LUNG CANCER NODULE DETECTION AND CLASSIFICATION USING SEMANTIC SEGMENTATION AND DEEP LEARNING ALGORITHM

MANJARI P.

(Roll No: 19PB31)

Dissertation submitted in partial fulfilment of the requirements for the degree of

MASTER OF ENGINEERING

Branch: INFORMATION TECHNOLOGY

Specialization: BIOMETRICS AND CYBER SECURITY

of Anna University

JUNE 2021

DEPARTMENT OF INFORMATION AND TECHNOLOGY

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE - 641004

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE - 641 004

LUNG CANCER NODULE DETECTION AND CLASSIFICATION USING SEMANTIC SEGMENTATION AND DEEP LEARNING ALGORITHM

Bona fide record of work done by

MANJARI P

(Roll No:19PB31)

Dissertation submitted in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING

Branch: INFORMATION TECHNOLOGY

Specialization: BIOMETRICS AND CYBER SECURITY

of Anna University

June 2021

……………………..	…………………………
Dr. Sangeetha B	**Dr. Umamaheswari K**
Faculty guide	Head of the Department

Certified that the candidate was examined in the viva-voce examination held on .06.2021

………………..	…………………
(Internal Examiner)	**(External Examiner)**

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to **Dr.K.Prakasan**, Principal, PSG College of Technology, for giving us the opportunity to do our project with various facilities and infrastructure, without which success of the project would not have been possible.

I extend our heartfelt thanks to **Dr.K.Umamaheswari**, Professor and Head, Department of Information Technology, PSG College of Technology, for her unfailing support throughout this project.

I take this opportunity to express a deep sense of gratitude to **Dr. Chandran K R**, Professor, Department of Information Technology, for his cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I take this opportunity to thank our Coordinator, **Dr. Anitha Kumari K**, Associate Professor, Department of Information Technology, for her moral support and guidance in all the instances.

I express our sincere thanks to our faculty guide **Dr. Sangeetha B,** Assistant Professor (Sr. Gr.), Department of Information Technology, PSG College of Technology, who was always there to help me and played a major role in the completion of the project and I wish to thank her for enduring guidance and priceless advice throughout this project work.

I convey my regards to our tutor**, Ms. N. Ravitha Rajalakshmi,** Assistant Professor (Sr. Gr.), Department of Information Technology, for her guidance and encouragement all through the course of the project.

Finally, I thank all the committee members, faculties and Technical Assistant of the Information Technology Department, PSG College of Technology, who have been a source of great help in the venture.

SYNOPSIS

Lung cancer is the most deadly cancer when compared to other cancers. Early detection of lung cancer will improve the survival rate of the individual. Analyzing the CT scan images manually and detecting the existence of cancer is a tedious process for the physicians. This work focuses on developing a computer aided system to help the physicians in detecting the lung cancer nodules in the CT scan images efficiently. The data set are obtain from LUNA repository. The lung image which is in raw format along with the annotated file are collected, undergone thresholding for the elimination of bone structure. Context aware U-net used for the purpose of segmenting the nodule from the whole lung image. The context aware U-net is evaluated using the metrics like IOU, accuracy, precision, recall and loss. The segmented nodules and non-nodules are given into Convolutional Neural Network (CNN) algorithm which is an advanced deep learning algorithm for the purpose of feature extraction and classification. The performance of the classification system is evaluated using precision, recall and f-measure.

LIST OF FIGURES

LIST OF TABLES

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Lung cancer

Lung cancer has become the most common type of cancer accounting to 1.76 million deaths world-wide published by the Cancer Research institute. Cancer results from abnormalities (uncontrolled growth) in the cells. During medical checkup the patient may be found with nodule which may be benign or malignant. Nearly 80% of cancer is caused due to smoking. The other reasons may be exposure to the secondary smokes. The exposure of people to smoke from other people's cigarette pipes are called secondary or second-hand smoke. The prognosis of cancer at an early stage may improve the longevity of the individual's life. The primary issue with detection of cancer is to identify the cell proliferation as benign or malignant. The malignant lung cancer growth can be classified into two categories: Non-Small Cell Lung Cancer (NSCLC) and Small Cell Lung Cancer (SCLC). Different imaging modalities like Computer Aided Tomography (CT), MRI, Ultrasound images are used in the detection of lung cancer. Automated cancer detection tools are being developed with the support of digital image processing techniques and the advancements in machine learning algorithms. Computer - Aided diagnosis has recently developed as a promising tool to supplement the physicians in identifying a tumor as benign or malignant. Early detection of malignant nodule helps in better treatment.

1.2 Image Processing

Image processing is the process where the image is converted to digital format for the purpose of enhancing the image and performing some operations to get some information from the image. This process include rendering the original image from raw input, noise removal, image enhancement and segmentation. In the image processing the RCB image will be converted to grey scale image. Then the noise is removed using different types of filters based on the noise. The image quality in enhanced if the image is being blur.

1.3 Segmentation

Segmentation is the process of separation a required object from the whole image. There are different types of segmentation.

1.3.1 Region-based Segmentation

One simple way to segment different objects is to use the pixel values of image. In image the pixel values for objects and their background will be different.
In this case, we can set a threshold value. Based on the pixel value rising or falling in considerate with threshold the background and the objects can be classified. This technique is known as Threshold Segmentation.

1.3.2 Edge Detection Segmentation

In this method edge filter is applied to the image for classification. The output of the filter is taken into the consideration for classification. The region which are not separated by edge are considered as same region.

1.3.3 Image Segmentation based on Clustering

Segmenting is the process of putting customers into groups based on similarities, and clustering is the process of finding similarities in customers so that they can be grouped, and therefore segmented

1.3.4 Mask R-CNN

Mask R-CNN is created using improvement in Faster R-CNN. It is used for object detects. It return the bounding box along with the class labels for given image.

1.3.5 Semantic segmentation

Semantic segmentation describes the process of associating each pixel of an image with a class label(such as flower, person, road, sky, ocean, or car).

1.4 Deep learning

1.4.1 Deep learning algorithm evolution

Deep learning begins back at 1943 when neural networks of human brain based computer was developed by Warren McCulloch and Walter Pitts. Warren and Walter made use of the combination of algorithms and mathematics which was called as threshold logic.

Information is transmitted through each layer and the output of the previous layer providing input for the next layer. The initial layer in a network is called the input layer, while the final layer is called an output layer. All the layers in-between the first and last layer are referred to as hidden layers. Each layer is simple, uniform algorithm containing one kind of activation function. The earliest efforts in developing Deep Learning algorithms done by Alexey Grigoryevich Ivakhnenko and Valentin Grigor'evich developed the *Group Method of Data Handling* and *Cybernetics and Forecasting Techniques* respectively in the year of 1965. They used models with complicated equations such as polynomial activation functions that were then analysed in a static way. After each layer the chosen characteristics are passed to the next layer.

The first convolutional neural networks were coined by Kunihiko Fukushima. Fukushima designed neural networks that contains multiple pooling and convolutional layers. He developed an artificial neural network which is called as Neocognitron in the year of 1979, which utilized a hierarchical, multi-layered design. This design enabled the machine to recognize visual patterns. These are trained with recurring activation in multiple layers, which has gained strength over time. Additionally, Fukushima’s design increased the weight of certain connections that resulted in manual adjustment of important features.

1.4.2 Deep Learning Algorithm

Deep Learning Algorithm is a part of machine learning algorithm which uses multiple layer to extract minute layer of features from raw input. For instance low level identifies edges whereas higher level identifies the features which can be identified by human eyes.

Now a days the modern deep learning uses artificial neural network especially the convolutional neural network. In deep learning each level is converted to composite and abstract composition. In deep learning algorithm the word deep refers to the number of layers via which the data is being transformed. In this method the transformation from input to output is referred as substantial credit assignment path (CAP). Feature extraction is a key member of deep learning. Feature extraction is a process of an algorithm to automatically construct meaningful “features” of the data for purposes of training, learning, and understanding.

1.5 NEURAL NETWORK

A neural network is a collection of algorithms that attempts to recognize the underlying relationships in a series of data by imitating the functioning of the human brain. Neural networks must adapt to changing inputs. Thus, the network produces the best possible output without redesigning the performance criteria. Neural networks are quickly gaining interest in the design of commercial systems and have their origins in artificial intelligence.

