

CLASSIFICATION OF LUNG CANCER DETECTION USING CONVOLUTION NEURAL NETWORK (CNN)

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering
by

Nagishetty Vishal (38110347)

Kalisetti Sai Dinesh (38110225)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH - 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Nagishetty Vishal (38110347) Kalisetti Sai Dinesh (38110225)** who carried out the project entitled **“Classification of lung cancer detection using Convolution Neural Network (CNN)”** under our supervision from November 2019 to April 2020.

Internal Guide

Dr.J.Albert Mayan, M.E., Ph.D

Head of the Department

Dr.L.Lakshmanan.,M.E.,Ph.D.,

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

We **Nagishetty Vishal (38110347)** **Kalisetti Sai Dinesh (38110225)** hereby declare that the Project Report entitled “**Classification of Lung Cancer Detection using Convolution Neural Network (CNN)**” is done by me under the guidance of Assistant Professor **Dr.J.Albert Mayan, M.E., Ph.D** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Head of the Department**, Dept. of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.J.Albert Mayan, M.E., Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Early identification of lung nodules is right now the one of the most effective approaches to foresee and treat lung disease. Accordingly, the previous decade has seen a great deal of spotlight on computer aided diagnosis (CAD) of lung nodules , whose objective is to productively identify, portion lung nodules and arrange them as being generous or harmful. Powerful recognition of such nodules stays a test because of their intervention fit as a fiddle, size and surface. This paper plan to recognize and group the lung malignant growth utilizing CNN calculation. This paper introduces a methodology which utilizes a Convolutional Neural Network (CNNs) to order tumors found in lung disease screening figured tomography filters as threatening or amiable. CNNs have extraordinary properties, for example, spatial invariance, and take into consideration different component extraction. As a progressively mechanized methodology, the CNN technique utilizes crude picture information as information and can be straightforwardly classified as yield. The system can delineate picture of lung knob competitor into attributes of various goals and scales while experiencing the trouble in qualities portrayal of pneumonic knob for its radiological heterogeneity and fluctuation in sizes and shapes, in this way enormously diminishing the bogus positive of classification task and improving the scores on classification measurements.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	i
	LIST OF ABBREVIATIONS	v
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	2
3.	SYSTEM METHODOLOGY	8
	3.1 OBJECTIVE	8
	3.2 EXISTING SYSTEM	9
	3.2.1 LIMITATIONS OF EXISTING SYSTEM	9
	3.3 PROPOSED SYSTEM	9
	3.3.1 ADVANTAGES OF PROPOSED SYSTEM	10
	3.4 SYSTEM REQUIREMENTS	10
	3.4.1 HARDWARE REQUIREMENTS	10
	3.4.2 SOFTWARE REQUIREMENTS	11
	3.4.3 TECHNOLOGIES USED	11
	3.5 INTRODUCTION TO PYTHON	11
	3.5.1 WHAT CAN PYTHON DO?	11
	3.5.2 WHY PYTHON?	12
	3.5.3 PYTHON SYNTAX COMPARED TO OTHER PROGRAMMING LANGUAGES	13
	3.5.4 PYTHON IS INTERPRETED	13
	INTRODUCTION TO DEEP LEARNING	14
	HUMAN BRAIN Vs. ARTIFICIAL NEURAL NETWORKS	15

ARTIFICIAL NEURAL NETWORK	16
PROCESS	
INTRODUCTION TO CNN	17
THE PROBLEM SPACE	18
BIOLOGICAL CONNECTION	19
STRUCTURE	20
FIRST LAYER – MATH PART	20
FIRST LAYER-HIGH LEVEL	22
PERSPECTIVE	
FULLY CONNECTED LAYER	24
TRAINING	25
DESING AND IMPLEMENTATION CONSTRAINT	28
CONSTRAINT IN ANALYSIS	28
CONSTRAINT IN DESIGN	28
CONSTRAINT IN IMPLEMENTATION	29
NON FUNCTIONAL REQUIREMENTS	29
PERFORMANCE REQUIREMENTS	29
SAFETY REQUIREMENTS	30
ARCHITECTURE DIAGRAM	30
SYSTEM DESIGN	31
UML DIAGRAMS	31
SEQUENCE DIAGRAM	31
USECASE DIAGRAM	32
ACTIVITY DIAGRAM	33
COLLABORATION DIAGRAM	34
MODULES	35
PREPROCESSING	35
CONVOLUTION NEURAL NETWORK	35
CLASSIFICATION	35
PERFORMANCE MEASURE	35
CODING	36

	TESTING	37
	WHITE BOX TESTING	38
	BLACK BOX TESTING	38
	SOFTWARE TESTING STRATEGIES	39
	INTEGRATION TESTING	39
	PROGRAM TESTING	39
	SECURITY TESTING	40
	VALIDATION TESTING	40
	USER ACCEPTANCE TESTING	41
4.	RESULTS AND DISCUSSION	42
5.	CONCLUSION AND FUTURE WORK	45
	CONCLUSION	45
	FUTURE SCOPE	46
	REFERENCES	47
	APPENDIX	49
	1. SOURCE CODE	49
	2. SCREENSHOTS	52
	3. PAPER PUBLICATION	56
	4. PAPER SUBMISSION	66
	5. PLAGIARISM REPORT	67

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANSION

ACCP	American College of Chest Physicians
AJCC	American Joint Committee on Cancer
CHART	Continuous Hyper Fractionated Accelerated Radiotherapy
CT	Computerized tomography
CXR	Chest X-Ray
CI	Confidence Interval
DA	Decision Aid
GDG	Guideline Development Group
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
SEER	Surveillance Epidemiology and End Results
PET	Positron Emission Tomography

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
3.1	BLOCK DIAGRAM OF PROPOSED SYSTEM	10
3.2	MULTIPLE LAYERS OF ARTIFICIAL NEURAL NETWORK	15
	PROCESS OF ARTIFICIAL NEURAL NETWORKS	16
	IMAGE CLASSIFICATION	18
	VISUALIZATION OF 5X5 FILTER CONVOLUTION AROUND AN INPUT VOLUME	21
	PIXEL REPRESENTATION OF FILTER	22
	VISUALIZATION OF FILTER ON THE IMAGE	23
	A FULL CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE DIAGRAM	25
	SEQUENCE DIAGRAM	30
	USECASE DIAGRAM	32
	ACTIVITY DIAGRAM	33
	COLLABORATION DIAGRAM	34
	SAMPLE IMAGE OF LUNG FOR DETECTION	42
	OUTPUT FOR CANCEROUS IMAGE	43

LIST OF TABLES

TABLE NO.	TABLE NAME	PAGE NO.
4.1	FFC LAYER WITH COMPUTE THE CLASS SCORE	43
4.2	CONFUSION MATRIX	44
5.1	COMPARISON OF RESULTS WITH THE PREVIOUS WORKS	45

CHAPTER 1

INTRODUCTION

Lung cancer is one kind of cancer among those which have caused the most deaths. Computer aided diagnosis systems (CADs) can make use of the knowledge of more than one experienced doctor and existing resources to assist radiologists for making medical diagnostics. Convolutional neural network (CNN) to extract features of various levels and resolutions from different depth layers in the network for classification of lung nodule candidates. In this proposed system the method using convolution neural networks (CNN) is used. As a more automated approach, the CNN method uses raw image data as input and can be directly classified as output. A single CNN (Convolution neural network) model is more convenient to train and use than that of the machine learning algorithm

Lung cancer is one of the most registered and common kinds of illness globally, both in the volume of new cases and in the amount of deaths. 1,8 million new instances of lung unsafe production were tested worldwide in 2015. About 225,000 new cases of cancer were predicted in 2017, and 155,870 deaths are reported in the United States. The average lung cancer survival levels are estimated at 65 per cent. Although there are various forms of operation, lung cancer therapy treatments have been used. CT research is maybe the most appropriate way to continue the lung nodules time region because of its potential to produce 3D chest photos[1]. The key aim of this paper is to establish a CAD method for the early identification of lung cancer based on the examination of lung regions like chest CT images.

The technique of CAD on lung harmful development all around involves an acknowledgment structures CAD and a CAD. The aim of CAD is to segment the pneumonic handle patients as knobs though CAD targets arranging the recognized knobs as liberal or perilous knobs [2]. Nowadays, effective learning is considered as maybe the best response for certain issues of PC vision and model affirmation, for instance, picture examination, talk affirmation, trademark language getting ready.

CHAPTER 2

LITERATURE SURVEY

TITLE 1: Lung Cancer Detection Using Image Segmentation by means of Various Evolutionary Algorithms

Authors: K. Senthil Kumar, K. Venkatalakshmi, and K. Karthikeyan, Published in 2019

The objective of this paper is to explore an expedient image segmentation algorithm for medical images to curtail the physicians' interpretation of computer tomography (CT) scan images. Modern medical imaging modalities generate large images that are extremely grim to analyze manually. The consequences of segmentation algorithms rely on the exactitude and convergence time. At this moment, there is a compelling necessity to explore and implement new evolutionary algorithms to solve the problems associated with medical image segmentation. Lung cancer is the frequently diagnosed cancer across the world among men. Early detection of lung cancer navigates towards apposite treatment to save human lives. CT is one of the modest medical imaging methods to diagnose the lung cancer. In the present study, the performance of five optimization algorithms, namely, *k*-means clustering, *k*-median clustering, particle swarm optimization, inertia-weighted particle swarm optimization, and guaranteed convergence particle swarm optimization (GCPSO), to extract the tumor from the lung image has been implemented and analyzed. The performance of median, adaptive median, and average filters in the preprocessing stage was compared, and it was proved that the adaptive median filter is most suitable for medical CT images. Furthermore, the image contrast is enhanced by using adaptive histogram equalization. The preprocessed image with improved quality is subject to four algorithms. The practical results are verified for 20 sample images of the lung using MATLAB, and it was observed that the GCPSO has the highest accuracy of 95.89%.

TITLE 2: Lung Cancer Detection and Classification Using Deep CNN

Authors: S. Sasikala, M. Bharathi, B. R. Sowmiya, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8 Issue-2S December, 2018

Lung cancer is one of the most killer diseases in the developing countries and the detection of the cancer at the early stage is a challenge. Analysis and cure of lung malignancy have been one of the greatest difficulties faced by humans over the most recent couple of decades. Early identification of tumor would facilitate in sparing a huge number of lives over the globe consistently. This paper presents an approach which utilizes a Convolutional Neural Network (CNN) to classify the tumors found in lung as malignant or benign. The accuracy obtained by means of CNN is 96%, which is more efficient when compared to accuracy obtained by the traditional neural network systems.

TITLE 3: Deep Residual Learning for Image Recognition

Authors: Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the Image Net dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the Image Net test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are

foundations of our submissions to ILSVRC & COCO 2015 competitions¹, where we also won the 1st places on the tasks of ImageNet detection, Image Net localization, COCO detection, and COCO segmentation.

TITLE 4: Lung Nodule Classification Using Deep Features In CT Images

Authors: Kumar, Devinder & Wong, Alexander & Clausi, David, 12th Conference on Computer and Robot Vision, 2015.

Early detection of lung cancer can help in a sharp decrease in the lung cancer mortality rate, which accounts for more than 17% percent of the total cancer related deaths. A large number of cases are encountered by radiologists on a daily basis for initial diagnosis. Computer-aided diagnosis (CAD) systems can assist radiologists by offering a second opinion and making the whole process faster. We propose a CAD system which uses deep features extracted from an auto encoder to classify lung nodules as either malignant or benign. We use 4303 instances containing 4323 nodules from the National Cancer Institute (NCI) Lung Image Database Consortium (LIDC) dataset to obtain an overall accuracy of 75.01% with a sensitivity of 83.35% and false positive of 0.39/patient over a 10 fold cross validation

TITLE 5: Computer-Aided Detection of Lung Nodules using Outer Surface Features

Authors: Önder Demira, and Ali Y_Imaz Camurcu, Published 2015

In this paper, a computer-aided detection (CAD) system was developed for the detection of lung nodules in computed tomography images. The CAD system consists of four phases, including two-dimensional and three-dimensional preprocessing phases. In the feature extraction phase, four different groups of features are extracted from volume of interests: morphological features, statistical and histogram features, statistical and histogram features of outer surface, and texture features of outer surface. The support vector machine algorithm is optimized using particle swarm optimization for

classification. The CAD system provides 97.37% sensitivity, 86.38% selectivity, 88.97% accuracy and 2.7 false positive per scan using three groups of classification features.

