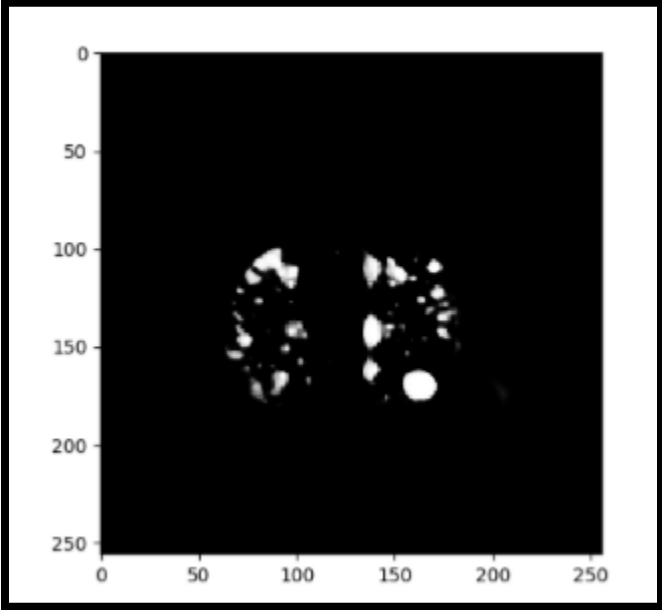


Figure 44 : Figure-U-Net predicted output from LUNA16 validation set

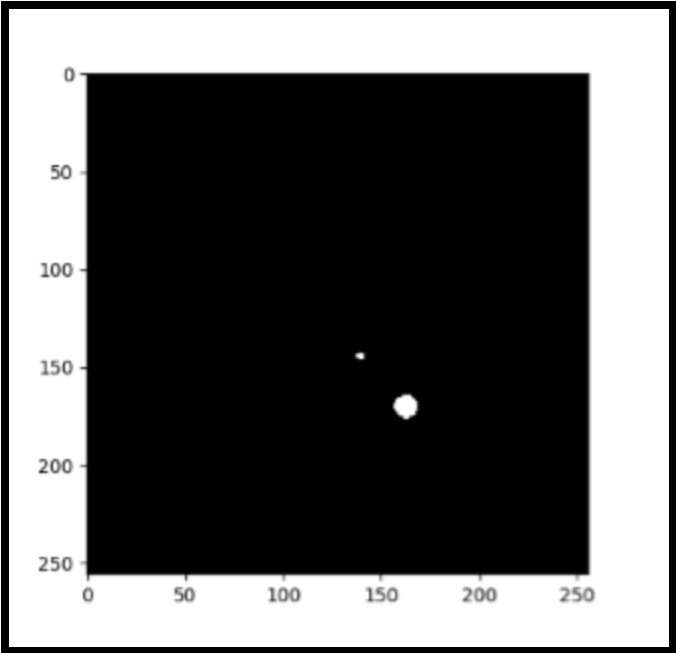


Figure 45 : U-Net sample labels mask from LUNA16 validation set showing ground truth nodule location



Most of the nodules are very much smaller. The trained U-Net is then applied to the segmented Kaggle CT scan slices to generate nodule candidates for each slice owned to each scan.

When the U-Net was trained on the LUNA 16 data, it should be run on 2D slices of Kaggle data and stack the 2D slices back to generate nodule candidates. In the best case, the output of U-Net would give the exact location of all the nodules. Also, it has the ability to identify the images with nodules as images positive for cancer and images without nodules as images negative for cancer. But the issue is U-Net produces a strong signal for the actual nodule, but also produces a lot of false positives. Therefore, it is crucial to introduce an additional classifier which determines the malignancy.

Hence, the next section discusses how Malignancy Classification was performed using Very Deep Neural Networks.

#### **4.2.4. Malignancy Classification**

There are commonly used classifiers such as linear discriminant classifiers and the neural network classifiers. Neural network classifiers' deep neural network concept has worked well in many other classification problems as discussed in chapter 3.

In our methodology, it was discussed that how ResNet-50 model is more suitable and how it can be used to classify the nodules as either benign or malignant. This section will discuss on how this “very deep neural network” architecture can be used for cancer nodule classification. Training a very deep neural network from the scratch is not something which can be easily performed as it requires powerful machines with number of GPUs and a very large processing time including massive datasets for training, testing and validation purpose. The most popular benchmark dataset, ImageNet contains one million images from one thousand categories and this dataset has been processed using number of GPUs (More than one). But the issue in most of the practical problems is the dataset need to be processed through the deep neural network is considerably much lower than network's originally trained networks and this was same for the lung cancer data set. If the weights of the deep neural network were trained from scratch by starting from random initialized parameters, the comparatively small training dataset would overfit badly hence this will be the

same for our dataset as well. Most of these neural networks have millions of parameters. If we train these networks with small datasets it results in overfitting means the network will only work for examples in training data or exactly similar examples but will not generalize well. (will not work well on additional examples)

Hence, training the neural network from the scratch for a considerably small dataset is not suitable. Transfer learning is a machine learning approach where a model developed for a task is reused as the training point for a model on the second task. In transfer learning, a base network will be first trained on a base dataset. Then the learned features will be repurposed or transferred to a second target network to train on a target dataset and task. This process will tend to work well if the features are general, meaning suitable to both base and task, instead of specific to the base task.

The approach we used was, selecting a deep net (VGG 16, ResNet-50), pretrained on a large-scale dataset such as Imagenet which consists of 1.2 million images belongs to 1000 different classes. The weights trained on these networks are saved separately and they are called 'ImagNet Pretrained weights'. When started working on specific problems where a small amount of training dataset is available, one takes these pre-trained weights and continue training. Then, given our dataset (Kaggle DSB), we started with these pretrained weights when training our new tasks.

This is the approach called *fine-tuning*. There are various approaches of fine-tuning. Here below are the approaches that we tried out.

- 1) Using the initial neural network only as the *feature extractor*. Just Replace and train the last layer
- 2) Training only the last few layers

Depending on the amount of data available, the complexity of the problem which is solving, it is possible to freeze (Do not change weights during back propagation) first few layers and train only last few layers. The initial layers of convolutional neural networks just learn the general

- 3) Freeze, Pre-train and Finetune (FPT)

4) Train all the layers

Next section will discuss about the fine-tuning approaches in more detail.

### ***Using the initial neural network only as the feature extractor***

#### **Download pre-trained models**

A model often contains two parts, the .json file which specify the neural network structure, and the .param file containing the binary parameters. The name convention is ‘name-symbol.json’ and ‘name-epoch.params’, where name is the model name, and epoch is the epoch number.

We downloaded, a pre-trained ResNet 50-layer and VGG 16 models on Imagenet.

**Transfer learning is an approach which can be used to avoid training deep neural networks from its scratch.**

To get started in the GPU VM, following frameworks were needed to be installed.

- CUDA : CUDA 8.0 was downloaded from NVIDIA web (can be downloaded when the registration was done).
- NVIDIA drives associated with CUDA Toolkit 8.0
- cuDNN : cuDNN v6.0 (registration with NVIDIA was required)
- GPU card with CUDA Compute Compatibility
- MKL : Intel’s Math Kernel Library(MKL) version 11.3
- OpenCV : Can be downloaded and install from the official OpenCV website.
- Scikit-learn :Scikit-learn 0.18 is easily installed via pip
- Tensorflow : Tensorflow with GPU support. Tensorflow program typically runs faster on a GPU than on a CPU.

For image classification, the first few layers of a CNN represent low level features of the inputs, such as color blobs or texture features, and the last layers represent high level features, specific to the classification task.

As the first approach, Transfer learning with a pre-trained CNN on ImageNet was picked in this research as a featurizer to generate features from the Data Science Bowl Dataset. Once the features are computed, several classification algorithms (LightGBM, Xgboost) were used to classify the images either as cancerous or non-cancerous.

We freeze every layer prior to the output layer and simply learn a new output layer. Imagenet pretrained models have 1000 outputs from last layer, so it is possible to replace this softmax layers. For an example to build a 2 class classifier, softmax layer will have 2 output classes.

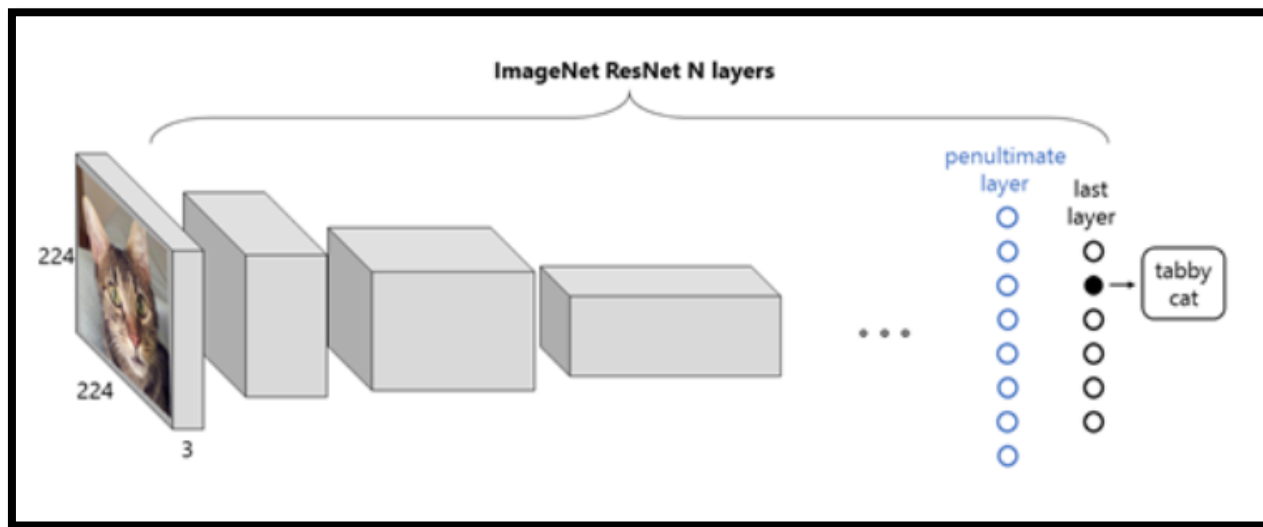


Figure 46 : Representation of a ResNet CNN with an image from Imagenet

In Figure 46 The input is an RGB image of a cat, the output is a probability vector, whose maximum corresponds to the label “tabby cat”

Figure 46 represents the typical scheme of a CNN classifying an image. The input image, in this case is a cat has a size of 224x224 with the depth of 3, related to the three color channels, red, green and blue (RGB). The image is exposed to convolutions in each internal layer, diminishing in size and growing in depth. The final layer outputs a vector of 1000 probabilities, related to 1000 classes in ImageNet. The predicted class of the network is the component with the highest probability.

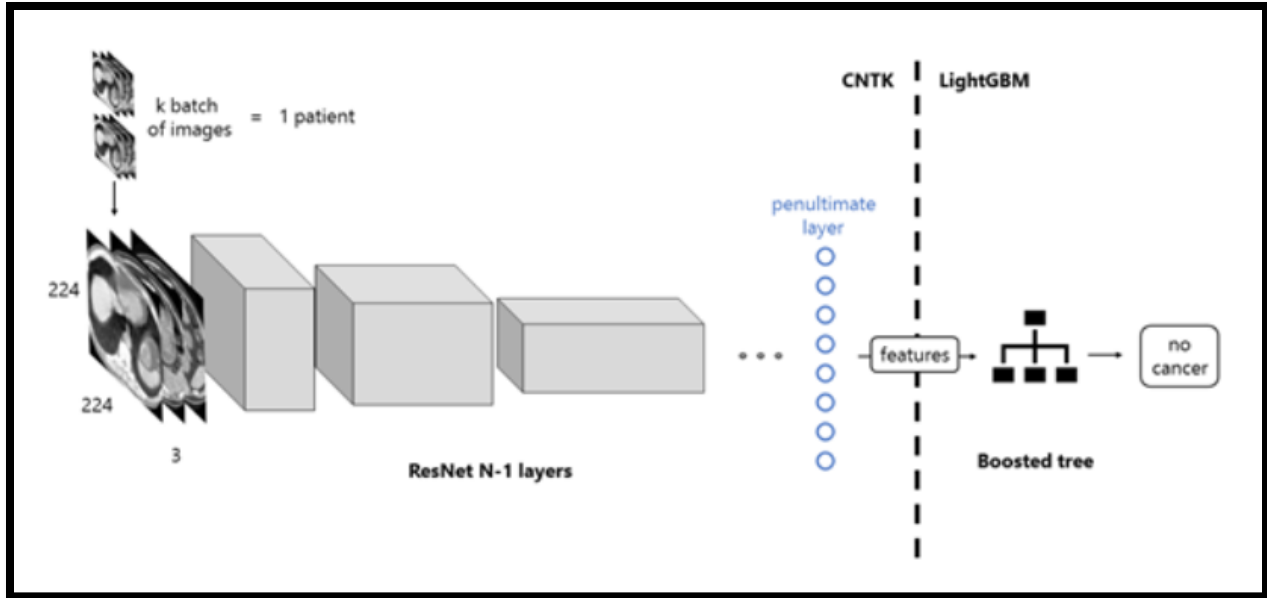


Figure 47 : Workflow of the Proposed Solution

The images of the scan are fed to the network in batches, with a forward propagation, transformed into features. Next these features are set as the input of the LightGBM boosted tree, which classifies the images as those of a patient with or without cancer.