A typical neural network is made up of a series of layers, each of which connected on either side of the layer from several dozen to hundreds of thousands or even millions of artificial neurons called units. Some of them, known as input units, are configured to obtain various forms of information that the network tries to understand, interpret and process from outside. Additional units are on the other side of the network and indicate how the data obtain responds these are known as the output units.

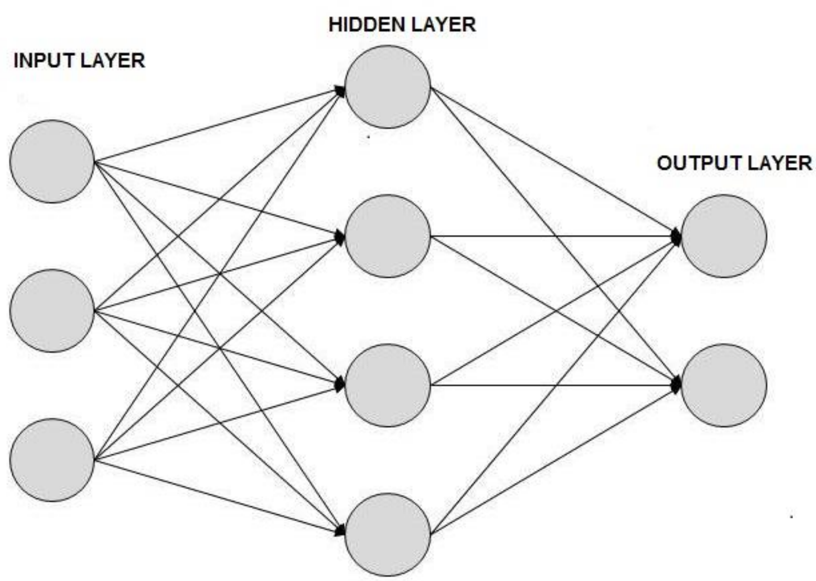


Fig 1.1.General Representation of Neural Network

There is one or more layers of hidden units in between the input unit and the output unit that together form most of the artificial brain. The majority of neural networks are fully connected which means that each hidden unit and each output unit in the layers are connected to each other. The relations between units are defined by a weight number, which can be positive or negative. The increasing the size, the more the unit impacts another units.

An interconnected neural network comprises layers of nodes. Each node is a perception and resembles a multiple linear regression. The perceptron transfers the signal produced by linear regressions into a non-linear activation function. Hidden layers change the weighting of the input to reduce the error margin of the neural network. Hidden layers are believed to extrapolate salient characteristics of input data that predict the outputs. It explains extracting features that are useful, such as the key element analysis, to achieve statistical techniques.

1.5.1 CONVOLUTIONAL NEURAL NETWORK

One of the well-known deep learning networks called the Convolution Neural Network (CNN) is applied in this paper to use the benefit of feature learning and classification.

Transformation is defined with respect to these parameters

$$A = \alpha\,(w^T x + b) \tag{1.1}$$

Where *x* is the inputs of neurons, *w* is the weights, *b* is bias, *T* is matrix transpose and *σ* is activation function. Computing higher-level features from lower-level features from observational data is one of the main principles of deep learning. CNN has recently been successfully applied in various works such as image identification, scene text recognition, object tracking, speech recognition, attitude prediction, natural language processing, visual saliency detection and recognition of human action.

The CNN is a neural network variant where its purpose is to learn correct feature representations of input data. There are two main differences between a CNN and MLPs, namely weight sharing and pooling. Each CNN layer can consist of many convolution kernels that are used to generate various feature maps. Each neighbouring neuron region is linked to a next layer feature map neuron. In addition, all the spatial positions of the input share the kernel to create the feature map. One or more full connected layers are used for classification after some convolution and pooling layers**.**

Using shared weights in CNN, the model can learn the same pattern that occurs at different input positions, without demanding separate detectors to learn for each position. Therefore, the model can be robust for translating inputs.

The pooling layers reduce the computational burden by decreasing the number of connections between the Convolutional layers. In addition, pooling layers boost translation invariance properties and enhance the receptive field of subsequent Convolutional layers. Usually, one or more fully connected layers are added at the end of the Convolutional network, and a loss function is used to quantify errors for training purposes.

During the CNN learning process, a small window is sliding over the inputs and the values of bias and weights through this window can be adjusted from different features of the input data without their position in the input data.

Deterministic Analysis

Basically, the system operates on known principles that include Machine Learning deterministic and probabilistic methods. Deterministic machine learning uses small data sets that are evaluated from any normal pattern for any deviations. This data is then analysed by IT professionals who devise models for further software analysis. Typically, the information obtained is compared to a baseline, as any data above normal levels is considered to be intrusive action

Probabilistic Analysis

The probability machine learning goes on, however, because it tests the patterns that could have missed the determinist analysis involved in the assessment. Importantly, to detect any anomalous data character, the system relies on clustering. The system relies on unsupervised operation where the system runs independently to produce a map which is then evaluated for any abnormal behavior by the same computer. Consequently, this can define the exact problem with conservative estimates placing it at 90 percent, since the analysis is conclusive and the method is more effective.

Deep Coding Networks

The benefits of deep coding networks in deep learning techniques have also recently gained traction. The method is flexible because the model is adaptive and adapts to new databases. In general, the method uses the outcomes from top-down approaches as inputs to bottom-up approaches. In addition, the model extracts features with linear models, which in turn are used to shape deep system architecture as construction elements of layers that are dependent on one another.

1.5.2 DESIGN OF CONVOLUTIONAL NEURAL NETWORK

An input, an output layer and several hidden layers form a Convolutional Neural Network. The hidden layers of a CNN consist usually of a series of convolutional layers, which combine with a multiplication or another dot product The activation feature is often a RELU layer, and then additional convolutions such as pooling layers, fully connected layers and normalization layers are called hidden layers since the activation and the final convolution covers their inputs and outputs. In addition, the final convolution also requires back propagation to weight the final product more precisely. The different types of hidden layers are

- Convolutional layer
- Pooling layer
- Fully connected layer

Convolutional layer

Regarding CNN applications, the following attributes should be used for every Convolutional layer of a neural network. Input layer is a form of (number of image) x (image width) x (image height) x (image depth). Convolutional kernel with a hyper parameter width and height and a width equal to the image's thickness Input is combined with Convolutional layers and its output is transferred to the next level. It resembles the neuron's response to a specific stimulus in the visual cortex

Each neuron processes information specifically for their area of reception. Although neural networks can be used fully connected to acquire features and to classify data, implementing this architecture to the image is not really practical. Due to very large input sizes associated with images, where each pixel is a specific parameter, a very high number of neurons would be required also in the shallow architecture. In the practice of conventional multi-layer neural networks with many layers with back propagating, it solves the problem of loss or explosion.

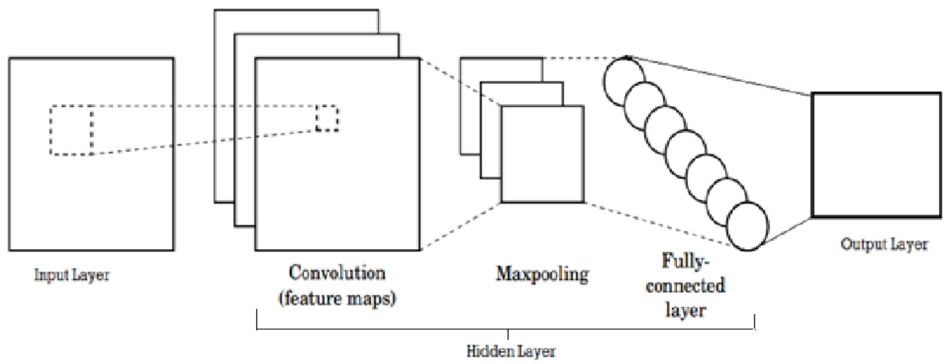


Fig .1.2 Convolutional Neural Network

Spatial arrangement

The size of the output volume of the convolution layer is regulated by three hyper parameters: depth, stride, and zero-padding.