TITLE 6: Very Deep Convolutional Networks for Large-Scale Image Recognition

Authors: Karen Simonyan Andrew Zisserman, Published as a conference paper at ICLR 2015

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our Image Net Challenge 2014 submission, where our team secured the first and the second places in the localization and classification tracks respectively. We also show that our representations generalize well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

TITLE 7: U-Net: Convolutional Networks for Biomedical Image Segmentation

Authors: Olaf Ronneberger, Philipp Fischer, and Thomas Brox, Published 2015

There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast

and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU.

TITLE 8: Image Net Classification with Deep Convolutional Neural Network

Authors: Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012.

In this paper, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way soft max. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce over fitting in the fully-connected layers we employed a recently-developed regularization method called “dropout” that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

TITLE 9: A Survey on Transfer Learning

Authors: Sinno Jialin Pan and Qiang Yang, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2009.

A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data

may be in a different feature space or follow a different data distribution. In such cases, knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data labeling efforts. In recent years, transfer learning has emerged as a new learning framework to address this problem. This survey focuses on categorizing and reviewing the current progress on transfer learning for classification, regression and clustering problems. In this survey, we discuss the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning and sample selection bias, as well as co-variate shift. We also explore some potential future issues in transfer learning research.

TITLE 10: Histograms of Oriented Gradients for Human Detection

Authors: Navneet Dalal and Bill Triggs, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005

In this paper, we study the question of feature sets for robust visual object recognition; adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

CHAPTER 3

SYSTEM METHODOLOGY

OBJECTIVE

Lung cancer is one kind of cancer among those which have caused the most deaths. Computer aided diagnosis systems (CADs) can make use of the knowledge of more than one experienced doctor and existing resources to assist radiologists for making medical diagnostics. Convolutional neural network to extract features of various levels and resolutions from different depth layers in the network for classification of lung nodule candidates. In this proposed system the method using convolution neural networks (CNN) is used. As a more automated approach, the CNN method uses raw image data as input and can be directly classified as output. A single CNN (Convolution neural network) model is more convenient to train and use than that of the machine learning algorithm

The objectives of this project include:

1. The Lung Image Database Consortium (LIDC) is developing a publicly available database of computed tomography (CT) scans as a medical imaging research resource to promote the development of computer-aided detection.
2. To obtain the best estimate of the location and spatial extent of lung nodules, expert thoracic radiologists reviewed and annotated each scan.
3. Computer aided diagnosis systems (CADs) can make use of the knowledge of more than one experienced doctor and existing resources to assist radiologists for making medical diagnostics.
4. Convolutional neural network to extract features of various levels and resolutions from different depth layers in the network for classification of lung nodule candidates.

5. As a more automated approach, the CNN method uses raw image data as input and can be directly classified as output. A single CNN model is more convenient to train and use than that of the machine learning algorithm.

EXISTING SYSTEM

The hand-crafted features are often used in traditional methods for nodule auto classification. These features may include those primary level features like the texture, shape and size of lung nodules. And it may also include the high level characteristics abstracted from primary level features such as small waves, Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) etc. The features used are self designated, lack of self learning ability like computers and the traditional methods are not intelligent enough to work in an end to end way.

LIMITATIONS OF EXISTING SYSTEM

- Time consuming process.
- High complexity on test case generation.
- Detection of lung cancer in CT Images has less accuracy rate.
- Lack of ability to identify the images of lung cancer.
- Manual test case generation is error-prone.

PROPOSED SYSTEM

In this proposed system the method using convolution neural networks (CNN) is used. As a more automated approach, the CNN method uses raw image data as input and can be directly classified as output. The network can map the image of lung nodule candidate into characteristics of different resolutions and scales while encountering the difficulty in characteristics description of pulmonary nodule for its radiological heterogeneity and variability in sizes and shapes, thus greatly reducing the false positive of classification task and improving the scores on classification metrics.

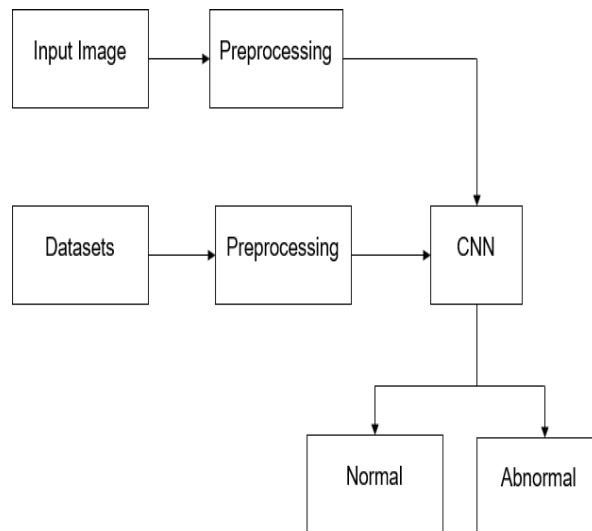


Fig.3.1 Blocked diagrams of proposed system

ADVANTAGES OF PROPOSED SYSTEM

- Automatic detection of Lung cancer in CT images has higher accuracy.
- It has a great improvement in performance.
- It uses raw input as data and directly classifies the output as images.
- It reduces the false positive of classification task and improve the scores on classification metrics

SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above

SOFTWARE REQUIREMENTS

- Operating System : Windows 7 , 8, 10 (64 bit)
- Software : Python 3.7
- Tools : Anaconda (Jupyter Note Book IDE)

TECHNOLOGIES USED

- CNN

INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

WHAT CAN PYTHON DO?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

WHY PYTHON?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End Of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of

which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

- It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

PYTHON SYNTAX COMPARED TO OTHER PROGRAMMING LANGUAGES

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

PYTHON IS INTERPRETED

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to

a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

INTRODUCTION TO DEEP LEARNING

Deep Learning is a specialized form of Machine Learning that uses supervised, unsupervised, or semi-supervised learning to learn from data representations.

It is similar to the structure and function of the human nervous system, where a complex network of interconnected computation units works in a coordinated fashion to process complex information.

Machine Learning is an approach or subset of Artificial Intelligence that is based on the idea that machines can be given access to data along with the ability to learn from it. Deep Learning takes Machine Learning to the next level.

There are many aspects of Deep Learning as listed below:

- Multiple levels of hierarchical representations
- Multi-layered neural networks
- Training of large neural networks
- Multiple non-linear transformations
- Pattern recognition

- Feature extraction
- High-level data abstractions model

HUMAN BRAIN VS. ARTIFICIAL NEURAL NETWORKS

The computational models in Deep Learning are loosely inspired by the human brain. The multiple layers of training are called Artificial Neural Networks (ANN).

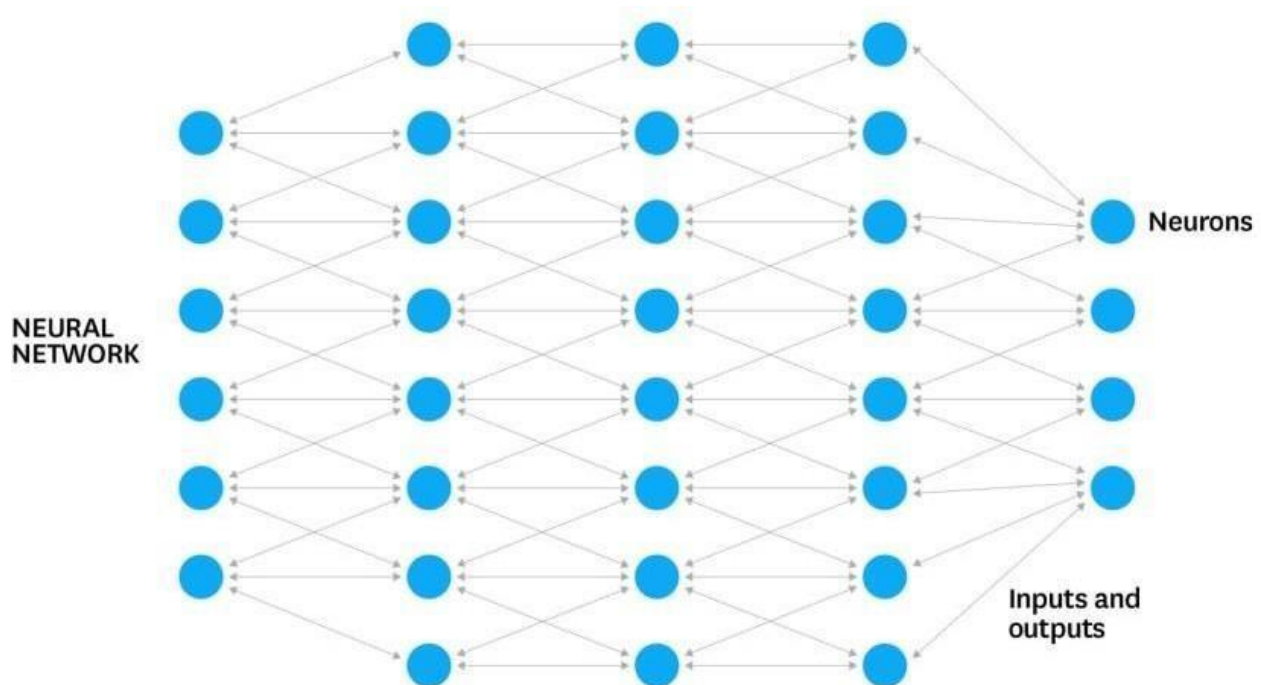


Fig.3.2 Multiple layers of Artificial Neural Networks

ANNs are processing devices (algorithms or actual hardware) that are modeled on the neuronal structure of the mammalian cerebral cortex but on a much smaller scale. It is a computing system made up of a number of simple, highly interconnected processing elements which process information through their dynamic state response to external inputs.

ARTIFICIAL NEURAL NETWORKS PROCESS

Artificial Neural Networks consist of the following four main parts:

- Neurons
- Nodes
- Input
- Output

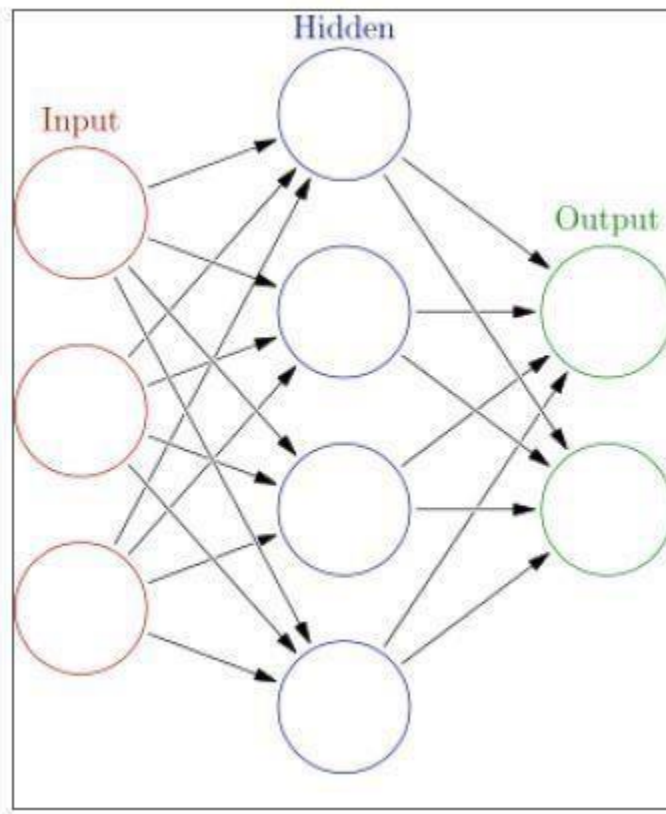


Fig.3.3 Process of Artificial Neural Networks

Neuron

Artificial Neural Networks contain layers of neurons. A neuron is a computational unit that calculates a piece of information based on weighted input parameters. Inputs accepted by the neuron are separately weighted.

Inputs are summed and passed through a non-linear function to produce output. Each layer of neurons detects some additional information, such as edges of things in a picture or tumors in a human body. Multiple layers of neurons can be used to detect additional information about input parameters.

Nodes

Artificial Neural Network is an interconnected group of nodes akin to the vast network of layers of neurons in a brain. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

Inputs

Inputs are passed into the first layer. Individual neurons receive the inputs, with each of them receiving a specific value. After this, an output is produced based on these values.

Outputs

The outputs from the first layer are then passed into the second layer to be processed. This continues until the final output is produced. The assumption is that the correct output is predefined.

Each time data is passed through the network, the end result is compared with the correct one, and tweaks are made to their values until the network creates the correct final output each time.