The featurizer can be created by removing the last layer of the CNN and use the output of the penultimate layer as features. The process is shown in the Figure 47. Each patient has got an arbitrary number of scan slices. The images are cropped to 224x224 and packed in groups of three to match the format of ImageNet. They are fed to the pre-trained network in k batches of 64, and then convoluted in each internal layer, until the penultimate one. This task was performed using Keras and Tensorflow backend. The output of the features then fed to the classifier.

### ***Training only last few layers***

It is possible to freeze (Don't change weights during backpropagation) first few layers and train only last few layers. Normally, the initial layers of conventional neural networks just learn the general features like edges and very general image features. When it is going deeper, last part of the layers learns specific features such as shapes and parts of the objects. Usually we start from a very small learning rate and allows to learn the network weights.

### ***Freeze, Pre-train and Finetune (FPT)***

There are two main steps that have been involved here.

#### 1) Freeze and Pretrain:

First replace the last layer with a small mini network of two small fully connected layers. Then all the pretrained layers should be freeze and new network should be trained. Next the weights of the network (Pretrained weights) should be saved

#### 2) Finetune:

Load the pretrained weights and train the complete network with a smaller learning rate. This resulted in a good accuracy with even small datasets.

### ***Train all the layers***

This approach can be try out only if there are millions of examples in the training dataset. If such a dataset is available, starting with pre-trained weights and training the complete network can be done. But, this approach was not tried out during this research as we had a small dataset comparatively to Imagenet dataset, which had been originally used to train the base deep neural networks.

### **4.3.Using LUNA16 + LIDC dataset to detect lung cancer**

CT scans in NDSB 2017 have been annotated either as cancerous or not. No annotations have been provided for nodule locations or no information can be obtained about slice annotations. In this research we selected two main very deep neural networks named VGG19 and ResNet-50 for cancer nodule classification. The main issue we faced here was the existing VGG-19 and ResNet-50 neural networks being 2D while the CT scanned image being 3D. Since no annotations have been provided for slices in CT scans and only 2D images can be feed to 2D neural networks, we got the pixel mean of all the slices obtained after sending them through the nodule segmenter for each

scan separately and then feed the resulted 2D image into the Neural Network. This approach can be resulted in information loss as we feed averaged images to the networks.

Hence, we thought of feeding LUNA16 and LIDC datasets and do the same process 16, since it contains nodule location and we can annotate the slices of LUNA16 scanned images either as cancerous or not using that information. Therefore, in this case, we will be not feeding the averaged CT scans as 2D images to the selected 2D neural networks, instead we will feed annotated slices of LUNA16 and LIDC to train our very deep networks.

LUNA 16 dataset consists of 888 CT scans. These scans were provided as MetaImage(.mhd) images that can be accessed and downloaded through LUNA16 website. In 888 scans, a total of 1900 annotations were annotated as nodules larger than 3 mm. In this research we have considered, annotations categorized as nodules larger then 3mm as relevant lesions.

To get more annotated nodules we used remaining annotated slices of LIDC dataset as well. LUNA16 is a subset of LIDC dataset. LUNA16 is considered as a refined dataset of LIDC. In both datasets there are 2645 nodules are located. LUNA16 excluded scans with a slice thickness greater than 2.5 mm. Therefore, in total, 888 CT scans are included. The LIDC/IDRI database also contains annotations which were collected during a two-phase annotation process using 4 experienced radiologists. Each radiologist marked lesions they identified as non-nodule, nodule  $< 3$  mm, and nodules  $\geq 3$  mm. The reference standard of LUNA16 consists of all nodules  $\geq 3$  mm accepted by at least 3 out of 4 radiologists. Annotations that are not included in the reference standard (non-nodules, nodules  $< 3$  mm, and nodules annotated by only 1 or 2 radiologists) are referred as irrelevant findings. So in our case as well we have selected only the annotatger than 3mm.

Going forward we will be explaining how these annotated nodules were used for lung cancer detection of the patient. We will be building two main models.

- 1) Building a model training the U-Net network, which predicts nodule masks from slices
- 2) Evaluating the classification model to classify nodules in the nodule masks and predict whether a considered slice is either cancerous or not

As it will be explained in the next section, we will be allocating a set of noduled slices to train the model with U-Net to segment nodule masks from the slices. We will be pulling 3 consecutive slices when its center slice contains nodules larger than 3 mm. Since there are 2645 such nodules, it is possible to pull 13225 maximum slices. But some of these slices have been duplicated as some of the nodules have been in the same slice. Hence when those duplicated slices are removed, there were 11341 noduled slices available which can be used to build our model.

All the LUNA16 noduled slices (7200 – After removing duplicates from total 9500 slices) were used to train the U-Net model and to train the nodule segmenter. The rest of the slices (4141) were used to train the classification model along with other 5000 non-noduled slices.

So in the next section let's see how the 2 different models were trained and tested.

#### **4.3.1. CT scan image reading**

The LUNA16 data set consists of .mhd files. As like as the data in NDSB 2017 dataset, each scan consists of number of slices, so the 3D volumetric behavior comes to the CT scan due to this.

.mhd files can be read from SimpleITK where SimpleITK is an abstraction layer and wrapper around the Insight Segmentation and Registration Toolkit (ITK). It is available in the following programming C++, Python, R, Java, C#, Lua, Ruby, Tcl.

Annotations.csv has multiple nodules listed for cancer patients. The information in csv consists with the diameter of the nodules in millimeters and (x,y,z) coordinates in mm using a coordinate system defined in the .mhd file. Pandas DataFrame named df\_node was to keep track of the case numbers and the nodule information.

#### **4.3.2. Getting the Nodule position of the mhd files**

The nodule locations are provided in millimeters relative to a coordinate system defined by the CT scanner. The image data are with 517x517 stack of arrays with varying lengths. In order to translate the voxel position to the world coordinate system, one needs to know the real world position of the [0,0,0] voxel and the voxel spacing in mm.



To find the voxel coordinates of a nodule, given its real world position, the `GetOrigin()` and `GetSpacing()` method of the itk image objects can be used.

```
origin = np.array(itk_img.GetOrigin())    # x,y,z Origin in world coordinates (mm)

spacing = np.array(itk_img.GetSpacing())  # spacing of voxels in world coor. (mm)
```

In the annotations.csv file, the x,y,z coordinates are given from world coordinates. To convert them to voxel coordinates we need the values of origin and spacing of the image, which can be found as mentioned earlier.

Hence the voxel center was found by,

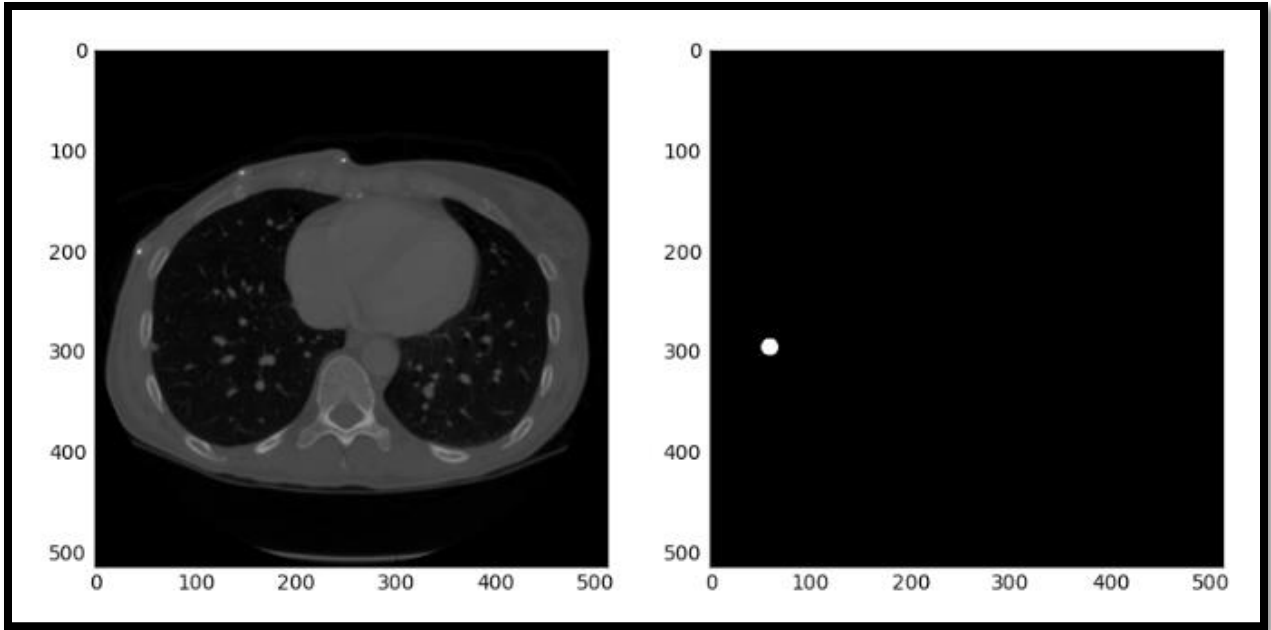
```
v_center = np rint((center-origin)/spacing)  # nodule center in voxel space (still x,y,z ordering)
```

Once the voxel center of the nodule was found, next step was to identify the slice where the center is located at, so it is possible to draw the nodule mask based on the nodule location.

The center of the nodule is located in the `v_center[2]` slice of the `img_array`. These information were passed to the mask making function, so it was possible to generate the mask and the image for `v_center[2]` slice and the slice above and below it.

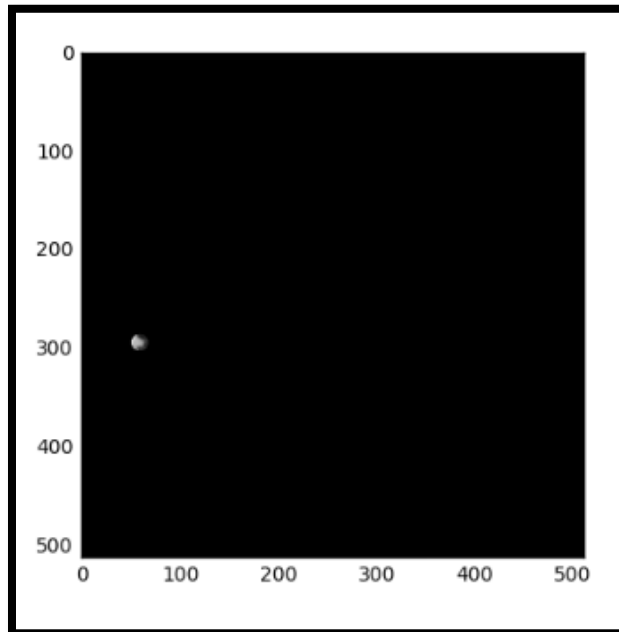
The reason for selecting the slices above and below the slice where center of the node located at is because, the nodule locations are defined in terms of spheres, and the nodules are irregularly shaped. Slices near the edges of the spheres may contain no nodule tissue. Using such slices would contaminate the training set with false positives. Therefore, to train the nodule segmentation model, only three slices were pulled.

This was how the scan slice, nodule mask and the masked slice were looked.



**Figure 48; Original CT slice on the left and its lung nodule mask on the right**

The left image is a cancer scan slice where a nodule is located at its left corner edge. Right side image is the masked nodule which was drawn using the nodule annotations.



**Figure 49: Masked Slice which highlight the nodule**

Masked slice which highlight the nodule

The nodule mask seems to be constructed properly.

Hence the mask nodule and its related slice were saved.