- The output volume depth determines the number of neurons in a layer contributing to the same input volume area. Such neurons learn to activate input for various features. For example, if the first Convolutional layer takes the raw image as input, then different neurons along the depth dimension will activate in the presence of different oriented edges or color blobs
- Stride controls how columns of depth are allocated around the dimensions of space (width and height). When the stride is 1, then push one pixel at a time of the filters. This results in strongly overlapping receptive fields between the columns and large volumes of output

Pooling layer

In order to streamline the underlying computation, modern networks may include local or global pooling layers. Pooling layers minimize data size by merging the neuron cluster outputs from one layer in the next layer into a single neuron.

Large clusters, usually 2x2, are mixed in the local pool. Global pooling affects all the neurons in the Convolutional layer. Pooling can also measure max, min or an average pooling. The maximum value of each cluster of neurons at the previous layer is used by Max pooling. Average pooling uses the average value of each of the previous layer's neuron clusters.

Fully connected layer

Fully connected layers link each layer of neuron to each layer of neurons. It is exactly the same as the Conventional neural perceptron multi-layer (MLP) network. To identify the objects, the flattened matrix passes through a fully connected layer.

Dense layer

A linear operation where each input is related by a weight to each. Activation function determines whether a neuron is activated or not by measuring a weighted sum and adding more bias with it. The activation function aims to introduce non-linearity into a neuron's output. The activation function performs the non-linear transition to the data, allowing it to learn more complex tasks and execute them.

Loss functions

A loss function is incredibly simple; it’s a way to assess how well the algorithm models the dataset. If the predictions are absolutely wrong, a higher number will be given by the loss feature. If they are pretty good, a lower number will be given. As the adjustment of the algorithm parts it try to improve the model and also find the loss function. Loss functions fall under four major categories:

Regressive loss functions

These are used when the goal variable is constant in case of regressive problems. Mean Square Error is the most widely used regressive loss function. Some reasons for failure are:

1. Squared Error Loss
2. Absolute Error Loss
3. Huber Loss

Classification loss functions

In classification problem, the output variable is typically a probability value f(x), called the input score x. Generally speaking, the score magnitude reflects our prediction's confidence. The target variable y is a binary, true variable 1 and false variable-1.Many algorithms of classification are:

1. Binary Cross Entropy
2. Negative Log Likelihood
3. Margin Classifier
4. Soft Margin Classifier

Embedding loss functions

It deals with issues where two outputs are identical and dissimilar to each other.

1.6 Objective

The detection of nodule in the lungs can be identified using the deep learning algorithm with the help of convolutional neural network which extract the features from the training data set and helps in prediction. This method takes computed tomographic images of lungs as input.

Even small nodule of size above 3 mm in diameter has the chance of becoming into cancerous nodule. The objective of the project is to detect this type of nodule which helps the physicians in providing better treatment and to reduce the death rate.

1.7 Organisation of the Report

The report is structured as follows. Chapter 2 contains the literature survey. Proposed design is explained in Chapter 3. Software requirements are specified in Chapter 4. The Implementation and result analysis are described in Chapter 5. Conclusion and future work are described in Chapter 6.

CHAPTER 2

Literature Survey

Tasnim Ahmed, Mst. Shahnaj Parvin, Mohammad Reduanul Haque, Mohammad Shorif Uddin [1] developed a system which uses Vanilla 3D CNN as a classifier to determine whether the given image is cancerous or non-cancerous. The pre-processing is done using thresholding technique. Luna 16 has been used as a data set. The convolution layer is formed of two pooling layer and one fully connected layer. This methods provides an accuracy of about 80 percentage.

Wafaa Alakwaa, Mohammad Nassef and Amr Badr [2] proposed a model to detect the nodule using three dimensional convolutional neural network which gives an accuracy of about 86.6%. Threshold, Watershed, and U-Net are used to identify the nodules of patients. The network can be trained end-to-end from raw image patches. Its main requirement is the availability of training database, but otherwise no assumptions are made about the objects of interest or underlying image modality. The interest of point is identified using U-Net architecture. The future work is to develop a deeper network and more extensive hyper parameter tuning.

Ivan William Harsono, Suryadiputra Liawatimena, and Tjeng Wawan Cenggoro [3] uses a unique way for extraction of features from the CT images. This method brings a way to standardize the CT scan of various age groups and varying body size. 3D slicer library is used for the high end reconstruction which can eliminate the data corruption partially. The noise in the CT scan are eliminated by iterative reconstruction of raw data. This propose method provide higher accuracy.

Moffy Vas and Amita Dessai [4] worked with morphological operations for the purpose of segmentation the region of interest from which features are extracted and utilized for the classification of nodule by artificial neural network. Median filter is used for the purpose of impulse noise reduction. This method produces 92% of accuracy.

Xiaojie Huang, Junjie Shan, and Vivek Vaidya [5] proposed a 3D CNN for nodule detection in low dose CT images. This proposed method reduced the complexity and made the model to train with less number of data set. The two methods involved in the system are nodule candidate generation using local geometrical model based filter and classification. The limitations in this method are pleural nodules were not addressed and this experiment was limited to cross validation.

Rohit Y. Bhalerao, Harsh P. Jani, Rachana K. Gaitonde, and Vinit Raut [6] proposed a model that uses Convolutional neural network technique. In this method three convolutional layers have been used for feature extraction. The accuracy obtained is 94.34% and precision is 91.75%

Ayman El-Baz, Garth M. Beache, Georgy Gimel'farb, Kenji Suzuki, Kazunori Okada, Ahmed Elnakib, Ahmed Soliman, and Behnoush Abdollahi [7] provided an overview about problems faced in oncology, the problems faced during lung cancer diagnosis. Nearly 360 articles have been surveyed. The challenges faced are

1. Search space Reduction: To reduce the search space of the nodule, accurate segmentation must be done. Segmenting the nodule is difficult because of the inhomogeneity of the lung regions and the pulmonary structures with similar densities like veins, arteries and bronchi.
2. Need for the larger database. Larger the data set the accuracy of validation will be high.
3. Automation level, speed and ability to detect the nodule are the different measuring factors. These factors will get affected when the nodule is in different shape, or it is attached to the wall of lungs.

Ying Su , Dan Li , Xiaodong Chen propose a work that detects the lung nodule using faster R-CNN framework. They have used Database LIDC-IDRI as the data set. Data enhancement: most lung nodules are spheroids in reality, and the radiologist only marked the CT image with the largest cross-sectional area, whereas one or two slices adjacent to each other are not marked. The upper and lower images of the larger lung nodules marked by the radiologist are added to the data set. Then, the next image of the medium lung nodule is added to the data set. Using this method to enhance the original data, the original 3,042 nodules were enhanced to 7,000 slices as a complete data set. The accuracy obtained is about 91.2%. The future enhancement is to add small benign nodules to the data set.

Benita K. J. Veronica proposed an effective neural network model for lung nodule detection in CT images with optimal fuzzy model. The data set used in ELCAP lung image database. For the process of segmentation it uses Fuzzy C-Means (FCM). The classification algorithm used is Artificial Neural Network (ANN) along with weight optimization. The specificity sensitivity, specificity, accuracy is of the segmentation algorithm are 96.1, 99.86 and 99.85 respectively. The accuracy of the classification algorithm is 86%. The future enhancement includes improving the classification accuracy by extracting some unique features of lung image.

Issa Ali , Gregory R. Hart , Gowthaman Gunabushanam , Ying Liang , Wazir Muhammad , Bradley Nartowt , Michael Kane , Xiaomei Ma and Jun Deng proposed Lung Nodule Detection via deep Reinforcement learning. This method uses LIDC/IDRI (LUNA) data set. They have used reinforcement learning model based on deep artificial neural networks. The training results yielded an overall accuracy of 99.1% [sensitivity 99.2%, specificity 99.1%, positive predictive value (PPV) 99.1%, and negative predictive value (NPV) 99.2%]. In the test, the results yielded an overall accuracy of 64.4% (sensitivity 58.9%, specificity 55.3%, PPV 54.2%, and NPV 60.0%). Accuracy Sensitivity Specificity are lesser while testing.

Nasrullah Nasrullah, Jun Sang , Mohammad S. Alam , Muhammad Mateen , Bin Cai and Haibo Hu developed a system to automatically detect and classify the nodule using multiple techniques. They have used LIDC/IDRI (LUNA) data set. 3D Faster R-CNN with CMixNet and a U-Net-like encoder–decoder has been used to detect for the presence of nodules. 3D CMixNet with GBM has been used to classify the nodule and non nodule. The accuracy obtained is 88.79%.