INTRODUCTION TO CNN

Convolutional neural networks (CNN) sounds like a weird combination of biology and math with a little CS sprinkled in, but these networks have been some of the most influential innovations in the field of computer vision. 2012 was the first year that neural nets grew to prominence as Alex Krizhevsky used them to win that year's ImageNet competition (basically, the annual Olympics of computer vision), dropping the

classification error record from 26% to 15%, an astounding improvement at the time. Ever since then, a host of companies have been using deep learning at the core of their services. Facebook uses neural nets for their automatic tagging algorithms, Google for their photo search, Amazon for their product recommendations, Pinterest for their home feed personalization, and Instagram for their search infrastructure.

However, the classic, and arguably most popular, use case of these networks is for image processing. Within image processing, let's take a look at how to use these CNNs for image classification.

THE PROBLEM SPACE

Image classification is the task of taking an input image and outputting a class (a cat, dog, etc) or a probability of classes that best describes the image. For humans, this task of recognition is one of the first skills we learn from the moment we are born and is one that comes naturally and effortlessly as adults. Without even thinking twice, we're able to quickly and seamlessly identify the environment we are in as well as the objects that surround us. When we see an image or just when we look at the world around us. These skills of being able to quickly recognize patterns, generalize from prior knowledge, and adapt to different image environments are ones that we do not share with our fellow machines.



What We See

```

08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 40 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 34 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 34 84 20 35 17 12 50
32 98 81 28 64 23 67 10 24 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 24 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 45 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48

```

What Computers See

Fig.3.4 Image Classification

Inputs and Outputs

When a computer sees an image (takes an image as input), it will see an array of pixel values. Depending on the resolution and size of the image, it will see a 32 x 32 x 3 array of numbers (The 3 refers to RGB values). Just to drive home the point, let's say we have a color image in JPG form and its size is 480 x 480. The representative array will be 480 x 480 x 3. Each of these numbers is given a value from 0 to 255 which describe the pixel intensity at that point. These numbers, while meaningless to us when we perform image classification, are the only inputs available to the computer. The idea is that you give the computer this array of numbers and it will output numbers that describe the probability of the image being a certain class (.80 for cat, .15 for dog, .05 for bird, etc).

What We Want the Computer to do

Now that we know the problem as well as the inputs and outputs, let's think about how to approach this. What we want the computer to do is to be able to differentiate between all the images it's given and figure out the unique features that make a dog a dog or that make a cat a cat. This is the process that goes on in our minds subconsciously as well. When we look at a picture of a dog, we can classify it as such if the picture has identifiable features such as paws or 4 legs. In a similar way, the computer is able perform image classification by looking for low level features such as edges and curves, and then building up to more abstract concepts through a series of convolutional layers. This is a general overview of what a CNN does. Let's get into the specifics.

BIOLOGICAL CONNECTION

But first, a little background. When you first heard of the term convolutional neural networks, you may have thought of something related to neuroscience or biology, and you would be right. Sort of. CNNs do take a biological inspiration from the visual cortex. The visual cortex has small regions of cells that are sensitive to specific

regions of the visual field. This idea was expanded upon by a fascinating experiment by Hubel and Wiesel in 1962 (Video) where they showed that some individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation. For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges. Hubel and Wiesel found out that all of these neurons were organized in a columnar architecture and that together, they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks (the neuronal cells in the visual cortex looking for specific characteristics) is one that machines use as well, and is the basis behind CNNs.

STRUCTURE

Back to the specifics. A more detailed overview of what CNNs do would be that you take the image, pass it through a series of convolutional, nonlinear, pooling (down sampling), and fully connected layers, and get an output. As we said earlier, the output can be a single class or a probability of classes that best describes the image. Now, the hard part understands what each of these layers do. So let's get into the most important one.

FIRST LAYER – MATH PART

The first layer in a CNN is always a Convolutional Layer. First thing to make sure you remember is what the input to this conv (I'll be using that abbreviation a lot) layer is. Like we mentioned before, the input is a $32 \times 32 \times 3$ array of pixel values. Now, the best way to explain a conv layer is to imagine a flashlight that is shining over the top left of the image. Let's say that the light this flashlight shines covers a 5×5 area. And now, let's imagine this flashlight sliding across all the areas of the input image. In machine learning terms, this flashlight is called a filter (or sometimes referred to as a neuron or a kernel) and the region that it is shining over is called the receptive field. Now this filter is also an array of numbers (the numbers are called weights or parameters). A very important note is that the depth of this filter has to be the same as the depth of the input (this makes sure that the math works out), so the

dimension of this filter is $5 \times 5 \times 3$. Now, let's take the first position the filter is in for example. It would be the top left corner. As the filter is sliding, or convolving, around the input image, it is multiplying the values in the filter with the original pixel values of the image (aka computing element wise multiplications). These multiplications are all summed up (mathematically speaking, this would be 75 multiplications in total). So now you have a single number. Remember, this number is just representative of when the filter is at the top left of the image. Now, we repeat this process for every location on the input volume. (Next step would be moving the filter to the right by 1 unit, then right again by 1, and so on). Every unique location on the input volume produces a number. After sliding the filter over all the locations, you will find out that what you're left with is a $28 \times 28 \times 1$ array of numbers, which we call an activation map or feature map. The reason you get a 28×28 array is that there are 784 different locations that a 5×5 filter can fit on a 32×32 input image. These 784 numbers are mapped to a 28×28 array.

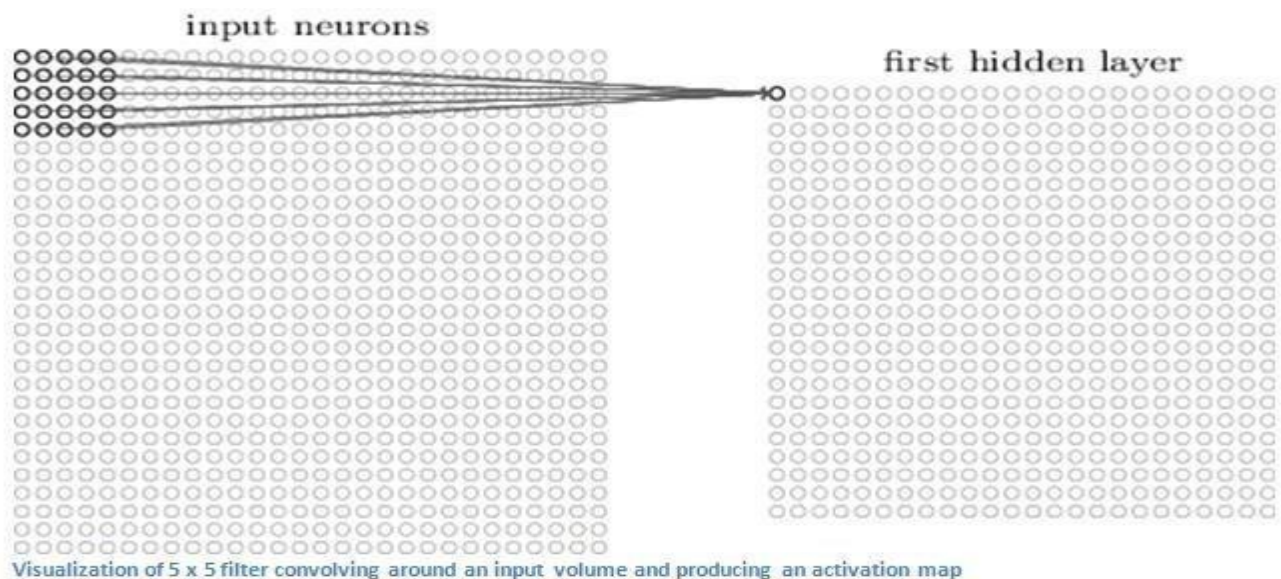


Fig.3.5 Visualization of 5x5 filter convolving around an input volume

Let's say now we use two $5 \times 5 \times 3$ filters instead of one. Then our output volume would be $28 \times 28 \times 2$. By using more filters, we are able to preserve the spatial dimensions better. Mathematically, this is what's going on in a convolutional layer.

FIRST LAYER – HIGH LEVEL PERSPECTIVE

However, let's talk about what this convolution is actually doing from a high level. Each of these filters can be thought of as **feature identifiers**. When I say features, I'm talking about things like straight edges, simple colors, and curves. Think about the simplest characteristics that all images have in common with each other. Let's say our first filter is $7 \times 7 \times 3$ and is going to be a curve detector. (In this section, let's ignore the fact that the filter is 3 units deep and only consider the top depth slice of the filter and the image, for simplicity.) As a curve detector, the filter will have a pixel structure in which there will be higher numerical values along the area that is a shape of a curve.

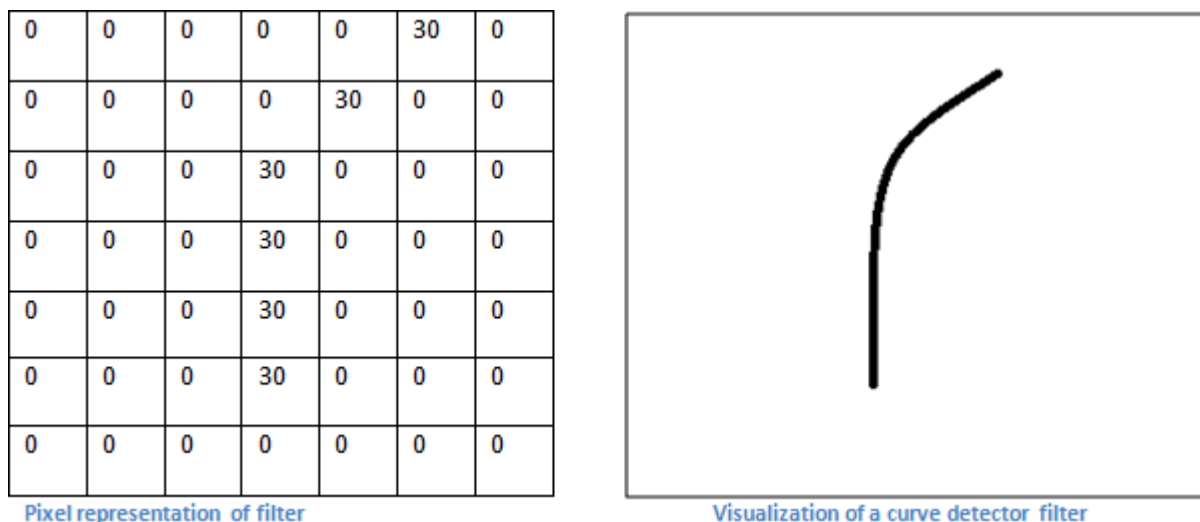


Fig.3.6 Pixel Representation of Filter

Now, let's go back to visualizing this mathematically. When we have this filter at the top left corner of the input volume, it is computing multiplications between the filter and pixel values at that region. Now let's take an example of an image that we want to classify, and let's put our filter at the top left corner.

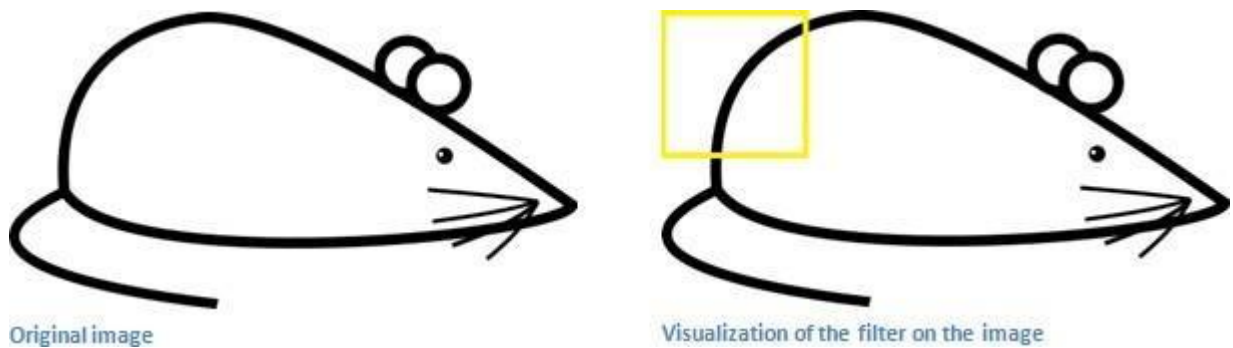
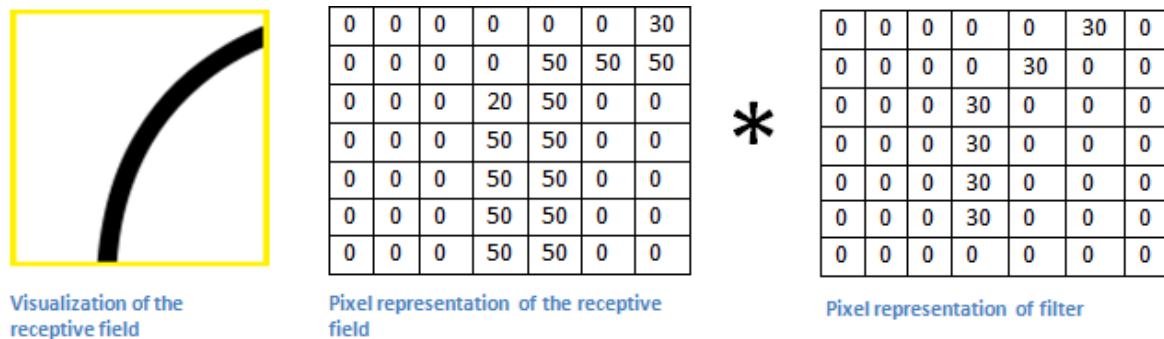