### 4.3.3. Isolation of the Lung ROI to narrow the nodule search

Since the nodule mask had been constructed properly, the next stage was to isolate the lungs from the saved lung slices. Several Skimage image processing modules were used for this process. The general strategy used was to threshold the regions within the image and to identify which of them are related to lung regions. Lungs have a high contrast with the surrounding tissue, so the thresholding was straightforward.

#### *Thresholding*

As the first step, the pixel values were standardized so it was possible to look at the intensity distribution

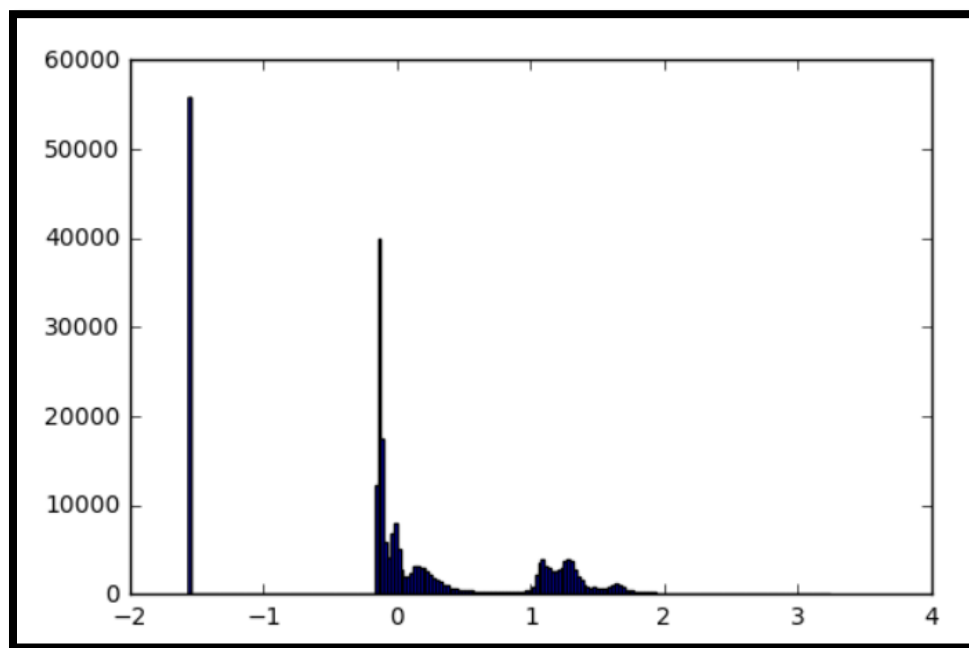


Figure 50: Pixel Histogram

The intensity distribution of a given slice

There is a peak near -1.5 and it is the black-out-of scanner part of the image. Peak around 0.0 are background and white clumps from 1.0 to 2.0 are the non-lung tissue and bones. This structure varies throughout the dataset. Following two images are two slices extracted from two different scans. When it was considered, two images have two different pixel value histograms.

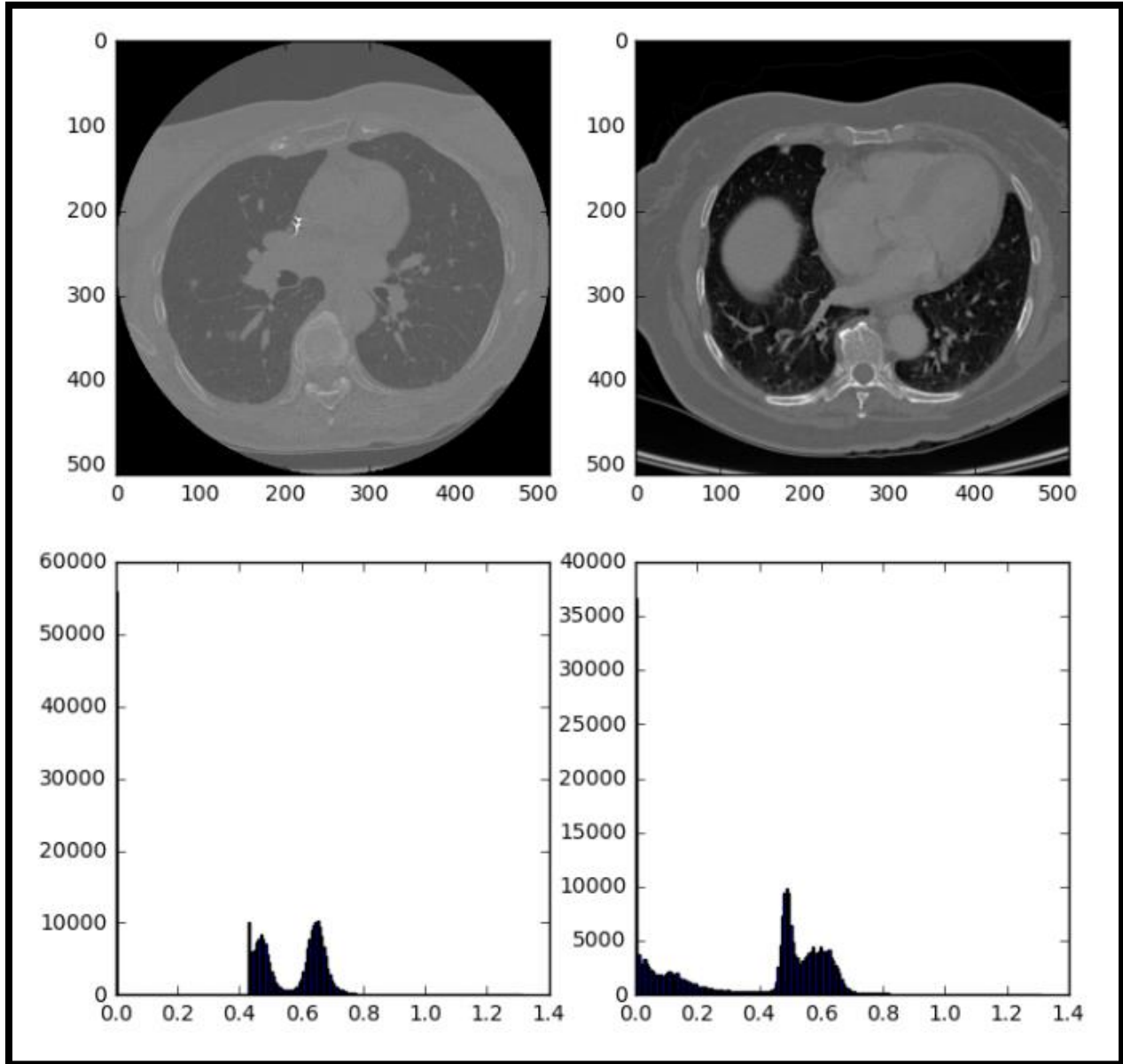


Figure 51 : Pixel value histogram of two random slices

The image in the left-hand side has the same black background around grey circular region. This black background does not represent in right hand side making very different pixel histograms.

The threshold values must be set between the lung pixel values and the denser tissue pixel values.

This was done by resetting the pixels with minimum value to the average pixel value near the center of the picture and then performed kmeans clustering with k=2. This approach worked well for both scenarios.

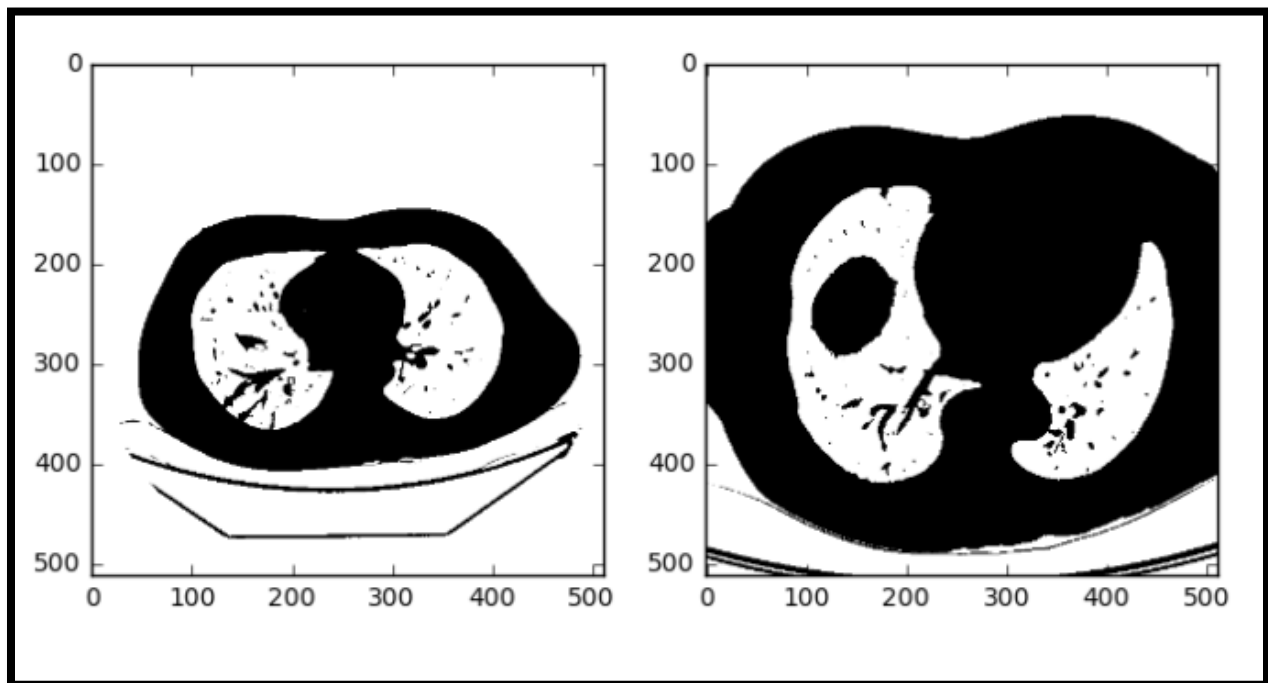


Figure 52: Segmented lung images through Thresholding

The separation of regions for both types of images and eliminates the black hole in the one on the left

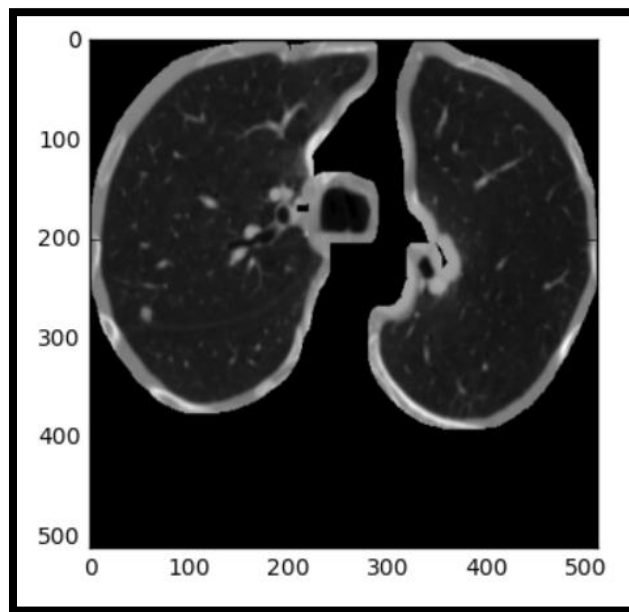
### ***Applying the ROI mask***

The next step was to apply the mask of the lung ROI to each of the images, cropping down to the bounding square of the lung ROI, and then resizing the resulting image to 512x512.

```
mask = np.load(working_path+"lungmask_0.py")
img = np.load(working_path+"images_0.py")
img = mask*img
```

Then some final pixel normalization was performed. This is because mask sends the non ROI area to 0 and the operation is not sensitive to the pixel value distribution. To fix this, we get the mean and standard deviation of the masked region and send the background to the lower end of the pixel distribution.

Final product which was possible to be sent to the training example set.



**Figure 53: Final Lung Mask**

Final Lung mask which will be sent to train the nodule segmentor.

These images and the corresponding trimmed and rescaled masks are randomized and sent to a single file that contains a numpy array of dimensions [ $\langle \text{num\_images} \rangle$ , 512, 512]. The 1 is important as the U-Net is enabled for multiple channels.

### ***Dice Coefficient Cost function for segmentation***

The network U-Net we used was implemented using the Keras framework. The loss function is the dice coefficient which compares the actual and the predicted node mask.

#### 4.3.4. Loading the Segmenter

The function call

```
model = get_unet()

model_checkpoint = ModelCheckpoint('unet.hdf5', monitor='loss', save_best_only=True)
```

This will compile and return the model tell Keras to save the model weights during checkpoints.