Haichao Cao, Hong Liu, Enmin Song, Senior Member,IEEE, Guangzhi Ma, Xiangyang Xu, Renchao Jin, Tengying Liu, and Chih-Cheng Hung developed a Two-Stage Convolutional Neural Networks for Lung Nodule Detection. They have used LIDC/IDRI (LUNA) data set. For segmentation improved U-net architecture has been used. The model includes two phases of detection. First phase prediction has 128*128*3 (stride size 64) for the rough detection and second-phase prediction in which a patch size of 64*64*3 is used for the scanning. The purpose of this second-phase prediction is to make an accurate segmentation in the local region where the suspected nodules can be located. The sensitivity has reached 90% which indicates that the network architecture based on 3D-CNN can adapt to the classification of true and false lung nodules to some extent.

Table 2.1 Related works comparison

S. No	Paper Title	Year	Dataset	Preprocessing	Methodology	Metrics	Advantage	Remarks
1	Lung Nodule Detection via Deep Reinforcement Learning	2018	(LIDC/IDRI)	Data Normalization	CNN, deep reinforcement learning	Accuracy of 64.4% (sensitivity 58.9%, specificity 55.3%)	Solving the major issue of false positives in CT screening of lung nodules.	Accuracy Sensitivity Specificity are lesser while testing
2	A Two-Stage Convolutional Neural Networks for Lung Nodule Detection	2020	LUNA dataset	-	improved U-Net segmentation 3D CNN	Competition performance metric (CPM) Sensitivity-95%	False Positive reduction.	Reduction in recall rate.
	An effective neural network model for lung nodule detection in CT images with optimal fuzzy model	2020	(ELCAP lung image database)	Adjust image intensity	Segmented -> the Fuzzy C-Means (FCM) Classification -> Artificial Neural Network (ANN) with weight optimization	Accuracy of 0.866	better fitness of 0.97 at the maximum iteration	improve the classification accuracy by extracting some unique features of lung image
4	Lung cancer detection system using lung CT image processing	2017	V.M.Salgaocar hospital, SMRC and the Manipal	Median Filter	Artificial neural network	accuracy of 92%	Improved accuracy by eliminating	Morphological closing leads to loss of nodules

			hospital both situated in Goa				impulse noise	
5	A novel approach for detection of Lung Cancer using Digital Image Processing and Convolution Neural Networks	2019	LIDC	Binarization, normalization and Median filter	U-Net architecture-segmentation 3D multipath architecture-prediction	Sensitivity 91.75% Specificity 95.70% Precision 91.75% Accuracy 94.34%	High accuracy and less false positive rate.	Increasing convolutional layers reduces efficiency and performance
6	Lung Cancer Detection using CT Scan Images	2018	LIDC	Median filter, Gaussian filter	Segmentation -> watershed segmentation Classification -> Support vector Classifier	Detection 92% accuracy classifier 86.6% accuracy	Increased Accuracy	Proper pre-processing and elimination of false objects.
7	Lung Nodule Texture Detection and Classification Using 3D CNN	2019	Mascow private dataset	-	3D CNN	70.36% of Area Under the Curve (AUC) 22.86% of mean average precision (mAP) detection capability	Detects and classifies the texture information	High noise in mascow dataset. Read and conversion is difficult Less training data set
8	Lung Cancer Detection Using CT Image Based on 3D Convolutional Neural Network	2020	LUNA 16 dataset	Resizing, averaging and thresholding	Vanilla 3D CNN	Accuracy - 80%	Better accuracy	Less training data set
9	Lung Cancer Detection and Classification	2017	LUNA 16 dataset	Thresholding and U-net for segmentation	3D-CNN	Accuracy of 86.6%	Enhancement on accuracy	High False Positive- 14%

	with 3D Convolutional Neural Network						due to U-Net architecture and segmentation	
10	Lung Nodule Detection based on Faster R-CNN Framework	2020	LIDC-IDRI	The upper and lower images of the lung nodules marked by the radiologist are added to the data set. Then, the next image of the medium lung nodule is added to the data set.	Faster R-CNN algorithm	Accuracy 91%	Better accuracy	Small nodule benign data set is less in number. The model's sensitivity to small nodules is less.

CHAPTER 3
PROJECT DESIGN

3.1 PROBLEM STATEMENT

Analysing the CT scan images manually and detecting the existence of cancer is a tedious process for the physicians. Developing a computer aided system to help the physicians in detecting the lung cancer nodules in the CT scan images efficiently. This work focuses on developing a computer aided system for detecting which are cancerous nodules with the help of CT scan images in efficient manner. Semantic segmentation is used to segment the nodule from the CT scan. Deep Learning algorithm is used to locate the cancerous nodules in lung images. The performance of the system is evaluated using dice coefficient, precision, recall and f-measure.

3.2 OVERVIEW OF THE PROJECT

In this proposed work, data is collected from the LUNA 16 for lung cancer nodule detection. This data set are available in .raw and .mhd format along with annotation file.

The nodule location and the size of the nodule will be given in the annotation file. First the image is rendered from the raw file. The rendered image are given as input to thresholding for the elimination of bone regions. The computed tomography has been measured using Hounsfield unit where lungs parts intensity will be high and bone region will be in negative value. To eliminate the bone region thresholding is used. Next the pre-processed image are taken under semantic segmentation where all the nodule and non-nodules are segmented from the image. The segmented nodule and non-nodule are trained using Convolutional neural network. The convolutional neural network consist 3 layers where different parameters have been trained.

After the feature extraction prediction of nodule is done. Performance measure are calculated using precision, f-measure and recall.

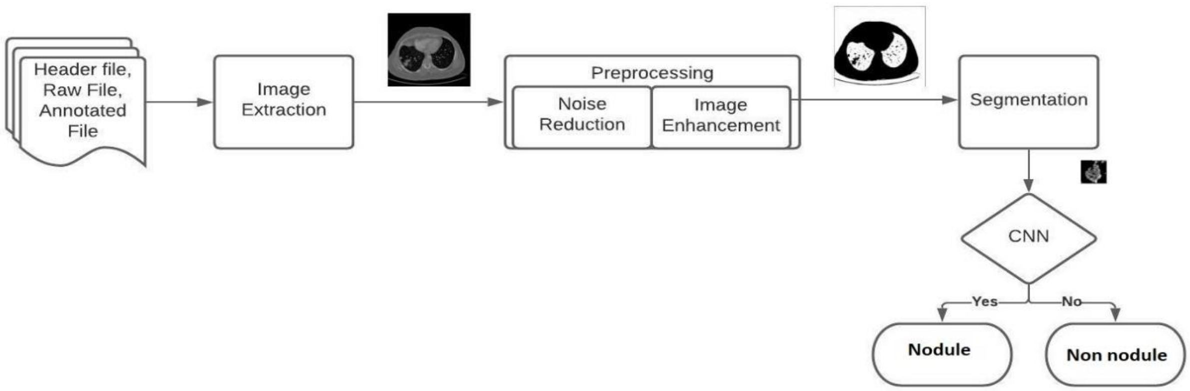


Fig 3.1. Lung Cancer Nodule Detection

3.3 MODULE DESCRIPTION

This project deals with the nodule detection with the size greater than 3mm diameter and has a chance to become cancerous.

The modules of the work are as follows:

- Image Extraction
- Pre-processing
- Segmentation
- Feature Extraction
- Prediction

3.3.1 Image Extraction

The dataset contains two different types of files which are .raw and .mhd files. These raw file are converted to usable format.

3.3.2 Semantic Segmentation

The data set contains the annotation file where the id of the patient, coordinates of nodule and the nodule size will be annotated. Using this coordinates and diameter of the nodule mask is created. The image and the mask is given into the semantic segmentation algorithm to segment the nodule (region of interest). The segmentation algorithm is evaluated used IOU, precision, recall, dice coefficient, accuracy and loss.

3.3.3 Feature Extraction

The segmented images are given into the convolutional neural network model for the purpose of feature extraction.

3.3.4 Prediction

The test images are fed into the model for the prediction of nodule. This model is then evaluated with parameters such as accuracy, f-measure and dice coefficients.

CHAPTER 4
SYSTEM REQUIREMENTS

The hardware and software requirements needed for implementing the proposed system is discussed in this Chapter.