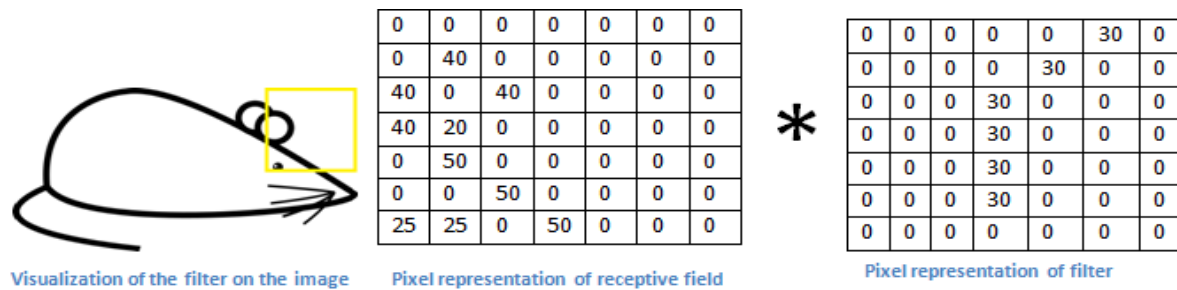
Fig.3.7 Visualization of Filter on the Image

Remember, what we have to do is multiply the values in the filter with the original pixel values of the image.



Multiplication and Summation = $(50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600$ (A large number!)

Basically, in the input image, if there is a shape that generally resembles the curve that this filter is representing, then all of the multiplications summed together will result in a large value! Now let's see what happens when we move our filter.



Multiplication and Summation = 0

The value is much lower! This is because there wasn't anything in the image section that responded to the curve detector filter. Remember, the output of this conv layer is an activation map. So, in the simple case of a one filter convolution (and if that filter is a curve detector), the activation map will show the areas in which there are mostly likely to be curves in the picture. In this example, the top left value of our 26 x 26 x 1 activation map (26 because of the 7x7 filter instead of 5x5) will be 6600. This high value means that it is likely that there is some sort of curve in the input volume that caused the filter to activate. The top right value in our activation map will be 0 because there wasn't anything in the input volume that caused the filter to activate (or more simply said, there wasn't a curve in that region of the original image). Remember, this is just for one filter. This is just a filter that is going to detect lines that curve outward and to the right. We can have other filters for lines that curve to the left or for straight edges. The more filters, the greater the depth of the activation map, and the more information we have about the input volume.

FULLY CONNECTED LAYER

Now that we can detect these high level features, the icing on the cake is attaching a **fully connected layer** to the end of the network. This layer basically takes an input volume (whatever the output is of the conv or ReLU or pool layer preceding it) and outputs an N dimensional vector where N is the number of classes that the program has to choose from. For example, if you wanted a digit classification program, N would be 10 since there are 10 digits. Each number in this N dimensional vector represents

the probability of a certain class. For example, if the resulting vector for a digit classification program is $[0 \ .1 \ .1 \ .75 \ 0 \ 0 \ 0 \ 0 \ 0 \ .05]$, then this represents a 10% probability that the image is a 1, a 10% probability that the image is a 2, a 75% probability that the image is a 3, and a 5% probability that the image is a 9. The way this fully connected layer works is that it looks at the output of the previous layer (which as we remember should represent the activation maps of high level features) and determines which features most correlate to a particular class. **For example**, if the program is predicting that some image is a dog, it will have high values in the activation maps that represent high level features like a paw or 4 legs, etc. Similarly, if the program is predicting that some image is a bird, it will have high values in the activation maps that represent high level features like wings or a beak, etc. Basically, a FC layer looks at what high level features most strongly correlate to a particular class and has particular weights so that when you compute the products between the weights and the previous layer, you get the correct probabilities for the different classes.

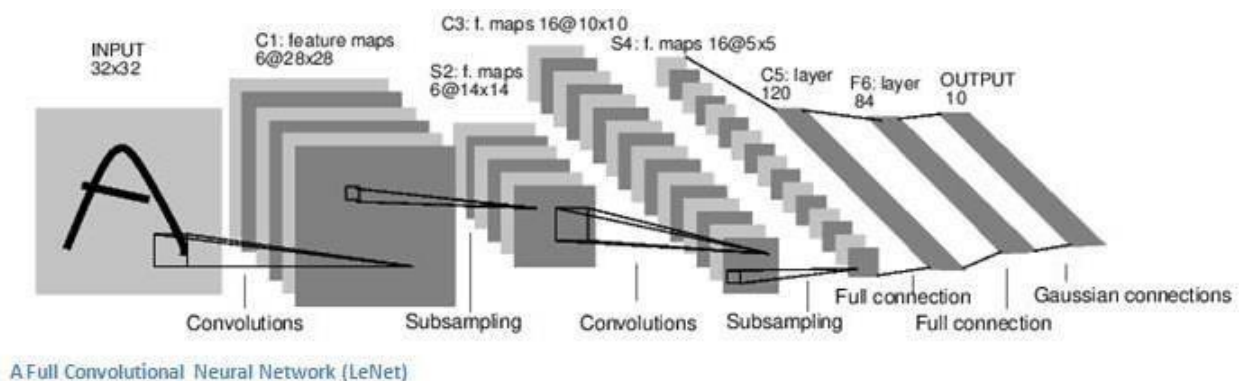


Fig.3.8 A Full Convolutional Neural Network

TRAINING

The way the computer is able to adjust its filter values (or weights) is through a training process called **backpropagation**.

Before we get into backpropagation, we must first take a step back and talk about what a neural network needs in order to work. At the moment we all were born, our minds were fresh. We didn't know what a cat or dog or bird was. In a similar sort of way, before the CNN starts, the weights or filter values are randomized. The filters don't know to look for edges and curves. The filters in the higher layers don't know to look for paws and beaks. As we grew older however, our parents and teachers showed us different pictures and images and gave us a corresponding label. This idea of being given an image and a label is the training process that CNNs go through. Before getting too into it, let's just say that we have a training set that has thousands of images of dogs, cats, and birds and each of the images has a label of what animal that picture is. Back to backprop.

So backpropagation can be separated into 4 distinct sections, the forward pass, the loss function, the backward pass, and the weight update. During the **forward pass**, you take a training image which as we remember is a 32 x 32 x 3 array of numbers and pass it through the whole network. On our first training example, since all of the weights or filter values were randomly initialized, the output will probably be something like basically an output that doesn't give preference to any number in particular. The network, with its current weights, isn't able to look for those low level features or thus isn't able to make any reasonable conclusion about what the classification might be. This goes to the **loss function** part of backpropagation. Remember that what we are using right now is training data. This data has both an image and a label. Let's say for example that the first training image inputted was a 3. The label for the image would be [0 0 0 1 0 0 0 0 0 0]. A loss function can be defined in many different ways but a common one is MSE (mean squared error), which is $\frac{1}{2}$ times (actual - predicted) squared.

$$E_{total} = \sum \frac{1}{2}(target - output)^2$$

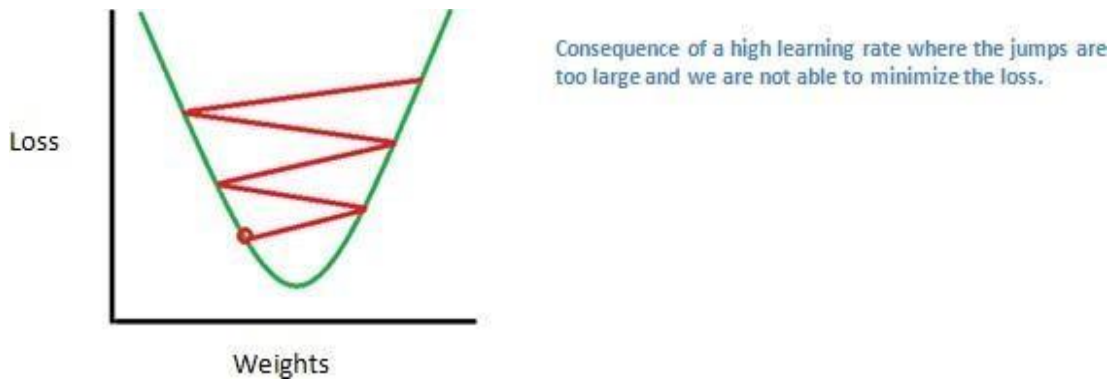
Let's say the variable L is equal to that value. As you can imagine, the loss will be extremely high for the first couple of training images. Now, let's just think about this intuitively. In order to get there, we want to minimize the amount of loss we have. Visualizing this as just an optimization problem in calculus, we want to find out which inputs (weights in our case) most directly contributed to the loss (or error) of the network.

This is the mathematical equivalent of a dL/dW where W are the weights at a particular layer. Now, what we want to do is perform a **backward pass** through the network, which is determining which weights contributed most to the loss and finding ways to adjust them so that the loss decreases. Once we compute this derivative, we then go to the last step which is the **weight update**. This is where we take all the weights of the filters and update them so that they change in the opposite direction of the gradient.

$$w = w_i - \eta \frac{dL}{dW}$$

w = Weight w_i = Initial Weight η = Learning Rate
--

The **learning rate** is a parameter that is chosen by the programmer. A high learning rate means that bigger steps are taken in the weight updates and thus, it may take less time for the model to converge on an optimal set of weights. However, a learning rate that is too high could result in jumps that are too large and not precise enough to reach the optimal point.



The process of forward pass, loss function, backward pass, and parameter update is one training iteration. The program will repeat this process for a fixed number of iterations for each set of training images (commonly called a batch). Once you finish the parameter update on the last training example, hopefully the network should be trained well enough so that the weights of the layers are tuned correctly.

DESIGN AND IMPLEMENTATION CONSTRAINTS

CONSTRAINTS IN ANALYSIS

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure

CONSTRAINTS IN DESIGN

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses
- Global actions and Constraint Realization

CONSTRAINTS IN IMPLEMENTATION

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

OTHER NONFUNCTIONAL REQUIREMENTS

PERFORMANCE REQUIREMENTS

The application at this side controls and communicates with the following three main general components.

- Embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture.

The components at this tier are the following:

- Web Server,
- Security Module,
- Server-Side Capturing Engine,
- Preprocessing Engine,
- Database System,
- Verification Engine,
- Output Module.

SAFETY REQUIREMENTS

- The software may be safety-critical. If so, there are issues associated with its integrity level
- The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
- If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.

ARCHITECTURE DIAGRAM

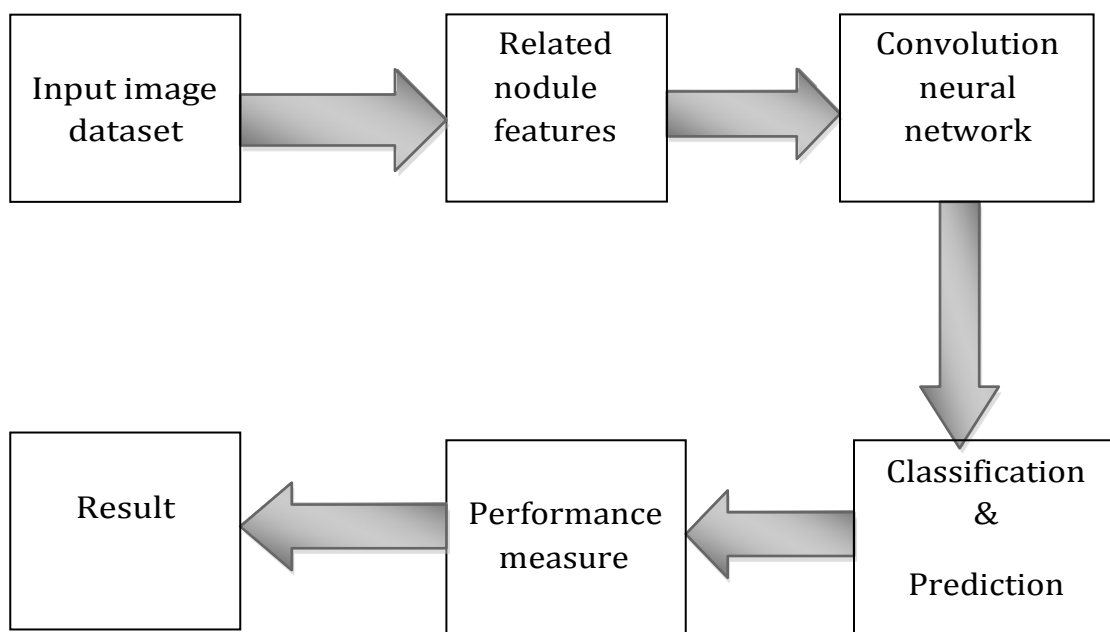


Fig.3.9 Architecture Diagram

SYSTEM DESIGN AND IMPLEMENTATION

UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software-intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- a. activities
- b. actors
- c. business processes
- d. database schemas

UML combines techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Botch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems. UML is a de facto industry standard, and is evolving under the auspices of the Object Management Group (OMG). UML models may be automatically transformed to other representations (e.g. Java) by means of QVT-like transformation languages. UML is extensible, with two mechanisms for customization: profiles and stereotypes.

SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

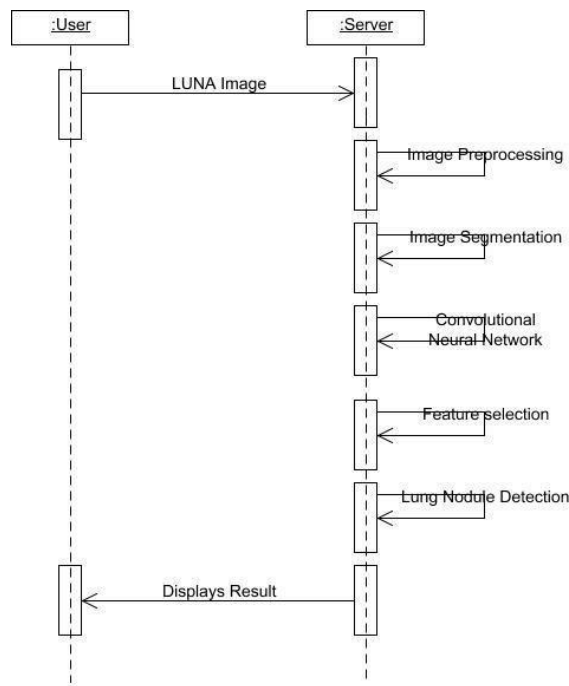


FIG.3.10 Sequence Diagram

USE CASE DIAGRAM

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

In software and systems engineering, a **use case** is a list of steps, typically defining interactions between a role (known in UML as an "actor") and a system, to achieve a goal. The actor can be a human or an external system. In systems engineering,

use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be captured in system or as contractual statements.

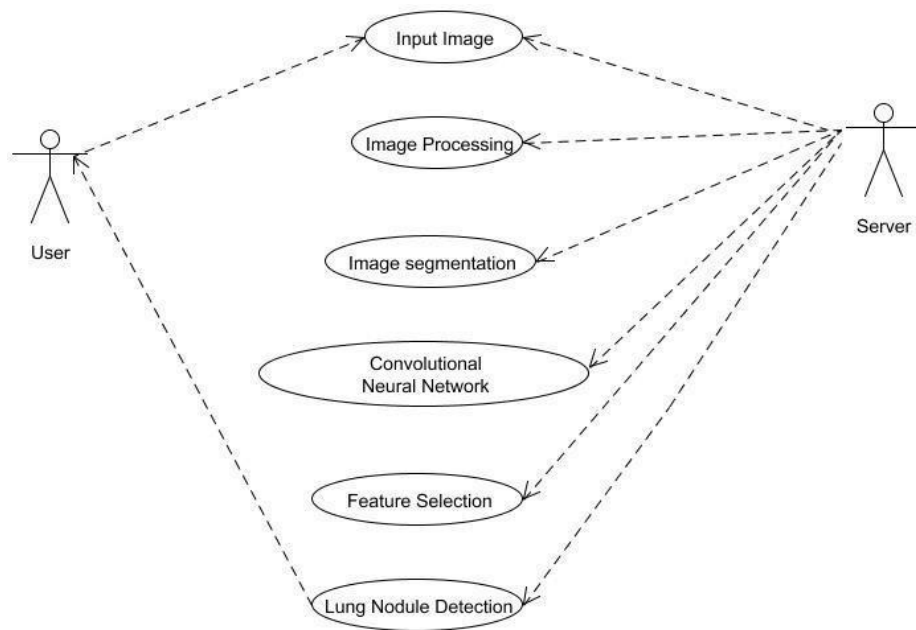


Fig.3.11 Use Case Diagram

ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

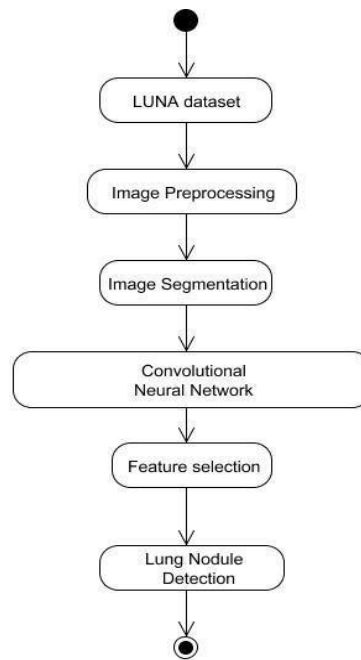


Fig.3.12 Activity Diagram

COLLABORATION DIAGRAM

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

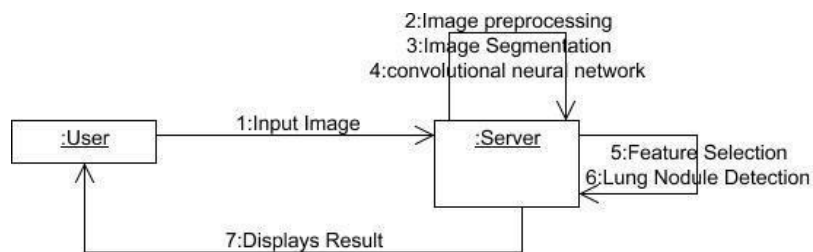


Fig.3.13 Collaboration Diagram

MODULES

PREPROCESSING

CT images contain the lung areas and bone areas, the negative sample coordinates obtained are more practical by limiting the sample range in the lung areas. Then all the points expressed in voxel coordinates. In the second stage, we will find those slices from the CT scans where those points are located according to their z coordinate. In each of these slices, we take the point (x, y) which is defined by x and y coordinates as center point and crop a certain size of square patch around the center point.

CONVOLUTION NEURAL NETWORK ALGORITHM

After observing the structure of our multi-resolution CNN model that the side output branches of the network seems to make more contributions to the processing of multi resolution information. The network configuration by removing these branches or their corresponding classifiers and conduct the lung nodular classification experiments with the modified configurations and corresponding training methods.

CLASSIFICATION

The image with big size contains more semantic information about the environment around the target, whereas the smaller one is more focused on the target itself and contains less semantic information. Then the images with different resolutions together for experiments and compare these with test images in the same size from the original test set.

PERFORMANCE MEASURE

A single CNN (Convolution neural network) model is more convenient to train and accuracy is better than that of the machine learning algorithm

CODING

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

Coding Standards

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear and easy to understand.
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

Naming Conventions

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

Value Conventions

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete.

Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

SOFTWARE TESTING STRATEGIES

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.
- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

PROGRAM TESTING

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data

fields, out-of-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

SECURITY TESTING

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

USER ACCEPTANCE TESTING

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following point:

- Input screen design.
- Output screen design.

CHAPTER 4

RESULTS AND DISCUSSIONS

The neural network based on convolutional and watershed segmentation has been implemented in MATLAB and the system is trained with sample data sets for the model to understand and familiarize the lung cancer. A sample image has been fed as an input to the trained model and the model at this stage is able to tell the presence of cancer and locate the cancer spot in the sample image of a lung cancer. The process involves the feeding the input image, preprocessing, feature extraction, identifying the cancer spot and indicate the results to the user. In case of the malignancy is present, a message indicating the presence of will be displayed on the screen



Fig 4.1 Sample Image of Lung for Detection

Lung cancer detection using the convolutional neural network which model by the end to end learning, i.e. Initialize the weights, Learning rate, gradient moment and hidden neurons. In the neural network, hidden units to form a zero matrix whatever weights will change to be hidden units would change the matrix value. Some of the parameter used for training the model of the neural network is shown in Table 1.



Fig.4.2 Output for Cancerous Image

Parameter	Value
Learning rate	0.0001
Weight	0.0002
Bias	0.1
Gradient moment	0.9
Hidden neurons	250
Epoch	100

Table.4.1: FFC Layer with Compute the Class Score

CNN has two layers such as 2 convolution layers and 2 sub sampling layer which is used to increase the accuracy of detection. The confusion matrix parameters derived from

CNN output are given in Table.2

S.No	Parameters	Values
1	Training images	70
2	Test images	30
3	True Positive	7
4	True Negative	22

5	False Positive	0
6	False Negative	1
7	Specificity	1
8	Sensitivity	0.875
9	Overall Accuracy	0.96

Table.4.2: Confusion Matrix

The confusion matrix shows the true positive, true negative, false positive and false negative. From the analysis true positive gives the correctly classified the lung cancer images and false positive gives the misclassification of images which means that the lung cancer is wrongly predicted as non-cancerous image.

CHAPTER 5

CONCLUSION AND FUTURE WORK

CONCLUSION

A convolutional neural network based system was implemented to detect the malignancy tissues present in the input lung CT image. Lung image with different shape, size of the cancerous tissues has been fed at the input for training the system. The proposed system is able to detect the presence and absence of cancerous cells with accuracy of about 96%. The accuracy of Lung cancer detection with the proposed convolutional neural network based method was compared with that obtained by previous works in Table 3:

Authors	Sensitivity	Specificity
Albrt Chon. et al	43%	85%
Devi Nutiyasari, et al	86.30%	###
Proposed Method	100%	###

Table.5.1 Comparison of Results with the Previous Works

In addition to deep convolutional network, the same dataset was classified by multilayer perceptron network Backpropagation algorithm with using GLCM features. The results show only 93% accuracy [10].

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

FUTURE SCOPE

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

REFERENCES

- [1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA: A Cancer Journal for Clinicians*, vol. 67, no. 1, pp. 7–30, 2017.
- [2] L. A. Torre, R. L. Siegel, and A. Jemal, "Lung cancer statistics," in *Lung Cancer and Personalized Medicine*, pp. 1–19, Springer, Berlin, Germany, 2016.
- [3] K. Doi, "Computer-aided diagnosis in medical imaging: historical review, current status and future potential," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.
- [4] Q. Song, L. Zhao, X. Luo, and X. Dou, "Using deep learning for classification of lung nodules on computed tomography images," *Journal of Healthcare Engineering*, vol. 2017, Article ID 8314740, 7 pages, 2017.
- [5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll'ar, "Focal loss for dense object detection," in *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, Venice, Italy, October 2017.
- [6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1, NIPS'12*, pp. 1097–1105, Curran Associates Inc., Lake Tahoe, NV, USA, December 2012.
- [8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of IEEE International Conference on Learning Representations*, Banff, AB, Canada, April 2014.

- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [10] W. Li, P. Cao, D. Zhao, and J. Wang, "Pulmonary nodule classification with deep convolutional neural networks on computed tomography images," *Computational and Mathematical Methods in Medicine*, vol. 2016, Article ID 6215085, 7 pages, 2016.
- [11] J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for CTimages," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 202–209, 2014.
- [12] W.-J. Choi and T.-S. Choi, "Automated pulmonary nodule detection system in computed tomography images: a hierarchical block classification approach," *Entropy*, vol. 15, no. 2, pp. 507–523, 2013.
- [13] A. A. A. Setio, F. Ciompi, G. Litjens et al., "Pulmonary nodule detection in CT images: false positive reduction using multiview convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.
- [14] E. L. Torres, E. Fiorina, F. Pennazio et al., "Large scale validation of the M5L lung CAD on heterogeneous CTdatasets," *Medical Physics*, vol. 42, no. 4, pp. 1477–1489, 2015.
- [15] A. A. A. Setio, A. Traverso, T. de Bel et al., "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge," *Medical Image Analysis*, vol. 42, pp. 1–13, 2017.
- [16] S. G. Armato III, G. McLennan, L. Bidaut et al., "The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on ct scans," *Medical Physics*, vol. 38, no. 2, pp. 915–931, 2011.
- [17] D. Manos, J. M. Seely, J. Taylor, J. Borgaonkar, H. C. Roberts, and J. R. Mayo, "The lung reporting and data system (LURADS): a proposal for computed tomography screening," *Canadian Association of Radiologists Journal*, vol. 65, no. 2, pp. 121–134, 2014.