#### 4.3.5. Training the Segmenter

Following code will automatically load the best saved model from the current directory , and trained according to the parameters set by the line. 20% of the total dataset was allocated as training and rest was training the dataset.

```
model.fit(imgs_train, imgs_mask_train, batch_size=2, nb_epoch=20,
          verbose=1, shuffle=True, callbacks=[model_checkpoint])
```

As the next step, the predictions were made on the test set.

After about 3 hours of training the loss value approached 0.3245.

#### 4.3.6. Malignancy classification

Once the segmented model is good enough to predict lung nodules in a given slice, it seemed that the predicted nodule image contained lots of true positives as well as lots of false positives.

To remove the false positives, we used the very deep neural network classification approach. And classified the slices as either cancerous or not.

We decided to select the rest of the 4141 nodule slices and another 5000 annotated non-nodule slices for training and testing our models.

The transfer learning approaches we tried out has been explained in NDSB data [classification](#) section so the same steps were followed here as well



## 5. Evaluation

### 5.1. Chapter Overview

Previous chapter discussed about how the implementation was carried out using different datasets and techniques. In this chapter an analysis on the results that came through the implementation will be discussed. This chapter further discusses about different sections of the evaluation and the authors point of view of the implemented application.

### 5.2. Results and Discussion

In medical diagnostics, it is common to present classifier performance using sensitivity (the true positive rate) and the specificity (the true negative rate) in addition of accuracy. To assess overall classification relevance, F1-score was also computed. And the accuracy will be too presented.

There were two main deep neural networks which used for cancer nodule classification.

- 1) VGG 16 model
- 2) ResNet-50 model

#### 5.2.1. Using the initial neural network only as the *feature extractor*

XGboost was selected as the machine learning model as it was selected as one of the best classification algorithm in number of papers.

Reasons for using and finetuning XGBoost as the learning model.

- 1) Easy to use
  - a. Easy to install
  - b. Highly developed R/Python interface for users
- 2) Efficiency
  - a. Automatic parallel computation on a single machine
  - b. Can be run on a cluster
- 3) Accuracy
  - a. Good results for most data sets
- 4) Feasibility
  - a. Customized objective and evaluation

- b. Tunable parameters

## Parameter Introduction

XGBoost has plenty of parameters. We can group them into

- 1) General parameter
  - a. Number of threads
- 2) Booster parameters
  - a. Step size
  - b. Regularization
- 3) Task parameters
  - a. Objective
  - b. Evaluation metric

## General Parameters

- Nthread- Number of parallel threads
  
- Booster
  - Gbtree – tree based model
  - Gblinear – Linear function
  
- Eta
  - Step size shrinkage used in update to prevent overfitting
  - Range in [0,1], default 0.3
  
- Gamma
  - Minimum loss reduction required to make a split
  - Range [0, ] , default 0
  
- Max\_depth
  - Maximum depth of a tree
  - Range [1, ] , default 1
  
- Min\_child\_weight
  - Minimum sum of instance weight needed in a child
  - Range [0, ~], default 6*
  
- Max\_delta\_step

Maximum delta step we allow each tree's weight estimation  
*Range [0, ~], default 0*

- Subsample

Subsample ratio of the training instance  
*Range [0, ~], default 0*

- Colsample\_bytree

Subsample ratio of columns when constructing each tree  
*Range [0, ~], default 0*

### Parameters for Linear Booster

- Lambda

L2 regularization term on weights  
Default 0

- Alpha

L1 regularization term on weights  
Default 0

- Lambda\_bias

L2 regularization term on bias  
Default 0

### Parameter Tuning

It is nearly impossible to give a set of universal optimal parameters, or a global algorithm achieving it. The key points of parameter tuning are,

- Control overfitting
- Deal with imbalanced data

- Trust the cross validation

Different parameters are used for addressing different issues of the model.

- Controlling the model complexity

Max\_depth, min\_child\_weight, gamma

- Robust to noise

Subsample, colsample\_bytree

- When data is imbalanced among classes

Balance the positive and negative weights, by scal\_pos\_weight parameter

Using “auc” as the evaluation metric

- Care about predicting the right probability

Cannot re-balance the dataset

Set parameter ‘max\_delta\_step’ to a finite number, will help convergence

To select the ideal parameters, result from xgb.cv was used

- Trust the score for the test
- Use early.stop.round to detect continuously being worse on test set
- If overfitting observed, reduce stepsize eta and increase nround at the same time

Following performance measures were calculated to measure how well the model had been trained

Precision =  $TP / (TP + FP)$

Accuracy =  $(TP + TN) / (TP + TN + FP + FN)$

Sensitivity or Recall =  $TP / (TP + FN)$

Specificity =  $TN / (TN + FP)$

Error rate =  $(FP + FN) / (TP + TN + FP + FN)$

F1-Score =  $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$

### 5.3.Evaluation Results for NDSB 2017

#### Using ResNet neural network for feature extraction

I've added these terms to the confusion matrix, and added the row and column totals:

n=198	Predicted NO	Predicted YES	
Actual NO	TN – 74	FP – 67	141
Actual YES	FN – 17	TP – 40	57
	91	107	198

Table 1 : Confusion Matrix

Diagnosis Test	Result
Precision	0.373831775
Error rate	0.424242424
Accuracy	0.575757575
Sensitivity or Recall	0.701754385
Specificity	0.524822695
F1-Score	0.487804877

Table 2 : Diagnosis Test Results

#### Best Parameters

- max\_depth=5,
- n\_estimators=1000,
- min\_child\_weight=1,
- learning\_rate=0.001,
- subsample=0.80,
- colsample\_bytree=0.80,
- seed=27,
- gamma=0,
- nthread=4,
- scale\_pos\_weight=1

Best threshold value-0.35

## Using VGG19 neural network for feature extraction

I've added these terms to the confusion matrix, and added the row and column totals:

<b>n=198</b>	<b>Predicted NO</b>	<b>Predicted YES</b>	
<b>Actual NO</b>	<b>TN – 70</b>	<b>FP – 71</b>	<b>141</b>
<b>Actual YES</b>	<b>FN – 21</b>	<b>TP – 36</b>	<b>57</b>
	<b>91</b>	<b>107</b>	<b>198</b>

Table 3 : Confusion Matrix

<b>Diagnosis Test</b>	<b>Result</b>
<b>Precision</b>	0.336448598
<b>Error rate</b>	0.464646465
<b>Accuracy</b>	0.535353535
<b>Sensitivity or Recall</b>	0.631578947
<b>Specificity</b>	0.496453901
<b>F1-Score</b>	0.439024390

Table 4 : Diagnosis Test Results

### Best Parameters

- max\_depth=8,
- n\_estimators=1000,
- min\_child\_weight=1,
- learning\_rate=0.001,
- subsample=0.80,
- colsample\_bytree=0.80,
- seed=27,
- gamma=0,
- nthread=5,
- scale\_pos\_weight=1

Best threshold value-0.35

### 5.3.1. Training only last few layers

This approach freezes the first few layers and trained only last few last layers as discussed during previous sections. This section discusses on how fine-tuning the VGG19 network would help in cancer detection.

#### VGG19 neural network for training last few layers

The VGG neural network consists of 19 layers as displayed below. Freezing different top-level layers and training the dataset from the rest of the neural network has been done during this section.

```
1 input_1
2 block1_conv1
3 block1_conv2
4 block1_pool
5 block2_conv1
6 block2_conv2
7 block2_pool
8 block3_conv1
9 block3_conv2
10 block3_conv3
11 block3_pool
12 block4_conv1
13 block4_conv2
14 block4_conv3
15 block4_pool
16 block5_conv1
17 block5_conv2
18 block5_conv3
19 block5_pool
```

Figure 54 : The layers of VGG neural network

Best parameters used to achieve the best results have been mentioned by the following section

Freezing the weights of the first top 15 layers and training the network on last four layers gave the best results.

Number_of classes	-	2
Batch_size	-	64
Number of epochs	-	10

Model compilation with 'SGD' optimizer with learning rate= 0.0001, momentum =0.9, loss='categorical\_crossentropy' and metrics = 'accuracy'

	Predicted		
	NO	YES	
Actual NO	TN – 85	FP – 56	141
Actual YES	FN – 20	TP – 37	57
	105	93	

**Table 3 : Confusion Matrix**

Diagnosis Test	Result
<b>Precision</b>	0.397849462
<b>Error rate</b>	0.383838384
<b>Accuracy</b>	0.616161616
<b>Sensitivity or Recall</b>	0.602836879
<b>Specificity</b>	0.649122807
<b>F1-Score</b>	0.479018889

**Table 4 : Diagnosis Test Results**

Threshold value – 0.24



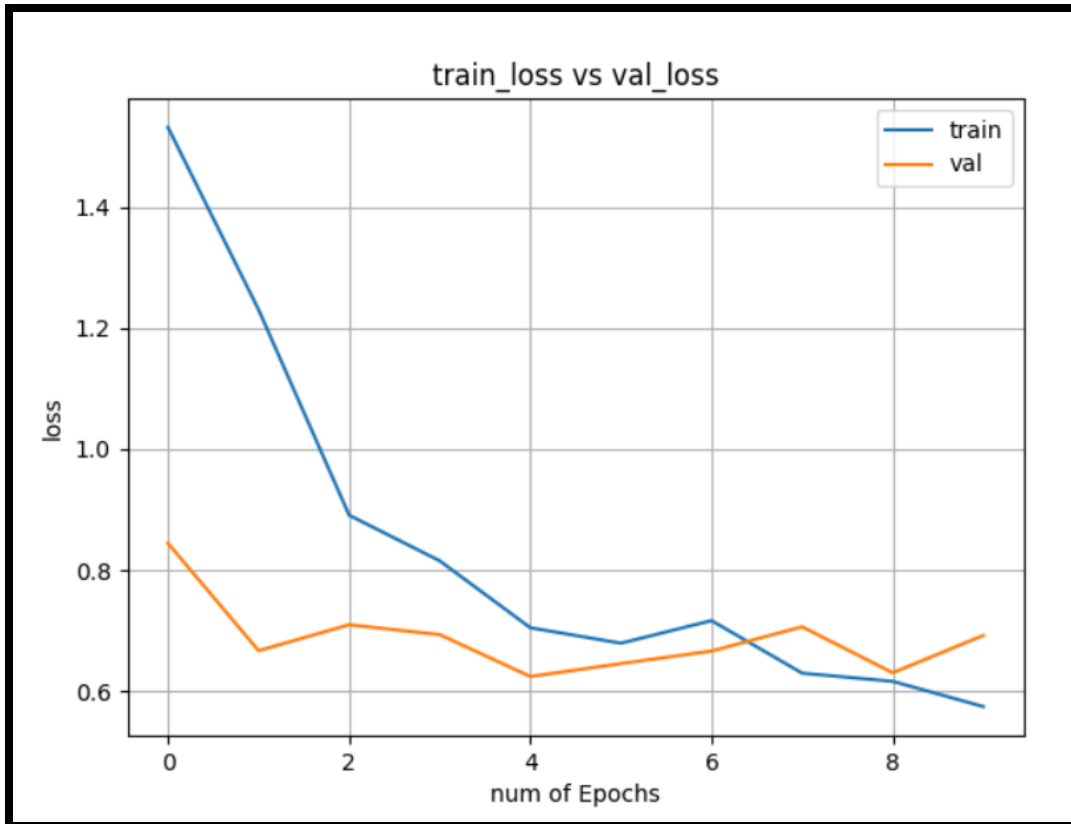


Figure 55 : Training loss vs Validation loss for the 15 layers freezed VGG model

### 5.3.2. Freeze, Pre-train and Finetune (FPT)

As mentioned above, this approach mainly involved in two steps.

#### Freezing and Pretraining:

First replace the last layer with a small mini network of two small fully connected layers. Then all the pretrained layers should be freeze and new network should be trained. Next the weights of the network (Pretrained weights) should be saved

Following steps were involved during this section

- 1) Specified the path to training and testing data along with number of classes (2 classes) and the image size (240\*240).
- 2) Created the Resnet-50 and VGG19 network graph (without top) and loaded imageNet pre-trained weights.
- 3) Added the top as per number of classes in our dataset (2 classes).

- 4) Replaced the last layer with a small mini network of two small fully connected layers. Dropout layer with value 0.2 was used to discard 20% of the weights and used softmax activation function for the last layer of the mini network.
- 5) Next freeze all the layers in Resnet-50 and VGG19 network separately.
- 6) Specified the complete model input and output, optimizer and loss and compile the new model.

```
Loss="categorical_crossentropy"
optimizer=optimizers.Adam()
metrics=["accuracy"]
```

- 7) Model training was done initially for 10 epochs, verbose=1 and saved the weights matrixes.