4.1 HARDWARE REQUIREMENTS

- Processor: Intel i5-6500 CPU or Higher
- System Type: 64 bit Operating System
- RAM: 4 GB or Higher

4.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 8.1
- Coding Language: Python 3.0
- Platform: Anaconda/Jupyter Notebook/Google Colab

4.3 SOFTWARE DESCRIPTION

4.3.1 GOOGLE COLABORATORY

Google Colab is free service and now supports free GPU. Colab can develop the language coding capabilities for Python programming. The application of common libraries like Keras, TensorFlow, PyTorch, and OpenCV develops in deep learning. Colab offers GPU, which is entirely free, which separates it from other free cloud services.

The Colaboratory was founded by Google to encourage Machine Learning and Research Learning. It is a notebook system for Jupyter that does not require installation and is completely running in the cloud. Colaboratory is a free Jupyter notebook environment that stores in Google Drive's notebooks. As part of Project Jupyter, the Colaboratory began, but Google ultimately took over the development. As , only kernels Python 2 and Python 3 are supported by Colaboratory and other Jupyter kernels Julia and R are not supported.

The whole experiment is conducted in colab platform. In colab GPU has been utilised for fast processing. The available GPU in colab are **Nvidia** K80s, T4s, P4s and P100s. The GPU are free to use. This GPU makes the process to be fast. The reason for choosing colab as a working environment is it does not need the system requirement. It is easy to load the data in the drive and can work with huge amount of data.

4.3.2 KERAS

Keras is a Python-based, high-level Network API that runs on TensorFlow, Keras, CNTK or Theano. It was built with a focus on enabling quick experimentation. It is a key for doing good research to provide the result with the minimum possible delay. Keras includes many implementations of widely used building blocks in neural networks such as layers, objectives activation functions, optimizers, and many methods to promote working with image and text data

- Enables easy and quick prototyping (via User friendliness, Modularity, and Extensibility)
- Supports both Convolutional and Recurrent networks, and also both combinations
- Runs on CPU and GPU seamlessly

4.4 Package DESCRIPTION

- SimpleITK fro rendering the from .raw file
- OpenCV for manipulating the image
- MatPlotLib for visualization of the image files and various presentation purposes.
- pandas (used for reading operations on csv files)
- Skimage for pre-processing
- Numpy nNumpy contains multidimensional matrix structures.
- ImageDataGenerator used for reshaping the dataset
- Tensorflow is an open source for numerical computations, machine learning mechanism. Tensor flow contains the all the packages for crating the model, compiling the model, fitting and predicting the model.

CHAPTER 5

IMPLEMENTATION AND RESULT ANALYSIS

This chapter shows the system design of lung cancer detection process

5.1 SYSTEM DESIGN OF PROPOSED METHODOLOGY

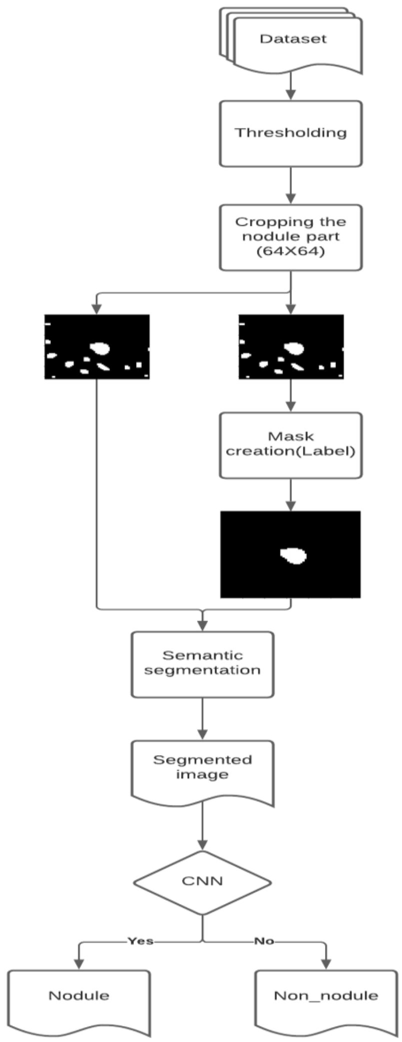


Fig 5.1 Working Model

5.2 DATASET DESCRIPTION

- LUNA - **LU**ng **N**odule **A**nalysis - CT images and annotations

- Number of CT scans- 888
- Annotated by **four** radiologists
- **.mhd -** Meta Image header
- **.raw –** Compressed CT scan
- **.csv** – contains the annotated data

Images

The complete dataset is divided into 10 subsets that should be used for the 10-fold cross-validation. All subsets are available as compressed zip files. In each subset, CT images are stored in MetaImage (mhd/raw) format. Each .mhd file is stored with a separate .raw binary file for the pixeldata.

Annotations

The annotation file is a csv file that contains one finding per line. Each line holds the SeriesInstanceUID of the scan, the x, y, and z position of each finding in world coordinates; and the corresponding diameter in mm. The annotation file contains 1186 nodules.

Candidates

The candidates file is a csv file that contains nodule candidate per line. Each line holds the scan name, the x, y, and z position of each candidate in world coordinates, and the corresponding class. The list of candidates is provided for reduction of ‘false positive reduction’ track.

5.3 Dataset Rendering

5.3.1 Annotated file Rendering

First annotated file is parsed using pandas. It contains seriesuid which means user identification number. CoordX, coordY represents the x and y coordinates of the nodule. CoordZ represents the slice number of the nodule. The diameter of the nodule is also provided in the csv file.

	seriesuid	coordX	coordY	coordZ	diameter_mm
0	1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222...	-128.699421	-175.319272	-298.387506	5.651471
1	1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222...	103.783651	-211.925149	-227.121250	4.224708
2	1.3.6.1.4.1.14519.5.2.1.6279.6001.100398138793...	69.639017	-140.944586	876.374496	5.786348
3	1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016...	-24.013824	192.102405	-391.081276	8.143262
4	1.3.6.1.4.1.14519.5.2.1.6279.6001.100621383016...	2.441547	172.464881	-405.493732	18.545150

Fig 5.2 Annotated file

5.3.2 Header file extraction:

The header file is extracted using SimpleITK package. The header contains

- RTTI gives the run-time type information about the image
- PipelineMTime
- UpdateMtime
- Dimension
- Size
- Spacing
- Origin

```
Image (0x1cf88c00)
  RTTI typeinfo:   itk::Image<float, 3u>
  Reference Count: 1
  Modified Time: 1610
  Debug: Off
  Object Name:
  Observers:
    none
  Source: (none)
  Source output name: (none)
  Release Data: Off
  Data Released: False
  Global Release Data: Off
  PipelineMTime: 1583
  UpdateMTime: 1609
  RealTimeStamp: 0 seconds
  LargestPossibleRegion:
    Dimension: 3
    Index: [0, 0, 0]
    Size: [512, 512, 121]
  BufferedRegion:
    Dimension: 3
    Index: [0, 0, 0]
    Size: [512, 512, 121]
  RequestedRegion:
    Dimension: 3
    Index: [0, 0, 0]
    Size: [512, 512, 121]
  Spacing: [0.761719, 0.761719, 2.5]
  Origin: [-198.1, -195, -335.21]
```

Fig 5.3 Header file

5.3.3 Image Extraction

The images is rendered from the raw and mhd file using SimpleITK

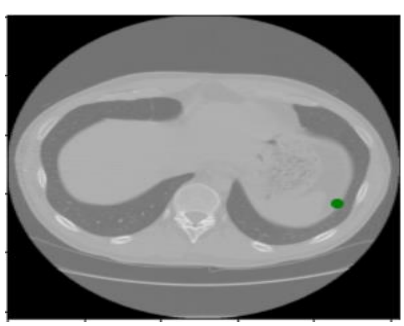


Fig 5.4 Lung Image

5.3.4 Converting Global to voxel coordinates

The header file contain the origin and spacing coordinates. The annotation file contains the nodule coordinates. To make the coordinates to synchronize global coordinates are converted to voxel coordinates.

```
x1 = np.abs(x-origin[0])/spacing[0]
x2 = np.abs(y-origin[1])/spacing[1]
x3 = np.abs(z-origin[2])/spacing[2]
```

Fig 5.5 Global to voxel coordinate conversion

5.3.5 Nodule Plotting

Using the nodule coordinated and the image the nodule is plotted in the lung image.