APPENDIX

1. SOURCE CODE

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import skimage, os
from scipy import ndimage as ndi
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
#import dicom
import scipy.misc
import numpy as np
from subprocess import check_output
#print(check_output(["ls", "../LUNG DATA"]).decode("utf8"))
print(os.listdir('E:/Multi-Resolution CNN and Knowledge in Lung Nodule'))
from skimage.morphology import ball, disk, dilation, binary_erosion,
remove_small_objects, erosion, closing, reconstruction, binary_closing
from skimage.measure import label,regionprops, perimeter
from skimage.morphology import binary_dilation, binary_opening
from skimage.filters import roberts, sobel
from skimage import measure, feature
from skimage.segmentation import clear_border
from skimage import data
from skimage import measure
print(os.listdir('E:/Multi-Resolution CNN and Knowledge in Lung Nodule/sample
images'))
import pydicom
import scipy.misc
import numpy as np
dataset = pydicom.dcmread('E:/Multi-Resolution CNN and Knowledge in Lung
Nodule/sample images/000000.dcm')
```

```

dataset
slice = dataset.pixel_array
slice[slice == -2000] = 0
plt.imshow(slice, cmap=plt.cm.gray)
%matplotlib inline
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
showid = 0 # from 0 to 4
assert showid in range(5)
srslst = ['1.3.6.1.4.1.14519.5.2.1.6279.6001.208737629504245244513001631764',
'1.3.6.1.4.1.14519.5.2.1.6279.6001.108231420525711026834210228428',
'1.3.6.1.4.1.14519.5.2.1.6279.6001.161002239822118346732951898613',
'1.3.6.1.4.1.14519.5.2.1.6279.6001.162901839201654862079549658100',
'1.3.6.1.4.1.14519.5.2.1.6279.6001.179162671133894061547290922949']
ctdat =
np.load('E:/Python/Full/lung/CT_training_Model/'+srslst[showid]+'_clean.npy',allow_pickle=True)
ctlab =
np.load('E:/Python/Full/lung/CT_training_Model/'+srslst[showid]+'_label.npy',allow_pickle=True)
print(ctdat.shape, ctlab.shape)
for idx in range(ctlab.shape[0]):
    if abs(ctlab[idx,0])+abs(ctlab[idx,1])+abs(ctlab[idx,2])+abs(ctlab[idx,3])==0: continue
    fig = plt.figure()
    z, x, y = int(ctlab[idx,0]), int(ctlab[idx,1]), int(ctlab[idx,2])
    dat0 = np.array(ctdat[0, z, :, :])
    dat0[max(0,x-10):min(dat0.shape[0],x+10), max(0,y-10)] = 255
    dat0[max(0,x-10):min(dat0.shape[0],x+10), min(dat0.shape[1],y+10)] = 255
    dat0[max(0,x-10), max(0,y-10):min(dat0.shape[1],y+10)] = 255

```

```

dat0[min(dat0.shape[0],x+10), max(0,y-10):min(dat0.shape[1],y+10)] = 255
plt.imshow(dat0)
plt.show()
def iou(box0, box1):
    r0 = box0[3] / 2
    s0 = box0[:3] - r0
    e0 = box0[:3] + r0
    r1 = box1[3] / 2
    s1 = box1[:3] - r1
    e1 = box1[:3] + r1
    overlap = []
    for i in range(len(s0)): overlap.append(max(0, min(e0[i], e1[i]) - max(s0[i], s1[i])))
    intersection = overlap[0] * overlap[1] * overlap[2]
    union = box0[3] * box0[3] * box0[3] + box1[3] * box1[3] * box1[3] - intersection
    return intersection / union
def nms(output, nms_th):
    if len(output) == 0: return output
    output = output[np.argsort(-output[:, 0])]
    bboxes = [output[0]]
    for i in np.arange(1, len(output)):
        bbox = output[i]
        flag = 1
        for j in range(len(bboxes)):
            if iou(bbox[1:5], bboxes[j][1:5]) >= nms_th:
                flag = -1
                break
        if flag == 1: bboxes.append(bbox)
    bboxes = np.asarray(bboxes, np.float32)
    return bboxes
pbb = np.load('E:/Python/Full/lung/Testing_images/'+srslst[showid]+'_pbb.npy')

```