**Finetuning:**

- 1) Loaded the pretrained weights of two networks separately and trained the complete network with a smaller learning rate.  
Model compiled with following values for two networks.

**FPT results for Resnet-50 model**

```
loss="categorical_crossentropy"
optimizer=optimizers.SGD(lr=0.001, momentum=0.8),
metrics=["accuracy"]
```

- 2) This resulted in a good accuracy with 12 epochs.

	Predicted		
	NO	YES	
Actual NO	TN – 92	FP – 49	141
Actual YES	FN – 15	TP – 42	57
	107	91	198

**Table 5 : Confusion Matrix**

Diagnosis Test	Result
Precision	0.461538462
Error rate	0.343434343
Accuracy	0.676767677
Sensitivity or Recall	0.736842105
Specificity	0.652482270
F1-Score	0.567567568

Table 6 : Diagnosis Test Result

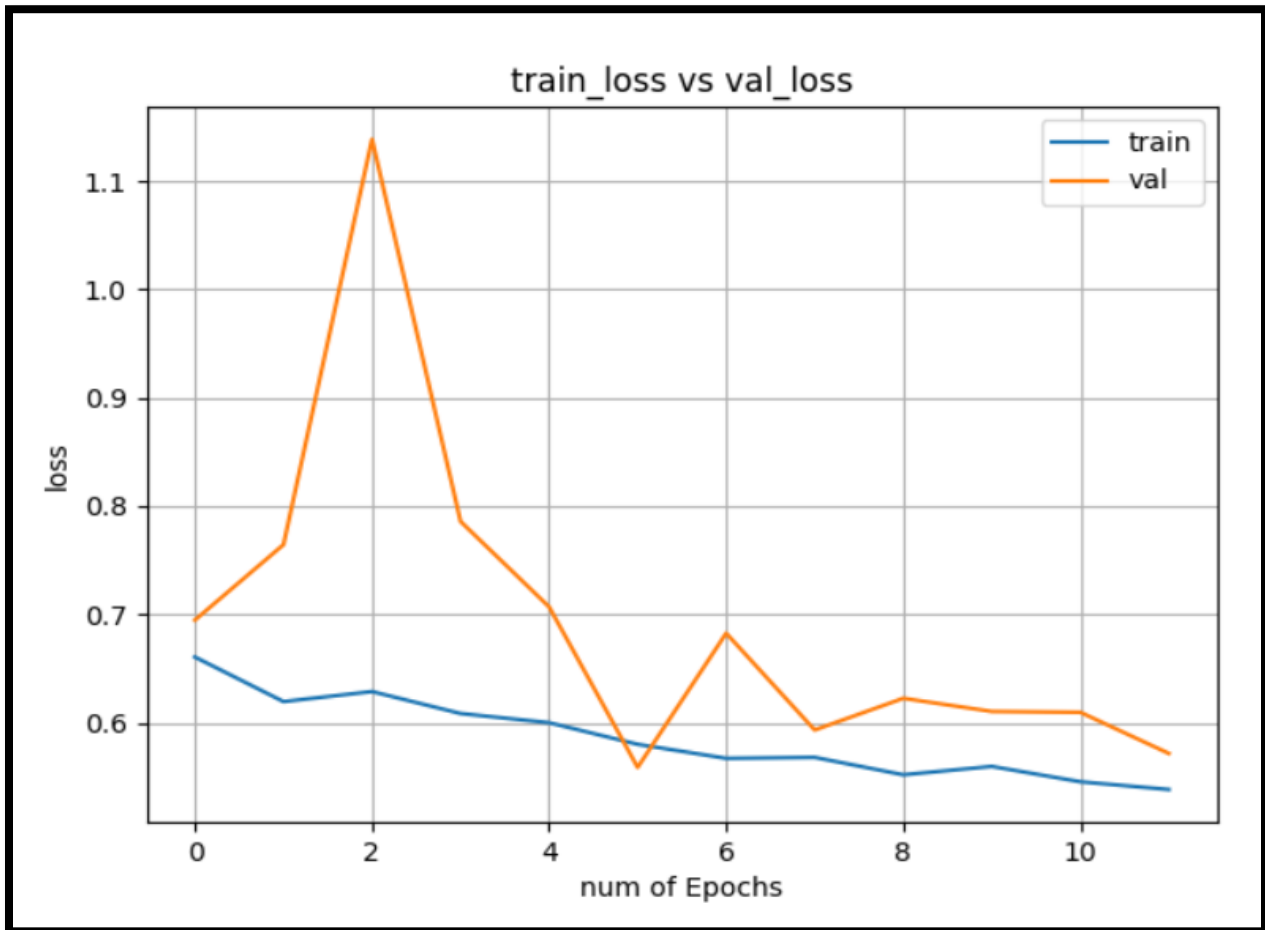


Figure 56 : Training loss vs Validation loss for FPT approach, Resnet-50 model

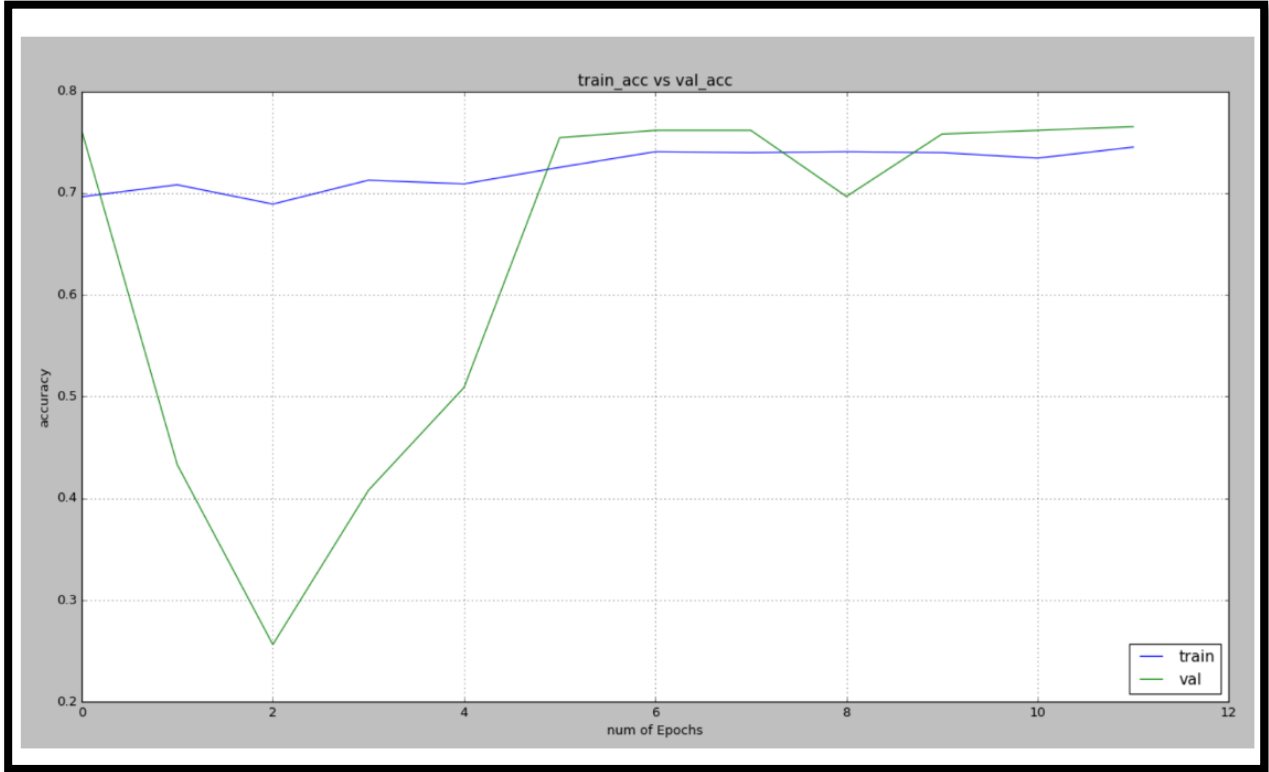


Figure 57 : Training accuracy vs Validation accuracy for FPT approach, Resnet-50 model

### FPT results for VGG19 model

```

loss="categorical_crossentropy"
optimizer=optimizers.SGD(lr=0.0001, momentum=0.9),
metrics=["accuracy"]

```

This resulted in a good accuracy with 30 epochs, with even small datasets.

	Predicted		
	NO	YES	
Actual NO	TN – 89	FP – 52	141
Actual YES	FN – 16	TP – 41	57
	105	93	

Table 7 : Confusion Matrix

Diagnosis Test	Result
Precision	0.440860215
Error rate	0.343434343
Accuracy	0.656565656
Sensitivity or Recall	0.719298245
Specificity	0.631205673
F1-Score	0.546666666