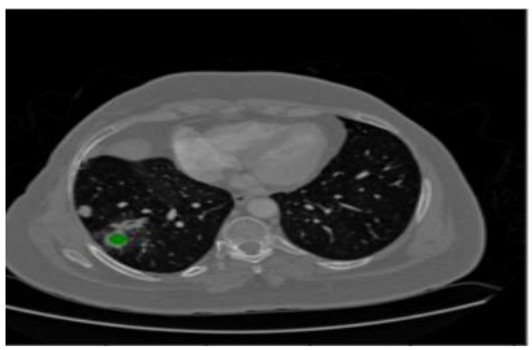


Fig 5.6 Nodule Plotting

5.4 Thresholding

Hounsfield units (HU) are a dimensionless unit universally used in computed tomography (CT) scanning to express CT numbers in a standardized and convenient form. Hounsfield units are obtained from a linear transformation of the measured attenuation coefficients. Generally this will be in negative values for bones. This region is not needed for the process. So bone images are eliminated using thresholding**.** Only Region of interest is extracted using thresholding.

Fig 5.7 Image Thresholding

5.5 Segmentation

5.5.1 Context aware U-net

The model used for segmentation is context aware U-net. This context aware U-net has expansion path and contraction path. The expansion path helps decreasing the spatial information and increasing the feature information. In contraction path both decreased spatial information and increased feature information are combined together. The down sampling allows for neglecting fine details and focusing on contextual information. The down sampled input is fed into a parallel, equivalent sequence of layers. The two parallel downward paths are realized as duplicates (i.e. with no shared parameters) in order to enable distinctive feature encodings for the context and original patch. The output of the encoding paths - i.e. bottom level - are concatenated and fed through 2 fully connected layers realized by 1x1x1 convolution followed by ReLU. Finally, the residuals of each level of both encoding paths are concatenated to the input of the corresponding decoding level to facilitate the contribution of spatial and context information in the final prediction map.

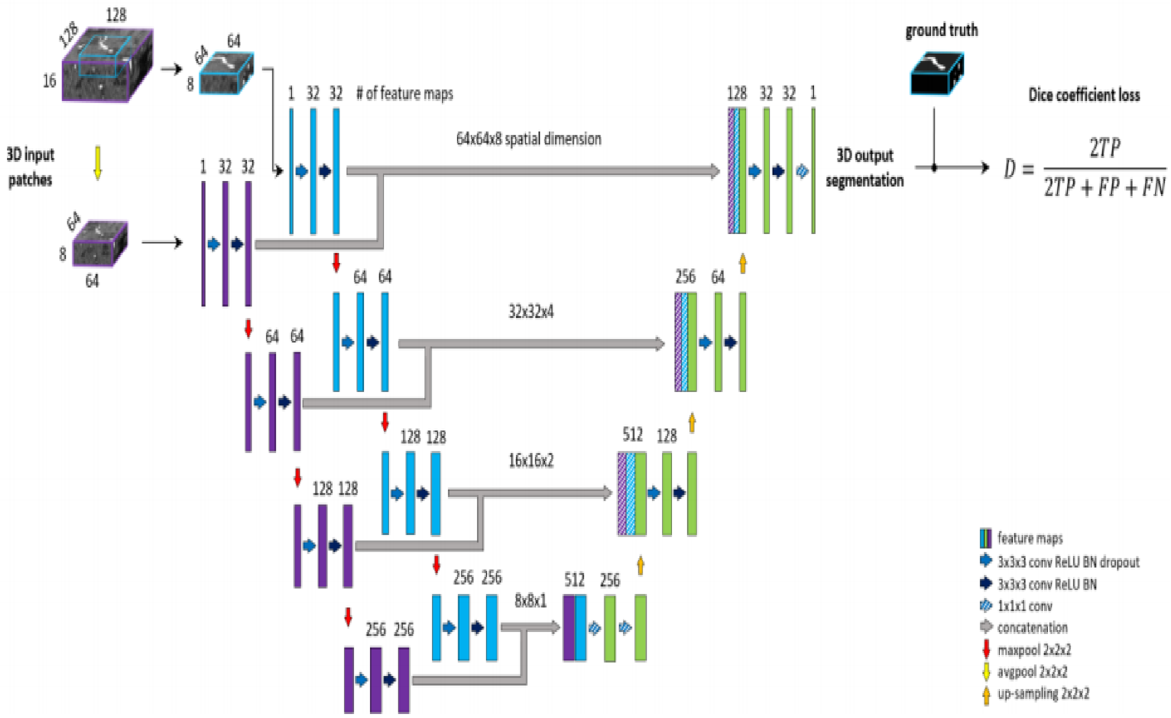


Fig 5.8 Context aware U-net

5.5.2 Model Creation for context aware U-net

Context aware U-net is created with convolutional neural network as the backbone. Fig 5.8 shows the summary of the model where there are 35,31,521 and 35,26,785 total and trainable parameter respectively.

conv2d_235 (Conv2D)	(None, 16, 16, 128)	442496	concatenate_72[0][0]
batch_normalization_236 (BatchN	(None, 16, 16, 128)	512	conv2d_235[0][0]
dropout_107 (Dropout)	(None, 16, 16, 128)	0	batch_normalization_236[0][0]
conv2d_236 (Conv2D)	(None, 16, 16, 128)	147584	dropout_107[0][0]
batch_normalization_237 (BatchN	(None, 16, 16, 128)	512	conv2d_236[0][0]
concatenate_73 (Concatenate)	(None, 32, 32, 128)	0	batch_normalization_231[0][0] batch_normalization_223[0][0]
conv2d_transpose_31 (Conv2DTran	(None, 32, 32, 64)	73792	batch_normalization_237[0][0]
concatenate_74 (Concatenate)	(None, 32, 32, 192)	0	concatenate_73[0][0] conv2d_transpose_31[0][0]
conv2d_237 (Conv2D)	(None, 32, 32, 64)	110656	concatenate_74[0][0]
batch_normalization_238 (BatchN	(None, 32, 32, 64)	256	conv2d_237[0][0]
dropout_108 (Dropout)	(None, 32, 32, 64)	0	batch_normalization_238[0][0]
conv2d_238 (Conv2D)	(None, 32, 32, 64)	36928	dropout_108[0][0]
batch_normalization_239 (BatchN	(None, 32, 32, 64)	256	conv2d_238[0][0]
concatenate_75 (Concatenate)	(None, 64, 64, 64)	0	batch_normalization_229[0][0] batch_normalization_221[0][0]
conv2d_transpose_32 (Conv2DTran	(None, 64, 64, 32)	18464	batch_normalization_239[0][0]
concatenate_76 (Concatenate)	(None, 64, 64, 96)	0	concatenate_75[0][0] conv2d_transpose_32[0][0]
conv2d_239 (Conv2D)	(None, 64, 64, 32)	27680	concatenate_76[0][0]
batch_normalization_240 (BatchN	(None, 64, 64, 32)	128	conv2d_239[0][0]
dropout_109 (Dropout)	(None, 64, 64, 32)	0	batch_normalization_240[0][0]
conv2d_240 (Conv2D)	(None, 64, 64, 32)	9248	dropout_109[0][0]
batch_normalization_241 (BatchN	(None, 64, 64, 32)	128	conv2d_240[0][0]
conv2d_241 (Conv2D)	(None, 64, 64, 1)	33	batch_normalization_241[0][0]

Total params: 3,531,521
Trainable params: 3,526,785
Non-trainable params: 4,736

Fig 5.9 Context aware U-net summary report

5.5.3 Training the model

Using the coordinated and the diameter a mask is created. The mask and the image cropped in the size of 64X64 are given into the semantic segmentation algorithm.