```

lbb = np.load('E:/Python/Full/lung/Testing_images/'+srslst[showid]+'_lbb.npy')
pbb = np.array(pbb[pbb[:,0] > -2])
pbb = nms(pbb, 0.1)
# print pbb.shape, pbb
print('Detection Results according to confidence')
for idx in range(pbb.shape[0]):
    fig = plt.figure()
    z, x, y = int(pbb[idx,1]), int(pbb[idx,2]), int(pbb[idx,3])
#s print z,x,y
    dat0 = np.array(ctdat[0, z, :, :])
    dat0[max(0,x-10):min(dat0.shape[0],x+10), max(0,y-10)] = 255
    dat0[max(0,x-10):min(dat0.shape[0],x+10), min(dat0.shape[1],y+10)] = 255
    dat0[max(0,x-10), max(0,y-10):min(dat0.shape[1],y+10)] = 255
    dat0[min(dat0.shape[0],x+10), max(0,y-10):min(dat0.shape[1],y+10)] = 255
    plt.imshow(dat0)

```

2. SCREENSHOTS

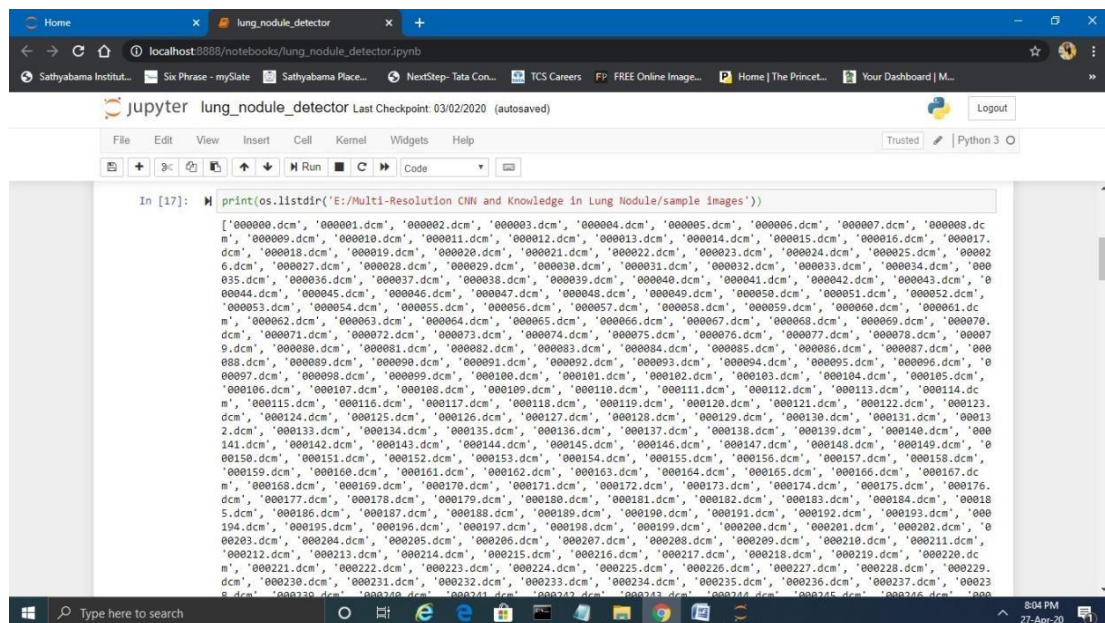


FIG.1 Preprocessing of an Image

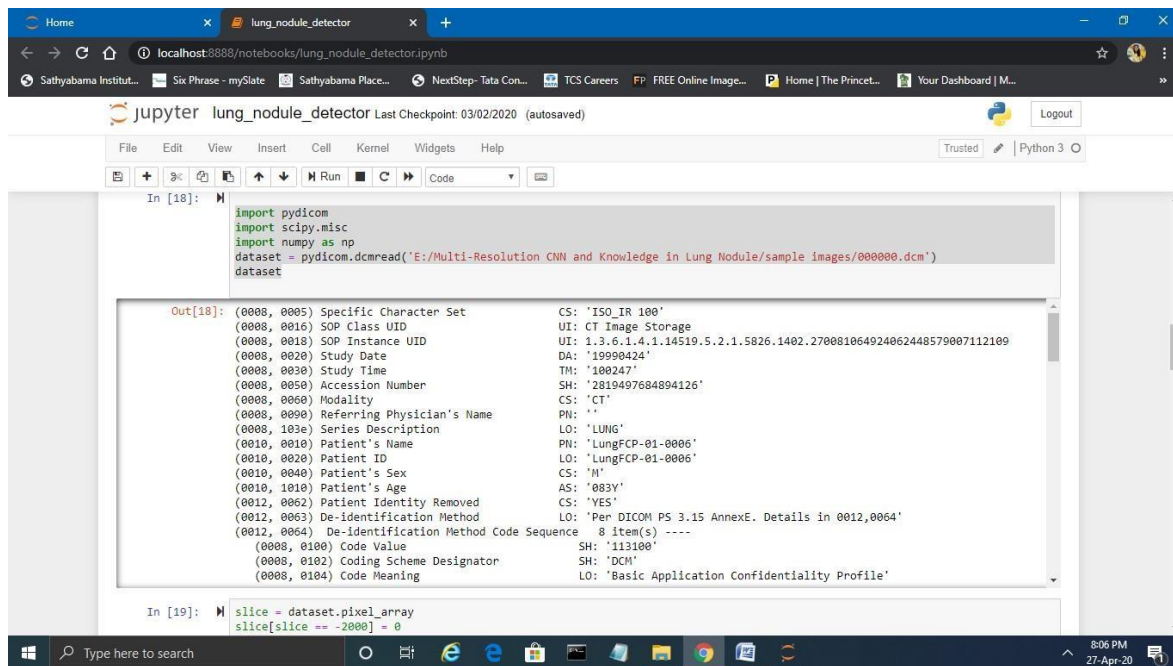


Fig.2. Reading a Dataset

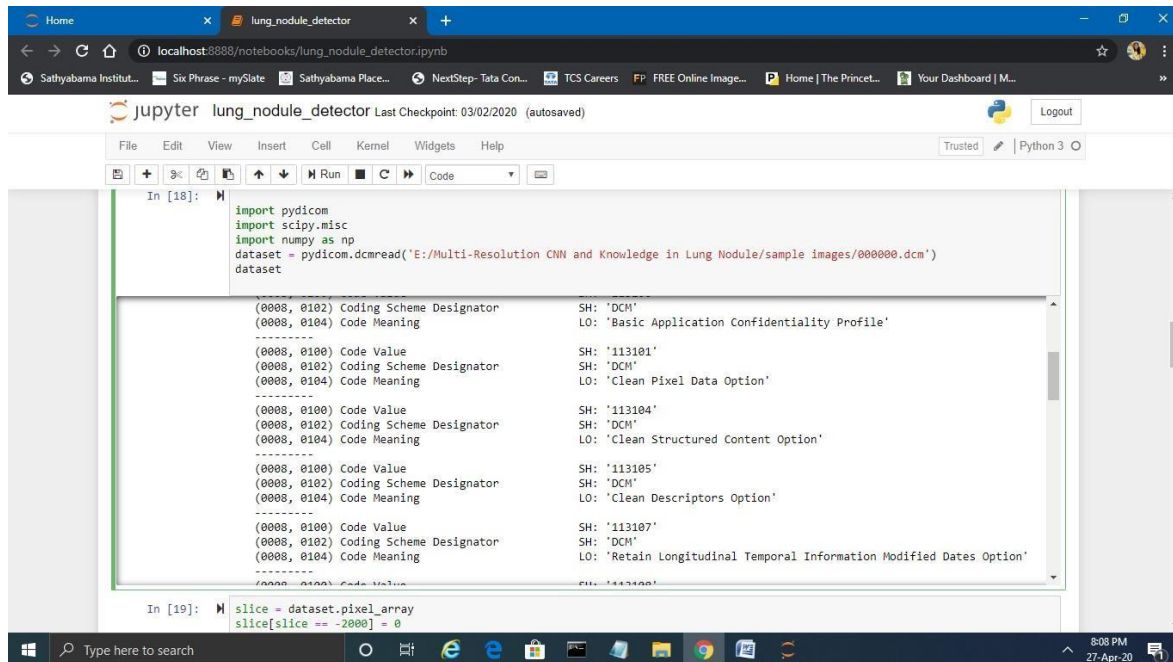


Fig.3. Cleaning the Image

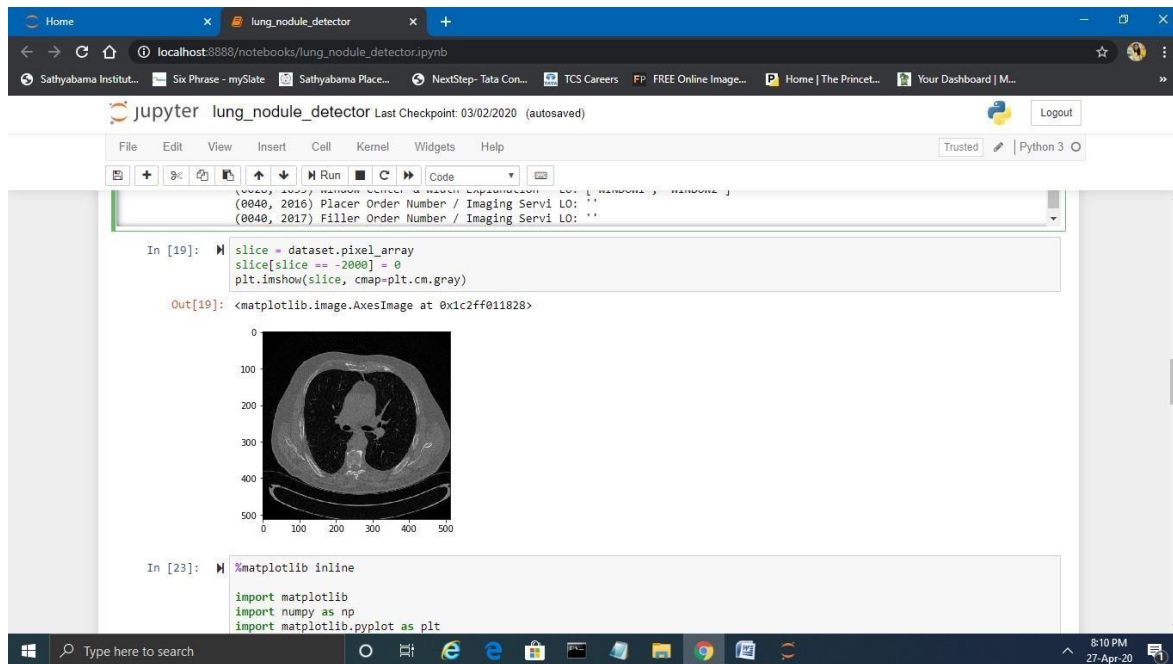


Fig.4. Preprocessed Image of Lung

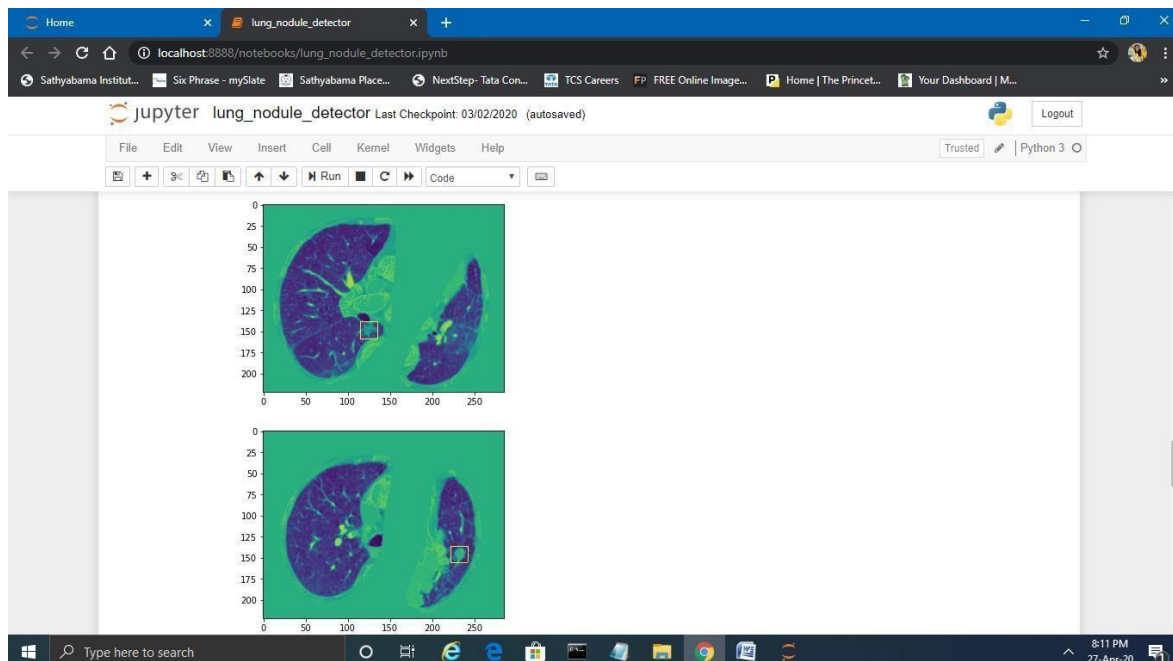


Fig.5. Testing the Lung Image

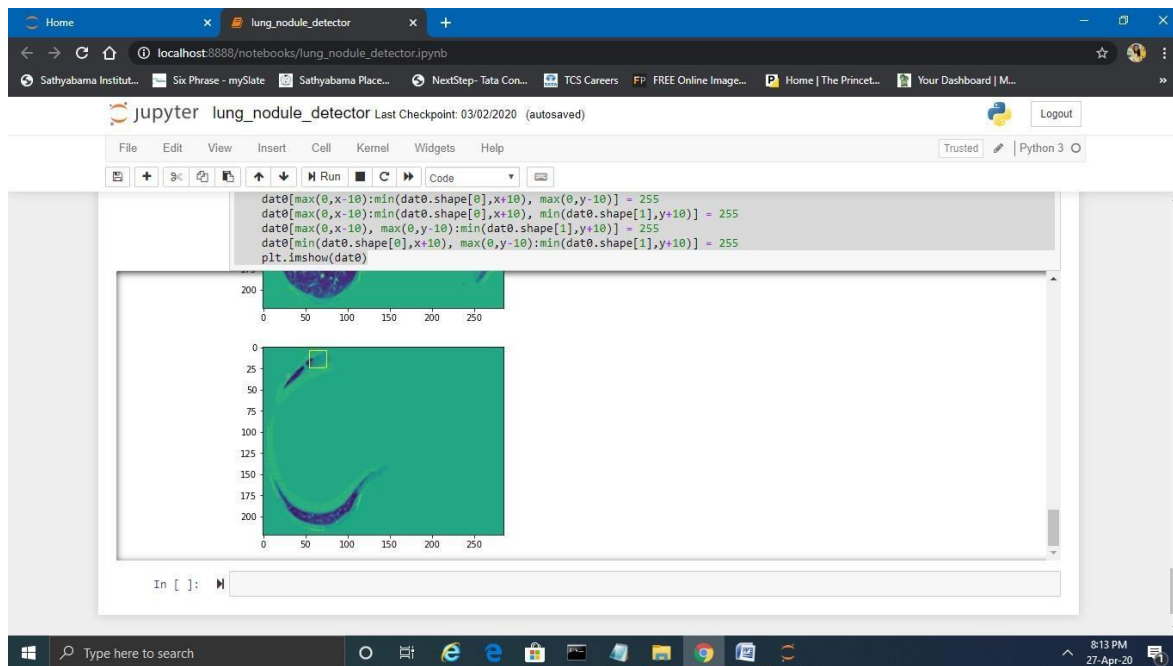


Fig.6. Lung Cancer Detected

3. PAPER PUBLICATION

PREDICTION OF LUNG CANCER USING CONVOLUTION NEURAL NETWORK (CNN)

MORAMPUDI SADHANA ^[1], MEDATATI DAMINI ^[2], B KEERTHI SAMHITHA ^[3]

**[1][2] UG Student, Dept. of Computer Science, Sathyabama University, Chennai,
India**

**[3] Assistant professor, Dept. of Computer Science, Sathyabama University,
Chennai, India**

ABSTRACT: - Early identification of lung nodules is right now the one of the most effective approaches to foresee and treat lung disease. Accordingly, the previous decade has seen a great deal of spotlight on computer aided diagnosis (CAD) of lung nodules , whose objective is to productively identify, portion lung nodules and arrange them as being generous or harmful. Powerful recognition of such nodules stays a test because of their intervention fit as a fiddle, size and surface. This paper plan to recognize and group the lung malignant growth utilizing CNN calculation. This paper introduces a methodology which utilizes a Convolutional Neural Network (CNNs) to order tumors found in lung disease screening figured tomography filters as threatening or amiable. CNNs have extraordinary properties, for example, spatial invariance, and take into consideration different component extraction. As a progressively mechanized methodology, the CNN technique utilizes crude picture information as information and can be straightforwardly classified as yield. The system can delineate picture of lung knob competitor into attributes of various goals and scales while experiencing the trouble in qualities portrayal of pneumonic knob for its radiological heterogeneity and fluctuation in sizes and shapes, in this way enormously diminishing the bogus positive of classification task and improving the scores on classification measurements.

KEYWORDS: - lung nodule classification, Convolutional neural network, Detection.

1. INTRODUCTION

The Lung harm is one of the most certified and typical sorts of sickness wherever all through the world, both in number of new patients and in number of fatalities. There were assessed to be 1.8 million new examples of lung dangerous development worldwide in 2015 [1]. In the US in 2017, a total of 225,000 lung infection new cases and 155,870 passings are recorded, speaking to 26% outright number of dangerous development related passings [1]. As demonstrated by Torre et al. [2], a proportion of 55% of patient's continuance pace of lung danger can be practiced with limited revelation. Along these lines, it is imperative to take a gander at and watch lung knobs eagerly when they are still toward the starting time frame. CT test is perhaps the most ideal ways for starting period area of lung knobs due to its ability to create 3D photos of the chest, making significant standards of knobs and tumor pathology. In any case, early recognizable proof of lung knobs is an irksome and dull endeavor: radiologists need to genuinely and mindfully separate a colossal number of pictures in CT channels. Doi [3] shows that radiologists may miss up to 30% of lung knobs on account of spreads among them and other conventional anatomic structures.

CAD structures are proposed to support masters and radiologists in lung handle recognizable proof and finding. The technique of CAD for lung harmful development all around involves an acknowledgment structures CADe and a CAD. The goal of CADe is to segment the pneumonic handle candidates as knobs or nonnodules though CADe targets arranging the recognized knobs as liberal or perilous knobs [4]. In light of its arrangement, accuracy of lung handle request is basic for the achievement of a CADe system. Nowadays, significant learning is considered as maybe the best response for certain issues of PC vision and model affirmation, for instance, picture examination, talk affirmation, trademark language getting ready, etc. Convolutional neural framework, a kind of neural frameworks using convolution directors in its layer, is comprehensively applied in object disclosure and gathering with an astounding introduction to the extent accuracy. This ideal situation of significant learning makes it potential for lung handle area and request structures.

To handle the request issue of lung handle up-and-comers, two sorts of testing issues must be centered on:

1) The radiological heterogeneity [12] may achieve the immaterialness of specific knobs however other non-knobs are highlighted. This clashing component is definitely not hard to offer climb to the difficulty of perceiving knobs and non-knobs, genuinely provoking a development in false positive up-and-comers and fake negative up-and-comers.

2) The lung knobs are normally in different sizes and variable shapes. The width dispersal of the lung knobs in the LUNA16 [29] enlightening record changes in a wide range, which allows the tremendous knobs for better unmistakable evidence while the little knobs show lower level of recognizing confirmation when both are recognized by a framework model. It is a lot of equivalent to that the lung knobs show different objectives while facing the framework. This will point of fact convey troubles to the partition of the framework. Also, the variable conditions of lung knobs moreover speak to a test to the fundamental idea with respect to the moved multi-objectives model. The part mapping of different objectives can be gotten from the side-yields after each convolution block (each convolution square contains various layers of convolution) and looking at pooling layers at different profundities. Furthermore, a short time later these features are pooled or cut into a comparative size and are solidified to set up the classifier.

We move and improve a multi-objectives convolution neural framework to complete the extraction of different repeat qualities of the handle candidates. It is the primary gone through for this kind of structure being used for lung handle contender game plan. The exploratory results show that it is fruitful to beat the obstacle realized by the huge assortment in sizes and conditions of lung knobs and the radiological heterogeneity. The framework model applied in the edge acknowledgment of pictures is "picture input/picture yield" plan, which is for pixel-wise request.

We improve the planning and testing plan of the source framework to improve the framework execution in target space. In this way, we get an impressively increasingly clear separation among classes and achieve higher scores on most by far of the request estimations on lung handle contender portrayal. During the arrangement organize, different classifiers are added to the branches in different profundities independently for mishap figuring, and stunned incline cycle is done all the while to together catch up on the update of weight regard, right now the association of the framework. While at the testing stage, the request execution is improved by averaging the desire results of the classifiers related at the mix layer and the most significant layer, however the figure delayed consequences of shallow classifiers with low resolute quality are ignored.

2. RELATED WORK

Neural systems have been utilized widely in the grouping of dangerous tumors [13, 14]. Be that as it may, for the specific instance of tumor grouping, CNNs are progressively fitting: since disease is basically recognized by the calcification designs [13, 14], nearby associated examples are increasingly significant for the order procedure. Be that as it may, on the off chance that we utilize a completely associated neural system, where all regions of the picture influence one another, over the top superfluous loads are figured and off base functionalities start to bring into the characterization methodology. In certain works, for example, [2], broad pre-handling is utilized, preceding utilizing a neural system for grouping. While the precision accomplished hence is noteworthy, numerous calculations are utilized, and in this way can't be utilized straightforwardly in future equipment usage of neuromorphic circuits, for example, those in [15].