Table 8 : Diagnosis Test Result

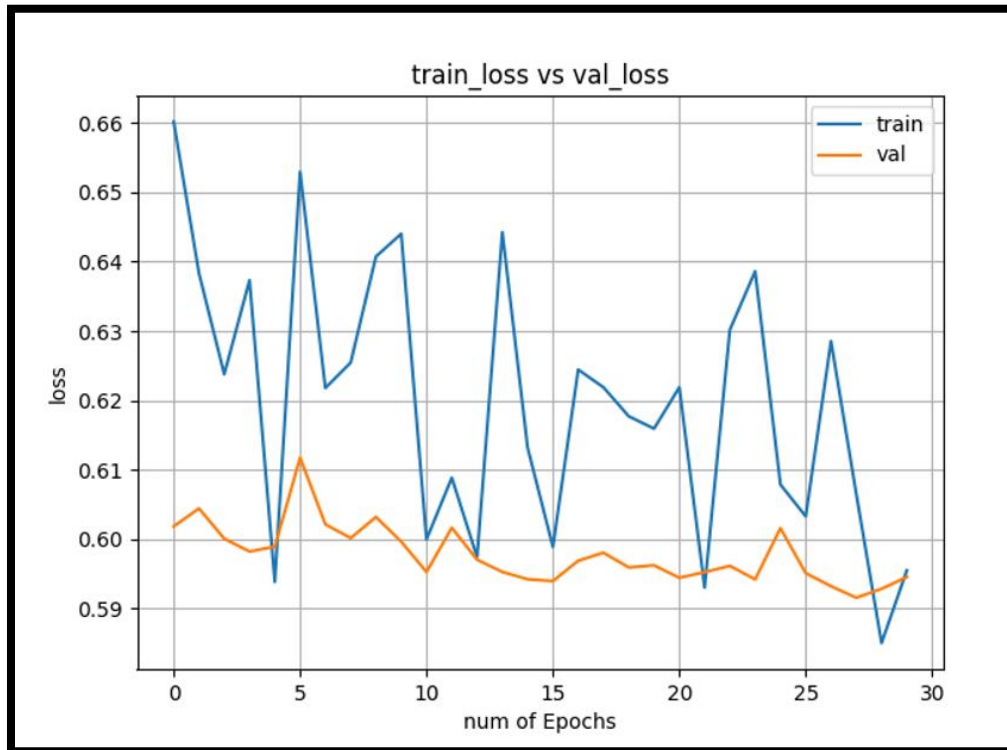


Figure 58 : Training loss vs Validation loss for FPT approach, VGG19 model

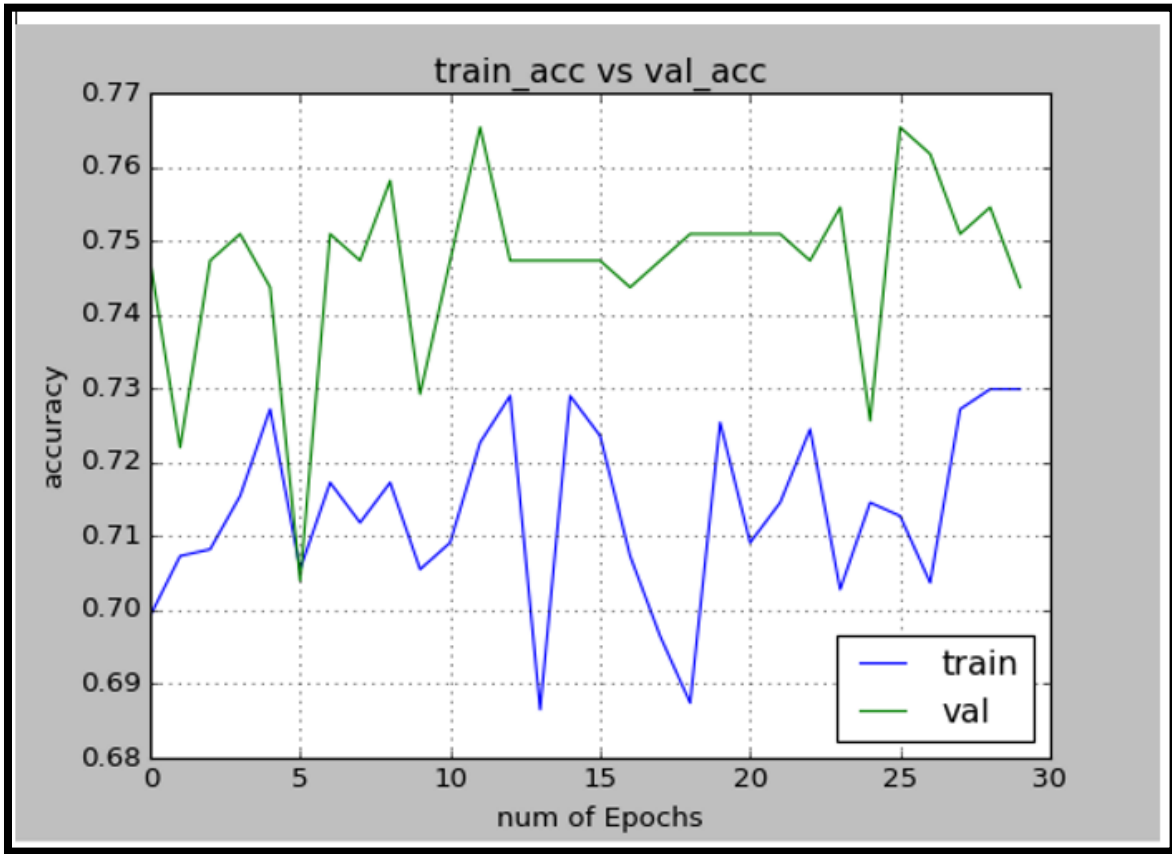


Figure 59 : Training accuracy vs Validation accuracy for FPT approach, VGG19 model

Our evaluation results have been compared with other cancer nodule detection approaches in the next section.

## 5.4.Evaluation Results for LUNA16 dataset

### Using ResNet neural network for feature extraction

I've added these terms to the confusion matrix, and added the row and column totals:

<b>n=198</b>	<b>Predicted NO</b>	<b>Predicted YES</b>	
<b>Actual NO</b>	TN – 351	FP – 233	584
<b>Actual YES</b>	FN –155	TP – 261	416
	506	494	1000

Table 9 : Confusion Matrix

<b>Diagnosis Test</b>	<b>Result</b>
<b>Precision</b>	0.52834008
<b>Error rate</b>	0.388
<b>Accuracy</b>	0.612
<b>Sensitivity or Recall</b>	0.62540385
<b>Specificity</b>	0.60102740
<b>F1-Score</b>	0.57568966

Table 10 : Diagnosis Test Results

### Best Parameters

- max\_depth=6,
- n\_estimators=1000,
- min\_child\_weight=1,
- learning\_rate=0.001,
- subsample=0.80,
- colsample\_bytree=0.80,
- seed=25,
- gamma=0,
- nthread=4,
- scale\_pos\_weight=1

Best threshold value=0.42

## VGG19 neural network for training last few layers

Freezing the weights of the first top 14 layers and training the network on last five layers gave the best results.

Number\_of classes - 2  
Batch\_size - 64  
Number of epochs - 15

Model compilation with 'SGD' optimizer with learning rate= 0.0001, momentum =0.9, loss='categorical\_crossentropy' and metrics = 'accuracy'

	Predicted		
	NO	YES	
Actual NO	TN – 402	FP – 182	584
Actual YES	FN –113	TP – 303	416
	515	485	1000

Table 11 : Confusion Matrix

Diagnosis Test	Result
Precision	0.62474227
Error rate	0.295
Accuracy	0.705
Sensitivity or Recall	0.72836538
Specificity	0.68835616
F1-Score	0.67051852

Table 12 : Diagnosis Test Results

Threshold value – 0.42



## FPT results for Resnet-50 model

Following steps were involved during this section

- 1) Specified the path to training and testing data along with number of classes (2 classes) and the image size (240\*240).
- 2) Created the Resnet-50 network graph (without top) and loaded imageNet pre-trained weights.
- 3) Added the top as per number of classes in our dataset (2 classes).
- 4) Replaced the last layer with a small mini network of two small fully connected layers. Dropout layer with value 0.3 was used to discard 30% of the weights and used softmax activation function for the last layer of the mini network.
- 5) Next freeze all the layers in Resnet-50 network separately.
- 6) Specified the complete model input and output, optimizer and loss and compile the new model.

```
Loss="categorical_crossentropy"  
optimizer=optimizers.Adam()  
metrics=["accuracy"]
```

- 7) Model training was done initially for 10 epochs, verbose=1 and saved the weights matrixes.

### Finetuning:

- 8) Loaded the pretrained weights of two networks separately and trained the complete network with a smaller learning rate.  
Model compiled with following values for two networks.
- 9) This resulted in a good accuracy with 60 epochs.

	Predicted		
	NO	YES	
Actual NO	TN – 471	FP – 113	584
Actual YES	FN –84	TP – 332	416
	555	445	1000

Table 13 : Confusion Matrix

Diagnosis Test	Result
Precision	0.74606742
Error rate	0.197
Accuracy	0.803
Sensitivity or Recall	0.79807692
Specificity	0.80650685
F1-Score	0.77419355

Threshold value -0.45

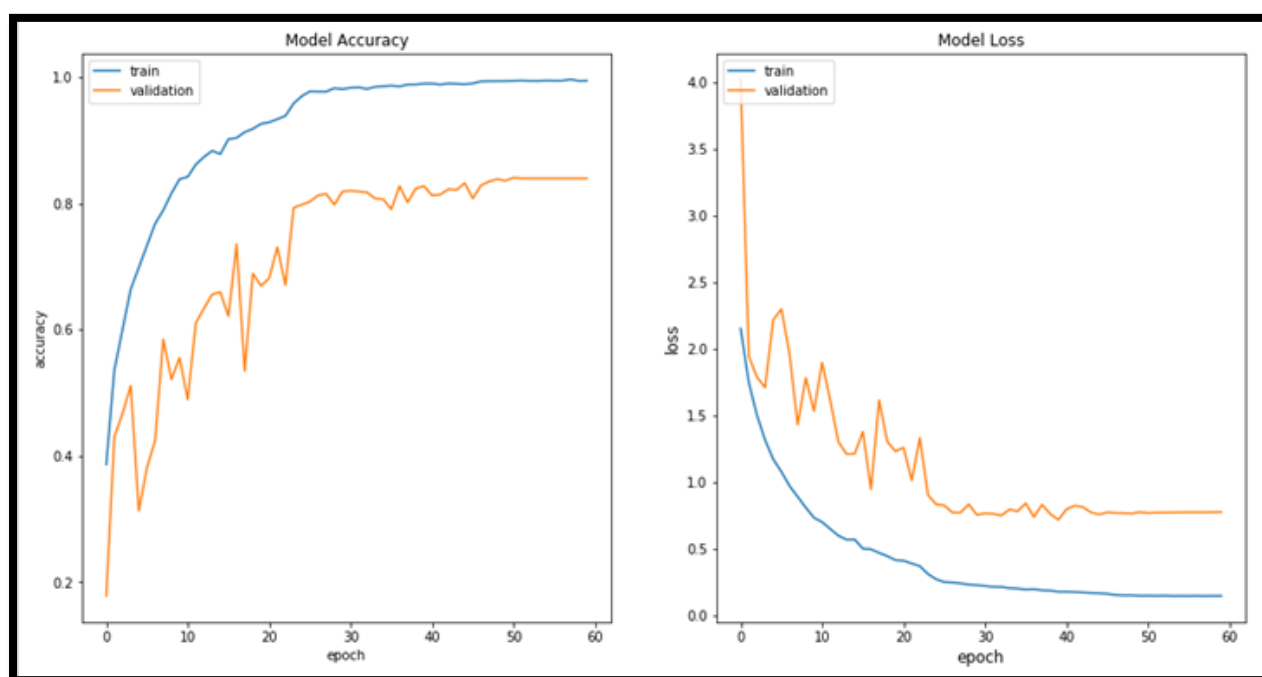


Figure 60 : Model Accuracy and loss for FPT approach

### FPT results for VGG-19 model

Following steps were involved during this section

- 1) Specified the path to training and testing data along with number of classes (2 classes) and the image size (240\*240).
- 2) Created the VGG-19 network graph (without top) and loaded imageNet pre-trained weights.

- 3) Added the top as per number of classes in our dataset (2 classes).
- 4) Replaced the last layer with a small mini network of two small fully connected layers. Dropout layer with value 0.3 was used to discard 30% of the weights and used softmax activation function for the last layer of the mini network.
- 5) Next freeze all the layers in VGG19 network separately.
- 6) Specified the complete model input and output, optimizer and loss and compile the new model.

```
Loss="categorical_crossentropy"  
optimizer=optimizers.Adam()  
metrics=["accuracy"]
```

- 7) Model training was done initially for 15 epochs, verbose=1 and saved the weights matrixes.

### **Finetuning:**

- 8) Loaded the pretrained weights of two networks separately and trained the complete network with a smaller learning rate.

Model compiled with following values for two networks.

- 9) This resulted in a good accuracy with 30 epochs.

Following is how the accuracy varied in test and the train data set

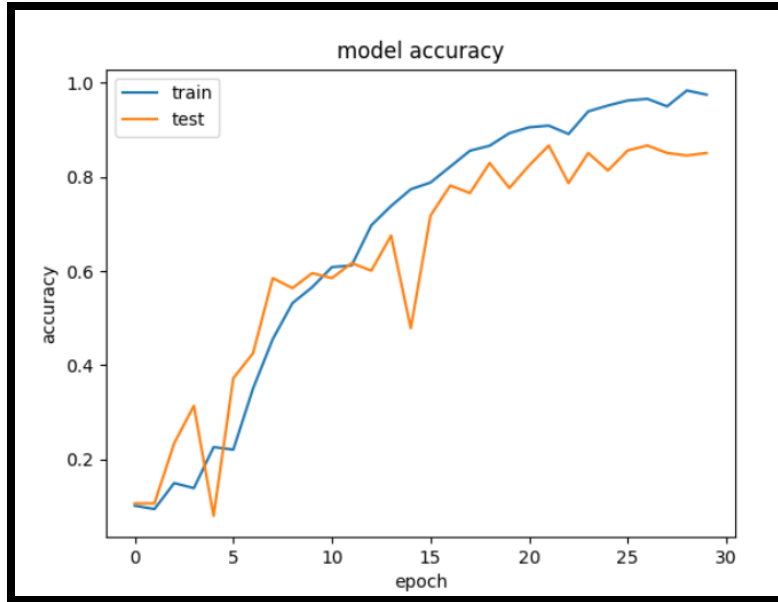


Figure 61 : Model Accuracy for VGG19 FPT approach

**Confusion matrix VGG19FPT approach**

	Predicted		
	NO	YES	
Actual NO	TN – 402	FP – 182	584
Actual YES	FN – 103	TP – 313	416
	505	495	1000

Table 14 : Confusion Matrix

## Performance Measurements

Diagnosis Test	Result
Precision	0.63232323
Error rate	0.285
Accuracy	0.715
Sensitivity or Recall	0.75240385
Specificity	0.68835616
F1-Score	0.68478261

Table 15: Diagnosis Results

Threshold value-0.41

## 5.5. Performance Evaluation with Non-3D CNN -NDSB 2017

Title	Neural Network used	Dataset	Technique	Accuracy	Sensitivity	Specificity	F1Score
Deep Learning for Categorization of Lung Cancer CT Images (Allison M Rossetto and Wenjin Zhou, 2017) [50]	DCNN (Deep Convolutional NN)	Kaggle Data Science bowl 2017	Matlab Neural Network Toolbox  With Nvidia GTX 1080GPU	0.72	0.49	-	0.59
A Deep Convolutional Neural Network for Lung Cancer Diagnostic (Mehdi Fatan et al, April 2018) [51]	DCNN	Kaggle Data Science Bowl 2017	Four Convolution layers which follow two maxpooling layers, a full-body convolution layer with 2 softmax units	-	0.87	0.99	0.95
<b>Ours</b>	<b>VDCNN (Very Deep Convolutional NN)</b>	<b>Kaggle Data Science Bowl 2017</b>	<b>Resnet-50</b>  <b>With FPT approach</b>	<b>0.68</b>	<b>0.74</b>	<b>0.65</b>	<b>0.56</b>
	<b>VDCNN (Very Deep Convolutional NN)</b>	<b>Kaggle Data Science Bowl 2017</b>	<b>VGG19</b>  <b>With FPT approach</b>	<b>0.65</b>	<b>0.71</b>	<b>0.63</b>	<b>0.54</b>

Table 16 : Performance Evaluation with Non-3D CNN

Rossto et al. [50] in ‘Deep Learning for Categorization of Lung Cancer CT Images’ has come up with accuracy of 72% and f1-Score 0.59. Then in 2018 Mehedi Fatan et al. [51] has come up with a solution using deep convolutional neural network with high sensitivity, specificity and F1score values such as 0.87, 0.991 and 0.95 respectively. Meanwhile our solution which uses very deep convolutional neural network came up with accuracy 0.68, sensitivity 0.74 and specificity 0.65. Our results are comparable with paper “Deep Learning for Categorization of Lung Cancer CT Images” [50].