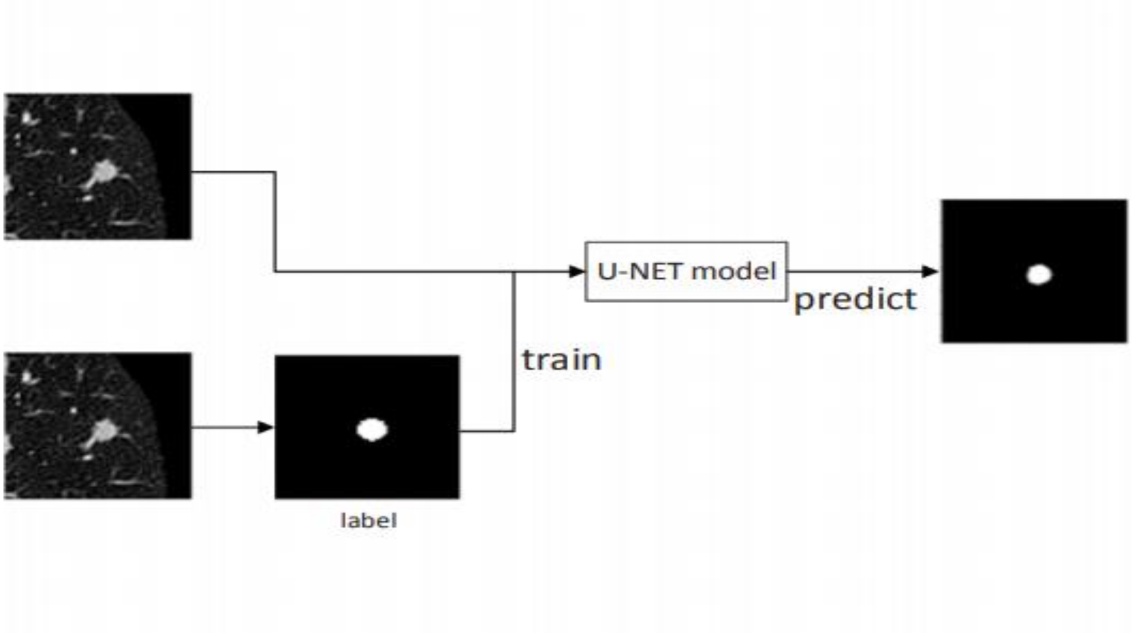


Fig 5.10 Segmentation architecture

Using the label and the image, segmented image is produced as an output as shown in the Fig 5.10.

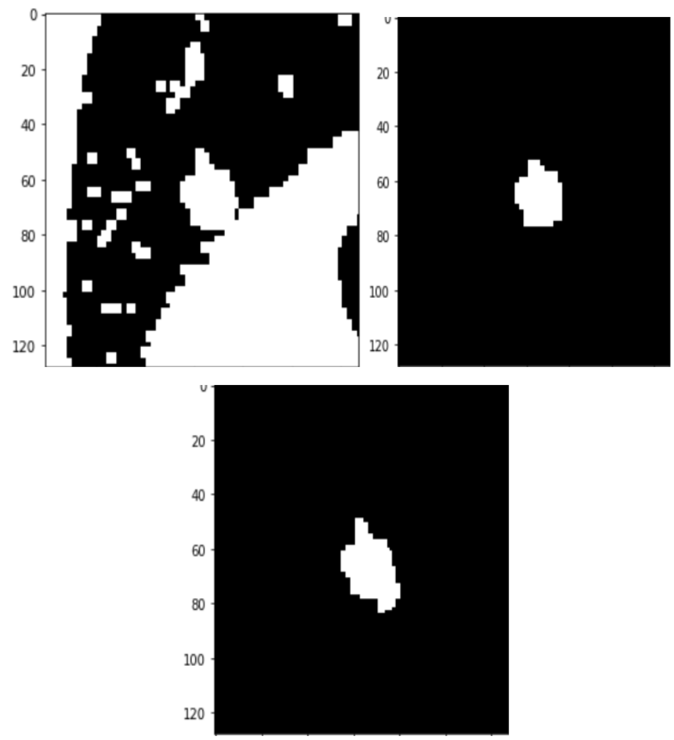


Fig 5.11 Image, Label and Segmented Image

5.5.4 Evaluation of Context aware U-net

The model is evaluated using IOU, Dice Coefficient, Precision, Recall, Accuracy and loss.

Table 5.1 Comparison of results for different optimizer

Optimizer	IOU	Dice Coef	Precision	Recall	Accuracy	Loss
Adam	99.70	90.38	91.50	90.08	99.42	2.99
Adamax	99.55	83.79	97.42	73.78	99.14	5.24
Nadam	99.77	92.55	92.89	92.28	99.54	2.23
SGD	98.99	46.36	90.54	30.72	97.69	6.28
Adagrad	94.46	9.94	27.77	45.60	94.59	35.48
Adadelta	87.42	7.70	6.07	58.88	70.57	65.85
RMSprop	99.61	88.81	90.41	86.89	99.31	5.81

5.6 Feature Extraction

5.6.1 Model Creation for classification

CNN model is created using three convolutional layers and pooling layer as shown in Fig 5.12.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_3 (Conv2D)            (None, 98, 98, 16)        448
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 49, 49, 16)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 47, 47, 32)        4640
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 23, 23, 32)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 21, 21, 64)        18496
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 10, 10, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6400)              0
_________________________________________________________________
dense_2 (Dense)              (None, 512)               3277312
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,301,409
Trainable params: 3,301,409
Non-trainable params: 0
```

Fig 5.12 Convolutional Neural Network summary report

5.6.2 Model Fitting

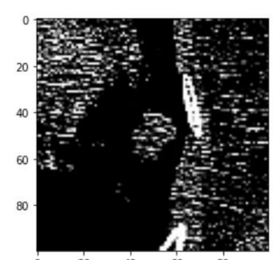
```
Epoch 16/30
25/25 [==============================] - 2s 73ms/step - loss: 0.2594 - accuracy: 0.9067
Epoch 17/30
25/25 [==============================] - 2s 83ms/step - loss: 0.4089 - accuracy: 0.8267
Epoch 18/30
25/25 [==============================] - 3s 108ms/step - loss: 0.4669 - accuracy: 0.8133
Epoch 19/30
25/25 [==============================] - 3s 105ms/step - loss: 0.4152 - accuracy: 0.8514
Epoch 20/30
25/25 [==============================] - 1s 28ms/step - loss: 0.4039 - accuracy: 0.8667
Epoch 21/30
25/25 [==============================] - 1s 26ms/step - loss: 0.2238 - accuracy: 0.8919
Epoch 22/30
25/25 [==============================] - 1s 56ms/step - loss: 0.2899 - accuracy: 0.8933
Epoch 23/30
25/25 [==============================] - 1s 45ms/step - loss: 0.1814 - accuracy: 0.9467
Epoch 24/30
25/25 [==============================] - 0s 8ms/step - loss: 0.2359 - accuracy: 0.9189
Epoch 25/30
25/25 [==============================] - 1s 27ms/step - loss: 0.1698 - accuracy: 0.9467
Epoch 26/30
25/25 [==============================] - 1s 28ms/step - loss: 0.1694 - accuracy: 0.9459
Epoch 27/30
25/25 [==============================] - 0s 8ms/step - loss: 0.1957 - accuracy: 0.9333
Epoch 28/30
25/25 [==============================] - 1s 34ms/step - loss: 0.1447 - accuracy: 0.9600
Epoch 29/30
25/25 [==============================] - 0s 8ms/step - loss: 0.1881 - accuracy: 0.9600
```

Fig 5.13 CNN Training Dataset Accuracy

In Figure 5.13, the CNN model is used and the resultant accuracy for the training dataset is 100%. The experiment trails at run for 10 epochs. If the value of the epoch is increased the accuracy performance will also increase.

5.7 Prediction

Any image is given inside the created model and predict whether it is nodule or non_nodule.