## 5.6. Performance Evaluation with 3D CNN and Other types -NDSB 2017

Title	Neural Network used	Dataset	Technique	Accuracy	Sensitivity	Specificity	F1Score
Deep Learning for Lung Cancer Detection: Tackling the Kaggle Data Science Bowl 2017 Challenge [52]	Original ResNet	Kaggles Data Science Bowl 2017 + Luna dataset	Direct ResNet-101 model		0.169	0.858	0.181
	Modified ResNet (researcher's architecture)	Kaggles Data Science Bowl 2017 + Luna dataset	Modified ResNet-101 (3D operations, replace the global average pooling operation with a 1x1x1 convolution, and narrowed the receptive field of each node to 16x16x16 voxels)	-	0.538	0.648	0.33
Lung Cancer Detection and Classification	3D CNN	Kaggle Data Science Bowl 2017	3D Convolutional Neural Network, weighted softmax cross	0.866	0.853	0.881	0.8660

with 3D Convolutional Neural Network (3D-CNN) [54]			entropy loss and Adam optimizer				
<b>Ours</b>	<b>VDCNN (Very Deep Convolutional NN)</b>	<b>Kaggle Data Science Bowl 2017</b>	<b>Resnet-50 With FPT approach</b>	<b>0.68</b>	<b>0.74</b>	<b>0.65</b>	<b>0.56</b>
	<b>VDCNN (Very Deep Convolutional NN)</b>	<b>Kaggle Data Science Bowl 2017</b>	<b>VGG19 With FPT approach</b>	<b>0.65</b>	<b>0.71</b>	<b>0.63</b>	<b>0.54</b>

**Table 17 : Performance Evaluation with 3D CNN and Other types**

Above grid focuses on results that were obtained from specific researches that had used Kaggle data science bowl 2017 dataset. And then its compared with the results we obtained. According to the figures mentioned in the above table our records in sensitivity and specificity is significantly better than some of the outputs. Our F1Score value is at 0.56 which is better than some of the mentioned F1score values. Both our VGG and ResNet models have outperformed results presented in paper, “Deep Learning for Lung Cancer Detection:Tackling the Kaggle Data Science Bowl 2017 Challenge” [52]

So, when a comparison is carried out with CNN 3D and Non-3D CNN types, our system had a good sensitivity and specificity values relatively.



## 5.7. Performance Evaluation Results for Luna16

Title	Neural Network used	Dataset	Technique	Accuracy	Sensitivity	Specificity	F1Score
Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images [57]	CNN	LIDC-IDRI	Two convolution layers consist of 32 filters, and the kernel size is 5. The pooling layer has a kernel size of 2	84.15%	83.96%	84.32%	
	DNN	LIDC-IDRI	Consists of a fully connected layer. The second layer is a fully connected layer of $512 \times 1$ . The third layer is a fully connected layer of $256 \times 1$ . After the third layer, there will be a dropout layer, with a parameter of 0.6, in which the unit will be hidden in 40%. The fourth layer is a fully connected layer of $64 \times 1$ , whose activation function is set to ReLU	82.37%	80.66%	83.9%	
	SAE	LIDC-IDRI	Made up of a fully connected layer. The neurons of the autoencoder's input and output are the same; the autoencoder is equivalent to the following function:  $Hw, b \cdot x = x,$  where $w$ and $b$ are the weight and crankiness, respectively, in the	82.59%	83.96%	81.35%	

			neural network operation and $x$ is the input parameter				
Classification of malignancy of lung nodules in CT images using Convolutional Neural Network [58]	Architecture 1	LIDC-IDRI	Convolutional layer with 32 filters and ReLU activation, another convolutional layer with 16 filters and ReLU activation, a pooling layer with kernel size of 2 and a dropout layer, followed by a fully-connected layer with 16 neurons, a ReLU activation and a dropout layer, on top, a fully-connected layer with 2 neurons and a softmax function	80.2%	75.2%	83.1%	
	Architecture 2	LIDC-IDRI	convolutional layer with 16 filters and ReLU activation, a pooling layer with kernel size of 2, another convolutional layer with 32 filters and ReLU activation, a pooling layer with kernel size of 2 and a dropout layer, followed by fully-connected layer with 16 neurons, a ReLU activation and a dropout layer, on top, a fully-connected layer with 2 neurons and a softmax function	80.2%	82.9%	78.6%	
	Architecture 3	LIDC-IDRI	convolutional layer with 16 filters and ReLU activation, a pooling layer with kernel size of 2, another convolutional layer with 32 and ReLU activation, a convolution layer with 8 filters	82.3%	79.4%	83.8%	

			and ReLU activation, followed by fully-connected layer with 16 neurons, a ReLU activation and a dropout layer, another fully-connected layer with 32 neurons and ReLU activation, on top, a fully connected layer with 2 neurons and a softmax function.				
Computer-aided classification of lung nodules on computed tomography images via deep learning technique [59]	CNN	LIDC	The DBN(deep belief network) and CNN deep learning frameworks	NA	73.30%	78.70%	
<b>Ours</b>	<b>CNN</b>	<b>LIDC</b>	<b>Resnet-50 With FPT approach</b>	<b>80.3%</b>	<b>79.8%</b>	<b>80.6%</b>	<b>0.77</b>

**Table 18 : Performance Evaluation Results for Luna16**

## Comparison of VGG 19 and ResNet-50 with their different Transfer learning approaches – For NDSB 2017

### VGG19 Model

Transfer Learning method	Accuracy	Sensitivity	Specificity	F1-Score
Training Last few layers	0.705	0.72836538	0.68835616	0.67051852
FPT approach	0.656565656	0.719298245	0.631205673	0.546666666

Table 19 : Comparison of VGG19 model with their different transfer learning approaches

### ResNet-50 Model

Transfer Learning method	Accuracy	Sensitivity	Specificity	F1-Score
Extracting Features	<b>0.575757575</b>	<b>0.701754385</b>	<b>0.524822695</b>	<b>0.487804877</b>
FPT approach	<b>0.676767677</b>	<b>0.736842105</b>	<b>0.652482270</b>	<b>0.567567568</b>

Table 20 : Comparison of ResNet-50 model with their different transfer learning approaches

When compared the results obtained by Two models, comparatively the best output has been obtained by the Resnet-50 model outperforming the results of VGG19 model. Among the three transfer learning approaches, the FPT method has performed well in both networks , than the mere feature extraction and training Last few layers approach. The best F1-Score which was used as the comparison to evaluate the network performance, has been obtained by the ResNet model with F1-Score 0.56766, outperforming the F1-Score of VGG19 network which was obtained as 0.5467.

## Comparison of VGG 19 and ResNet-50 with their different Transfer learning approaches – LUNA16 + LIDC

### VGG19 Model

Transfer Learning method	Accuracy	Sensitivity	Specificity	F1-Score
Training Last few layers	0.705	0.72836538	0.68835616	0.67051852
FPT approach	0.715	0.75240385	0.68835616	0.68478261666

Table 21 : Comparison of VGG19 model with their different transfer learning approaches

### ResNet-50 Model

Transfer Learning method	Accuracy	Sensitivity	Specificity	F1-Score
Feature Extraction Process	0.612	0.62540385	0.60102740	0.57568966
FPT approach	0.803	0.79807692	0.80650685	0.77419355

Table 22 : Comparison of ResNet-50 model with their different transfer learning approaches

When compared the results obtained by Two models, comparatively the best output has been obtained by the Resnet-50 model outperforming the results of VGG19 model for LUNA16 dataset as well. Among the three transfer learning approaches, the FPT method has performed well in both networks than feature extraction and training Last few layers approach. The best F1-Score which was used as the comparison to evaluate the network performance, has been obtained by the ResNet model with F1-Score 0.77419355, outperforming the F1-Score of VGG19 network which was obtained as 0.68478261666.

## 6. Conclusion

This research was performed to detect lung cancer in CT scanned images using two very deep neural networks, VGG19 and ResNet-50. Each very deep neural network model was trained using three main Transfer learning approaches,

- 1) Using the initial neural network only as the feature extractor
- 2) Training only last few layers
- 3) Freeze, Pre-train and Fine-tuning (FPT)

Prior for the classification, the image preprocessing was done to segment the images using different techniques such as Threshold, Watershed. U-Net, 2D neural network was used to identify the nodule masks (Nodule Classification) of the patients. Then the segmented nodule masks were used classified either as cancerous or not using two famous very deep neural networks named Resnet-50 and VGG19.

The above three Transfer Learning approaches in VGG19 and ResNet were used for CT scan classification either as cancerous or not and the results were assessed and compared using accuracy, sensitivity and specificity and F1-Score.

In VGG19 neural network applied for NDSB dataset, the best results were given by the third Transfer Learning approach “Freeze, Pre-train and Fine-tuning” (FPT), with accuracy-65.657%, sensitivity-71.929% , specificity-63.121% with F1-Score-0.5466666666.

In ReNet neural network for NDSB the best results were given by the third Transfer Learning approach, which is the “Freeze, Pre-train and Fine-tuning” (FPT) method and the accuracy, sensitivity, specificity and the F1-Score-were 68%, 73%, 65% and 0.567567568 respectively. This outperformed the VGG19 neural Network.

Using the dataset Luna16 and LIDC, we extracted 2D noduled slices and trained a U-Net network to segment nodules from a given slice. Then the segmented outputs were feed to two Very deep neural networks, ResNet-50 and VGG19 to perform nodule classifications. The best results were again given by the ResNet-50 model, by outperforming the VGG19 results.

ResNet-50 classified lung nodules with 0.80 accuracy, 0.79 Sensitivity, 0.81 Specificity and 0.77 F1-Score using the FPT approach.

VGG19 classified lung nodules with 0.72 accuracy, 0.75 Sensitivity, 0.69 Specificity and 0.68 F1-Score using the FPT approach.

## 7. Future works

The generality of this method also suggests that this approach can be used for other cancer detection problems. The design of the networks do not depend on any feature of lungs. Therefore, this approach can be used for other cancer types as well.

This approach can be extended to detect the exact location of the cancer nodules in the future, though currently it only classifies whether the images were either cancerous or not.

Training 3D CNN for nodule detection will be a next interesting approach that can be tried out after obtaining promising results from 2D CNN. Many of the successful results have been obtained by utilizing 3D CNN s.

The model parameters in all the cases have been saved based on the best accuracy. In addition to this we can use other matrices such as AUC, F1 score etc.