```
[[0.]]
nodule
```

Fig 5.14 Predicted nodule

5.8 Evaluation Metrics

Evaluation metrics selects three criteria to test the CNN model: Precision, recall and F-Measures. True Positive (TP), False Positive FP, False Negative (FN) and True Negative (TN) are four parameters for the calculation of precision, recall and F-Measures. The TP is the number of nodule class samples taken into the nodule class. TP is completely optimistic. FP is the number of non-nodules samples entered in the nodule class .FN is the number of nodule samples classified into the non-nodule class. TN is the number of non-nodule samples in the non-nodule class.

5.8.1 Precision

- Precision is a metric that quantifies the number of correct positive predictions made.
- Precision, therefore, calculates the accuracy for the minority class.
- It is calculated as the ratio of correctly predicted positive examples divided by the total number of positive examples that were predicted.

$$\text{Precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{5.1}$$

5.8.2 Recall

- Recall is a metric that quantifies the number of correct positive predictions made out of all positive predictions that could have been made.
- Unlike precision that only comments on the correct positive predictions out of all positive predictions, recall provides an indication of missed positive predictions.
- In this way, recall provides some notion of the coverage of the positive class.

$$\text{Recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{5.2}$$

5.8.3 F-Measure

F-Measure provides a way to combine both precision and recall into a single measure that captures both properties.

$$\text{F-Measure}=\frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \tag{5.3}$$

5.9 Performance Evaluation

5.9.1 ROC Curve

The ROC curve is plotted against the test and validation dataset. As a result 0.94 area has been obtained as shown in Fig 5.15. The ROC value lies between 0 to 1 where 0 denotes a bad classifier and 1 denotes **an excellent** classifier. As the result obtained is 94% the classification is good.

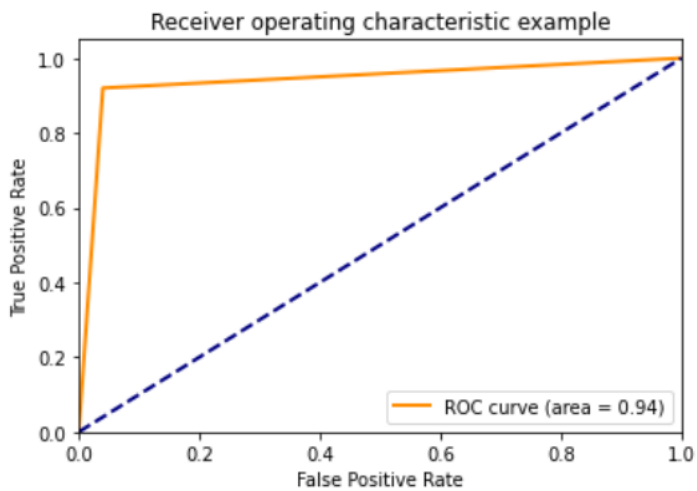


Fig 5.15 ROC Curve

5.9.2 Confusion matrix

A **confusion matrix** is a technique for summarizing the performance of a classification algorithm. Confusion matrix is plotted for the predicted output and presented in Fig 5.16 where there are 24 nodules and 24 non nodule has been predicted correct, 2 non nodules have been predicted as nodules and no nodule has been predicted as non-nodule.

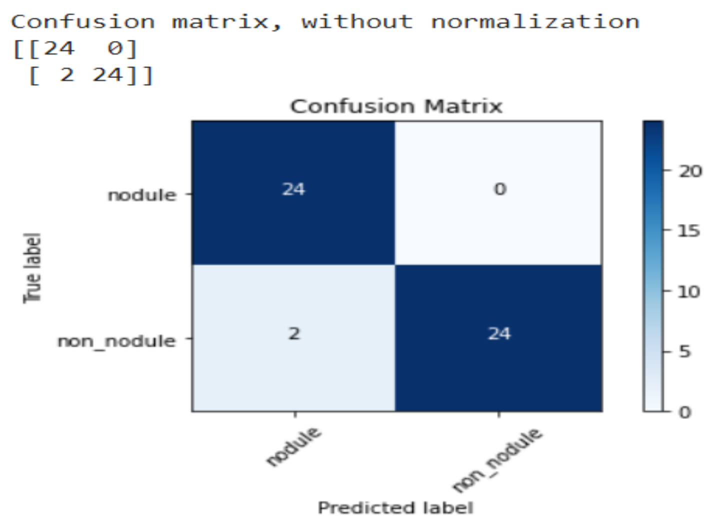


Fig 5.16 Confusion matrix

5.9.3 Precision, Recall and F-measure

From Fig 5.18 it is observed that the final precision, recall and f-measure value obtained are 95%, 92% and 93% respectively. The accuracy is 95%. Out of 50 testing images there is no true negative images and there is only two false positives. The loss of the classification method is about 10%.

Table 5.2 Performance measure

Sl.No	Parameters	Values
1	Testing images	50
2	True Possitive	24
3	True Negative	0
4	False Possitive	2
5	False Negative	24
6	Precision	95
7	Recall	92

8	**F-measure**	**93**
9	**Accuracy**	**95**
10	**Loss**	**10**

CONCLUSION AND FUTURE WORK

In the proposed work, the CAD system uses Convolutional Neural Network (CNN), an advanced deep learning technique on the LUNA dataset. Deep learning based algorithm helps in automatic detection of nodules. Non nodule annotations are also provided in LUNA data set for the reduction of false positive rate. At first the image is rendered and undergone thresholding process for elimination of bone parts. Next a mask is created for the nodule using the diameter given in annotated file. The image and the mask is given into context aware U-net as an input for the purpose of segmentation of nodule. While evaluating the segmentation architecture 99.7% IOU, 90.38% Dice Coefficient, 91.50% Precision, 90.08% Recall, 99.42% accuracy and 2.99% loss has been obtained. The segmented nodule and non-nodule are trained using Convolutional neural network architecture for classification. When predicting the test data set 95% precision, 92% recall and 93% F-measure has been reached. The accuracy is 95% because the data set is lesser in number which can reduce the precision rate. The future work includes adding more data set for training to increase the accuracy rate.

PSG COLLEGE OF TECHNOLOGY
Department of Information Technology

VIRTUAL NATIONAL CONFERENCE ON
COMPUTATIONAL AND INFORMATION SCIENCE
NCCIS - 2021

This is to certify that

Manjari P

PG Student, Department of IT, PSG College of Technology

has presented a paper titled

"Lung Cancer Nodule Detection Using Deep Learing Algorithm "

in the Virtual National Conference on Computational and Information Science - NCCIS 2021
organized by Department of IT, PSG College of Technology held during 23 - 24, April 2021

HoD - IT

BIBLIOGRAPHY

[1] Ahmed, T., Parvin, M. S., Haque, M. R., & Uddin, M. S. (2020). Lung cancer detection using CT image based on 3D convolutional neural network. *Journal of Computer and Communications*, *8*(03), 35.

[2] Alakwaa, W., Nassef, M., & Badr, A. (2017). Lung cancer detection and classification with 3D convolutional neural network (3D-CNN). *Lung Cancer*, *8*(8), 409.

[3] Harsono, I. W. (2019). Lung Nodule Texture Detection and Classification Using 3D CNN. *CommIT (Communication and Information Technology) Journal*, *13*(2), 91-103.

[4] Vas, M., & Dessai, A. (2017, August). Lung cancer detection system using lung CT image processing. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)* (pp. 1-5). IEEE.

[5] Huang, X., Shan, J., & Vaidya, V. (2017, April). Lung nodule detection in CT using 3D convolutional neural networks. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)* (pp. 379-383). IEEE.

[6] Bhalerao, R. Y., Jani, H. P., Gaitonde, R. K., & Raut, V. (2019, March). A novel approach for detection of lung cancer using digital image processing and convolution neural networks. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)* (pp. 577-583). IEEE.

[7] Ali, I., Hart, G. R., Gunabushanam, G., Liang, Y., Muhammad, W., Nartowt, B., ... & Deng, J. (2018). Lung nodule detection via deep reinforcement learning. *Frontiers in oncology*, *8*, 108.

[8] Cao, H., Liu, H., Song, E., Ma, G., Xu, X., Jin, R., ... & Hung, C. C. (2020). A two-stage convolutional neural networks for lung nodule detection. *IEEE journal of biomedical and health informatics*, *24*(7), 2006-2015.

[9] Veronica, B. K. (2020). An effective neural network model for lung nodule detection in CT images with optimal fuzzy model. *Multimedia Tools and Applications*, 1-21.

[10] Vas, M., & Dessai, A. (2017, August). Lung cancer detection system using lung CT image processing. In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)* (pp. 1-5). IEEE.

[11] Bhalerao, R. Y., Jani, H. P., Gaitonde, R. K., & Raut, V. (2019, March). A novel approach for detection of lung cancer using digital image processing and convolution neural networks. In *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)* (pp. 577-583). IEEE.

[12] Su, Y., Li, D., & Chen, X. (2021). Lung Nodule Detection based on Faster R-CNN Framework. *Computer Methods and Programs in Biomedicine*, *200*, 105866.