## 8. References

- [1] S. Chen and K. Suzuki, "Computerized Detection of Lung Nodules by Means of "Virtual Dual-Energy" Radiography," in IEEE Transactions on Biomedical Engineering, vol. 60, no. 2, pp. 369-378, Feb. 2013. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6353554&isnumber=6413218> [June 10, 2017]
- [2] W E Brant, C A Helms. (2012, May 08). Fundamentals of Diagnostic Radiology. (4<sup>th</sup> Edition) [On-line].4(4). Available: <https://www.bookdepository.com/Fundamentals-Diagnostic-Radiology-4-Set-William-E-Brant/9781608319121> [June. 10, 2017]
- [3] G. Coppini, S. Diciotti, M. Falchini, N. Villari and G. Valli, "Neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms," in IEEE Transactions on Information Technology in Biomedicine, vol. 7, no. 4, pp. 344-357, Dec. 2003. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1263906&isnumber=28273> [June 10, 2017]
- [4] K. Shigehiko, D. Kunio (2007, June–July). "Computer-aided diagnosis in chest radiography". Computerized Medical Imaging and Graphics. [Online]. 31(4-5). pp.212-223. Available: <https://www.sciencedirect.com/science/article/pii/S0895611107000286> [ June. 10, 2017]
- [5] J V Forrest, P J Friedman (1981, January). "Radiologic Errors in Patients with Lung Cancer ", Western Journal of Medicine. [Online]. 134(6). pp.485-490. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1272834/>. [June. 10, 2017]
- [6] G. Coppini, S. Diciotti, M. Falchini, N. Villari and G. Valli. (2003, Dec). "Neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms". IEEE Transactions on Information Technology in Biomedicine. 7(4). pp. 344-357. Available : <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1263906&isnumber=28273> [June. 10, 2017]



- [7] D. Blostein and N. Ahuja. (1989, Dec). "Shape from texture: integrating texture-element extraction and surface estimation ". IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(12). pp. 1233-1251. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=41363&isnumber=1583> .[ June. 10, 2017]
- [8] T. Lindeberg. (1994). Scale-Space Theory in Computer Vision. Kluwer Academic. [On-line]. Available: <https://link.springer.com/book/10.1007%2F978-1-4757-6465-9#toc> [June. 15, 2017]
- [9] M. Assefa, I. Faye, A. S. Malik and M. Shoaib.(2013) "Lung nodule detection using multi-resolution analysis," 2013 ICME International Conference on Complex Medical Engineering, Beijing [Online], pp. 457-461. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6548290&isnumber=6548194> [June. 15, 2017]
- [10] J.C. Bobadilla , H. Pedrini.(2016) "Lung Nodule Classification Based on Deep Convolutional Neural Networks". Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. pp 117-124 [Online].10125. pp.117-124 Available : [https://link.springer.com/chapter/10.1007/978-3-319-52277-7\\_15](https://link.springer.com/chapter/10.1007/978-3-319-52277-7_15) [June. 15, 2017]
- [11] K. Simonyan, A.Zisserman.(2015, April). "Very Deep Convolutional Networks for large Scale Image Recognition". [Online]. (6). Available: <https://arxiv.org/abs/1409.1556> . [June. 15, 2017]
- [13] G. Dheepak, S Premkumar, R.Ramachandran. (2015). "Lung cancer detection by using artificial neural network and fuzzy clustering method". International Journal of Power Control and Computation(IJPCSC). [Online]. 7(1). Pp. 24-28. Available: <https://www.semanticscholar.org/paper/Lung-Cancer-Detection-by-Using-Artificial-Neural-DHEEPAK-Premkumar/d35fbc618b4982c2e8fcc8f2f063f0f1a312d94c> [June. 15, 2017]
- [14] Masood A. Hussain<sup>1</sup>, Tabassum M. Ansari , Prarthana S. Gawas , Nabanita Nath Chowdhury.(2015, March) "Lung Cancer Detection Using Artificial Neural Network & Fuzzy

Clustering”. American Journal of Biomedical Engineering. [Online]. 2(3). pp 136-142. Available: <https://pdfs.semanticscholar.org/63bd/5d0e2845940b1807d50f34ac47f4405fd3fe.pdf> [June. 15, 2017]

[15] R.S. Shriwas, A.D. Dikondawar. (2015, January). “Lung cancer detection and prediction by using neural network”. IPASJ International Journal of Electronics & Communication (IJEC). [Online]. 3(1). Available: <http://ipasj.org/IJEC/Volume3Issue1/IJEC-2015-01-11-6.pdf> [June. 15, 2017]

[16] A. Pathan, and B.K.Saptalkar. “Detection and Classification of Lung Cancer Using Artificial Neural Network”, International Journal on Advanced Computer Engineering and Communication Technology. [Online]. 1(1). Available: [http://www.irdindia.in/journal\\_ijacect/pdf/vol1\\_iss1/11.pdf](http://www.irdindia.in/journal_ijacect/pdf/vol1_iss1/11.pdf) [June. 15, 2017]

[19] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE,1998. pp. 2278-2324.

[20] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhber. (2010, December). “Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition”, Neural Computation. [Online]. 22(12). Available: <https://arxiv.org/pdf/1003.0358.pdf> [June. 25, 2017]

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, 2012, pp.1097-1105.

[22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun. (2014, February). “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks”. Computer Vision and Pattern Recognition. [Online]. (4). Available: <https://arxiv.org/abs/1312.6229> [June. 25, 2017]

[23] Karen Simonyan, Andrew Zisserman, ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, conference paper at ICLR, 2015.

- [24] A. Canziani, A. Paszke, E. Culurciello, (2016). An Analysis of Deep Neural Network Models for Practical Applications', CoRR.[Online]. abs/1605.07678(2). Available: <https://arxiv.org/abs/1605.07678> [June. 25, 2017]
- [25] M. Lin, O Chen, S Yan, "Network In Network", International Conference on Learning Representations (ICLR) 2014.
- [26] C Szegedy, W Liu, Y Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, A Rabinovich. (2015) "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [Online]. pp. 1-9. Available: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7298594&isnumber=7298593> [July 10, 2017]
- [27] A Canziani, A Paszke, E Culurciello,(2016, May) "An Analysis of Deep Neural Network Models for Practical Applications". CoRR. [Online]. (2). Available : <https://arxiv.org/abs/1605.07678v2> [July. 10, 2017]
- [28] S Ioffe, C Szegedy. (2015, March) "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift"[Online], 3. Available : <https://arxiv.org/pdf/1502.03167.pdf> [July 10, 2017]
- [29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 2818-2826.
- [30] J Jin, Aysegul D, E Culurciello, "Flattened Convolutional Neural Networks for Feedforward Acceleration". International Conference on Learning Representations (ICLR), 2015.
- [31] Cheng J Z , Ni D, Chou Y H, Q Jing, Tiu CM, Chang Y C, Huang C S, Shen D, Chen C M.(2016, April) "Computer-Aided Diagnosis with Deep Learning Architecture: Applications to Breast Lesions in US Images and Pulmonary Nodules in CT Scans". Scientific Reports. [Online]. 6. Available : <https://www.nature.com/articles/srep24454> [July 15, 2017]

- [32] H. Yang, H Yu, G Wang.(2016) “Deep Learning for the Classification of Lung Nodules”.CoRR.[Online]. abs/1611.06651. Available : <https://arxiv.org/ftp/arxiv/papers/1611/1611.06651.pdf> [ July. 15, 2017]
- [33] S. Ramaswamy and K Truong, “Pulmonary Nodule Classification with Convolutional Neural Networks.” Internet : [http://cs231n.stanford.edu/reports/2016/pdfs/324\\_Report.pdf](http://cs231n.stanford.edu/reports/2016/pdfs/324_Report.pdf) [July 15, 2017]
- [35] Kaggle, “Data Science Bowl 2017.” <https://www.kaggle.com/c/data-science-bowl-2017/#description> , [July 15, 2017]
- [36] LUNA16, “Lung nodule analysis 2016.” <https://luna16.grand-challenge.org/>,<https://www.kaggle.com/c/data-science-bowl-2017> [October 15, 2017]
- [37] Konstantinos Sechidis. “Comparison of different preprocessing techniques and feature selection algorithms in cancer datasets” .Internet: <http://syllabus.cs.manchester.ac.uk/pgt/2017/COMP61011/goodProjects/Sechidis.pdf> [August 15, 2017]
- [38] Youssef Kashef. “What are some ways of pre-processing images before applying convolutional neural networks for the task of image classification?” Internet: <https://www.quora.com/What-are-some-ways-of-pre-procesing-images-before-applying-convolutional-neural-networks-for-the-task-of-image-classification>> [August 18, 2017]
- [39] Sasidhar B, R Babu D R, R Shankar M, B Rao N .(2013, September). “Automated Segmentation of Lung Regions using Morphological Operators in CT scan”. International Journal of Scientific & Engineering Research. [Online]. 4(9). pp. 1114-1118 Available : <https://www.ijser.org/researchpaper/Automated-Segmentation-of-Lung-Regions-using-Morphological-Operators-in-CT-scan.pdf> [July 15, 2017]
- [40] D Baker, J Kilpatrick, A Chaudhry(Spring ,2017). “Predicting Lung Cancer Incidence from CT Imagery”. . Internet: <http://cs231n.stanford.edu/reports/2017/pdfs/515.pdf> [November 15, 2017]

- [41] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.
- [42] K. He and J. Sun, "Convolutional neural networks at constrained time cost," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 5353-5360.
- [43] R. K. Srivastava, K. Greff, and J. Schmidhuber(2015, November). "Highway networks". Presented at ICML 2015 Deep Learning workshop, 2015, pp 2-4.
- [44] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.
- [45] S Lalondrelle, S A Sohaib, I A Castellano, R Huddart, and V Khoo.(2012, August). "Investigating the relationship between virtual cystoscopy image quality and CT slice thickness", [Online]. 85(1016), pp. 1112–1117. Available : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3587099/#> [November 25, 2017]
- [46] M.S.Al-Tarawneh (2012, May), "Lung cancer detection using image processing techniques," Leonardo Electronic Journal of Practices and Technologies,[Online]. 20. pp. 147–158. Available: <https://pdfs.semanticscholar.org/bd16/59320feecb20a33a75f6003f8917212de30f.pdf> [December 3, 2017]
- [47] D Baker, J Kilpatrick, A Chaudhry(Spring ,2017). "Predicting Lung Cancer Incidence from CT Imagery".Internet: <http://cs231n.stanford.edu/reports/2017/pdfs/515.pdf> [November 15, 2017]
- [48] "Convolutional Neural Networks for Visual Recognition". Internet: <http://cs231n.github.io/transfer-learning/> [31<sup>st</sup> December 2017]

- [49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," CoRR, vol. abs/1505.04597, 2015.
- [50] A. M. Rossetto and W. Zhou. "Deep Learning for Categorization of Lung Cancer CT Images," IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies, 2017 pp. 272-273.
- [51] Mehdi Fatan Serj, Bahram Lavi, Gabriela Hoff, and Domenec Puig Valls. "A Deep Convolutional Neural Network for Lung Cancer Diagnostic". [Online]. Available: <https://arxiv.org/pdf/1804.08170.pdf>0020 [May. 16, 2018]
- [52] M Ravaut, G Manek, H Chen, J Lin, B Nazir, C Chen, T C Howe, Z Zeng, V Chandrasekhar, "Deep Learning for Lung Cancer Detection: Tackling the Kaggle Data Science Bowl 2017 Challenge".CoRR [Online]. Available: <https://arxiv.org/pdf/1705.09435.pdf> [May. 16, 2018]
- [53] F Taher<sup>1</sup>, N Werghi, H Al-Ahmad, R Sammouda.(2012). "Lung Cancer Detection by Using Artificial Neural Network and Fuzzy Clustering Methods". American Journal of Biomedical Engineering [Online], 2(3): pp.136-142. Available: <https://pdfs.semanticscholar.org/63bd/5d0e2845940b1807d50f34ac47f4405fd3fe.pdf> [May. 16, 2018]
- [54] W Alakwaa, M Nassef, A Badr.(2017) "Lung Cancer Detection and Classification with 3D Convolutional Neural Network (3D-CNN)". (IJACSA) International Journal of Advanced Computer Science and Applications. [Online]. 8(8). Available: <https://pdfs.semanticscholar.org/4338/88ac5229432ad85f1d31a1e660a96244960f.pdf> [May. 16, 2018]
- [55] T. Jin, H. Cui, S. Zeng and X. Wang, "Learning Deep Spatial Lung Features by 3D Convolutional Neural Network for Early Cancer Detection," 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, NSW, 2017, pp. 1-6. [May. 16, 2018]

[56] Hao Zhang. “Why convolutional neural networks is so important to learn?” Internet: <https://www.quora.com/Why-convolutional-neural-networks-is-so-important-to-learn> [May 17, 2018]

[57] QingZeng Song, Lei Zhao, XingKe Luo, XueChen Dou. (2017).“Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images”

[58] G L. F. da Silva, A C. Silva ´ , A C. de Paiva1, M Gattass “Classification of malignancy of lung nodules in CT images using Convolutional Neural Network”.  
<http://www.lbd.dcc.ufmg.br/colecoes/wim/2016/003.pdf>

[59] K L Hua, C H Hsu, Si C Hidayati, W H Cheng, Y J Chen “Computer-aided classification of lung nodules on computed tomography images via deep learning technique” .  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4531007/pdf/ott-8-2015.pdf>