CNN began as a development of counterfeit neural system by utilizing convolution administrators to accomplish better comprehension of the info picture. This engineering is structured dependent on the organic visual framework and in this way is exceptionally powerful for picture acknowledgment issues, paying little heed to measure

or scale. In such a design, every neuron is connected to another in a manner with the end goal that it reacts to the responsive field around it.

Misusing the upsides of profound learning, numerous strategies are acquainted with increment grouping exactness of lung knob up-and-comers in CT filters. Li et al. [10] proposed a 2D profound convolutional neural system to order pneumonic knob up-and-comers as knob or nonodule. The proposed system was prepared and approved on 62,492 locales of-intrigue (ROI) picture patches removed from 1,010 CT outputs of the LIDC/IDRI dataset. 40,772 out of 62,492 patches are lung nodules and 21,720 out of 62,492 patches are nonnodules. By tests, the technique can pick up to 86.4% precision and 89.0% affectability of lung knob order.

Kuruvilla et al [11] proposed a Computer helped technique for characterization of lung malignant growth in CT pictures. The creators utilized the measurable highlights, for example, mean, standard deviation, scenes, kurtosis, fifth focal minute, and 6th focal minute as highlights for order. The feed forward back proliferation neural system is demonstrated to give preferable arrangement results over the feed forward neural system. These outcomes show the high caliber of the strategy in deciding noncancerous nodules accurately.

Choi et al [12] presented a pneumonic knob location technique utilizing various leveled square characterization. The technique right off the bat partitions the picture into three-dimensional squares. The entropy investigation is then applied to choose instructive squares for recognizing knob competitors. At last, bolster vector machine (SVM) is utilized to order knob applicants as knob or nonodule.

Setio et al. [13] applied multiview convolutional systems (ConvNets) to recognize lung knob applicants. The strategy was prepared on 1,186 nodules extricated from the LIDC/IDRI dataset by the LUNA16 challenge. Torres et al. [14] utilized a feed forward neural system (FFNN) as a knob applicant classifier. The system arrangement incorporates 13 information neurons, 1 concealed layer with 25 neurons, and 1 neuron in the yield layer to display the twofold classifier. The preparation and testing

competitors were performed on 1,018 CT checks from the LIDC/IDRI dataset. The test extricated 1,186 lung nodules from LIDC-IDRI chest CT pictures and gave these nodules as positive contender for scientists. The results of the test shows that the best individual location framework can acquire up to 92.9% affectability of lung knob identification calculation.

3. EXISTING SYSTEM

The hand-made highlights are regularly utilized in custom strategies for nodule classification. These highlights may incorporate those essential level highlights the size, surface, lung nodules. Furthermore, it might likewise incorporate the elevated level attributes preoccupied from essential level highlights, for example, little waves, Local Binary Patterns and Histogram of Oriented Gradients. The highlights utilized are self-assigned, absence of self learning capacity and the customary techniques are not wise adequate toward work in a start towards finish way.

4. PROPOSED SYSTEM

In this proposed framework the technique utilizing convolution neural systems (CNN) is utilized. As a progressively computerized approach, the CNN strategy utilizes crude picture information as information and can be legitimately classified as yield. The system can delineate picture of lung knob up-and-comer into qualities of various goals and scales while experiencing the trouble in attributes depiction of aspiratory nodule for changeability in sizes, shapes, and its radiological heterogeneity accordingly enormously decreasing the bogus positive of classification task and improving the scores on classification measurements. We utilized the information move strategy to move and reproduce a multi-goals convolutional neural system to finish the order assignment of lung knob up-and-comer. The system can delineate picture of lung knob up-and-comer into qualities of various goals and scales while experiencing the trouble in attributes depiction of pneumonic knob for fluctuation's in sizes, radiological heterogeneity. From this manner enormously we can achieve our goal.

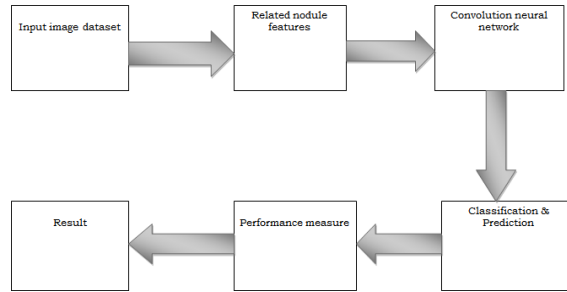


Fig.1 Overview of the Proposed System

5. MODULES DESCRIPTION

PREPROCESSING

CT pictures contain the lung zones and bones regions, the negative example organize acquired are progressively pragmatic by restricting the example extends in the lung zones. At that point every one of the focuses communicated in voxel organizes. In the subsequent stage, we will find those cuts from the CT examines where those focuses are situated by their z organize .In every one of these cuts, we take the point (x, y) which is defined by x and y facilitates as focus point and yield a specific size of square fix around the inside point.

CONVOLUTION NEURAL NETWORK ALGORITHM

In the wake of watching the structure of our multi-goals Convolution neural system that the side yield parts of the system appears to make more commitments to the handling of multi goals data. The system configuration through evacuating these branches or their relating classifiers and lead the lung nodular classification explores different avenues regarding the modified configurations and comparing preparing techniques.

CLASSIFICATION

The picture with large size contains progressively semantic data about nature around the objective, though the littler one is increasingly centered on the objective itself and contains less semantic data. At that point the pictures with various goals together for investigations and contrast these and test pictures in a similar size from the first test set.

PERFORMANCE MEASURE

A solitary CNN (Convolution neural system) model is increasingly advantageous to prepare and exactness is superior to anything that of the AI calculation.

6. CONVOLUTIONAL NEURAL NETWORK

Neural framework building successfully abuses spatial relationship of the data. Furthermore, weight participating in CNN empowers in learning a component paying little regard to its circumstance in the image, close by having the extra piece of breathing space of reduced counts when appeared differently in relation to a totally related ANN. The convolution layer of a CNN produces a segment map by convolving unmistakable sub territories of the image with an academic part. Further, non-straight establishment limits, for instance, a sigmoid, tanh or amended direct (ReLU) can in like manner be applied. The ReLu layer is similarly known to improve the blend properties when the slip-up is low, inciting stagnation in the regular sigmoid commencement work [4]. Another procedure for lessening counts is the pooling layer, where a region of the image/incorporate guide is picked and the most outrageous among them is picked as the specialist pixel. Therefore, a 2x2 or 3x3 cross section can be diminished to a singular scalar worth. Typical characteristics can in like manner be used for pooling, anyway in either case; the net effect is a tremendous reduction in the model size. More bits of knowledge concerning part size, feature maps, stride and various parameters related to a convolutional layer can be found in [4]. Besides, standard totally related

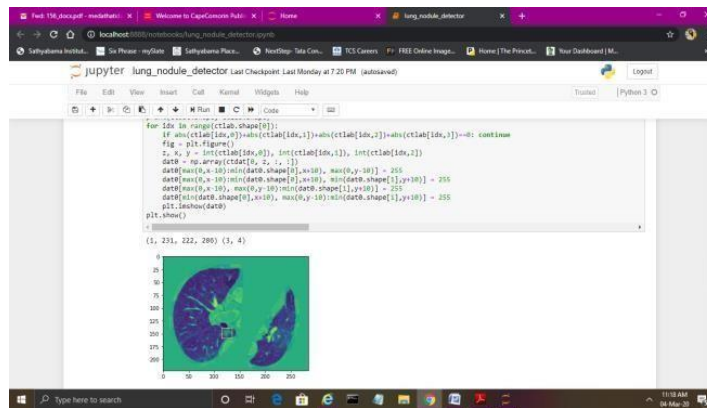


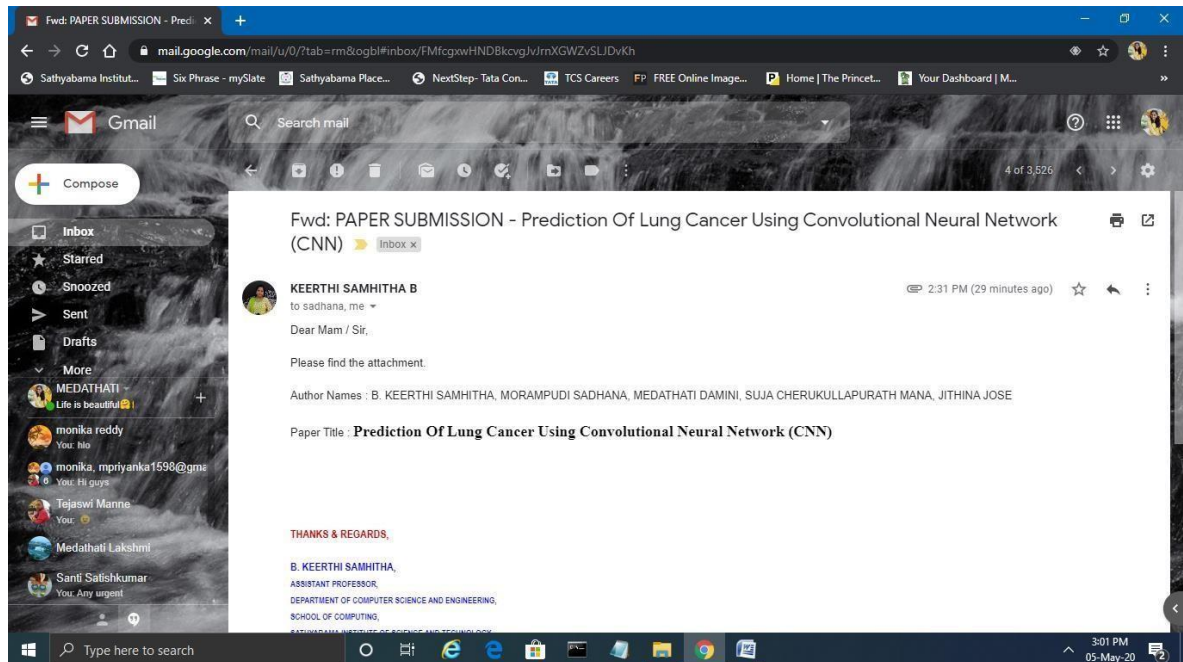
Fig 7.3 CT image of a lung

8. CONCLUSION

The proposing procedure CNNs to separate highlights of different levels, goal from various profundity layers in the system for classification of lung knob up-and-comers. In this proposed framework the strategy utilizing convolution neural systems (CNN) is utilized. As a progressively computerized approach, the CNN strategy utilizes crude picture information as information and can be legitimately classified as yield. A solitary CNN model is more advantageous to prepare and use than that of the AI calculation. We improved a goals that CNN to tackle difficult issues presented. We exhibit the significance and adequacy of getting multi-goals highlight data separated from this system on account of the arrangement undertaking of lung knob applicant.

4. PAPER SUBMISSION

Final paper sent to the IEEE International Conference on Communication and Electronics Systems (ICCES 2020) conducted by PPG Institute of technology.



5. PLAGIARISM REPORT

07

ORIGINALITY REPORT

7%

SIMILARITY
INDEX

3%

INTERNET
SOURCES

5%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Humble Independent School District

Student Paper

1%

2

www.hindawi.com

Internet Source

1%

3

Wangxia Zuo, Fuqiang Zhou, Zuoxin Li, Lin Wang.
"Multi-Resolution CNN and Knowledge Transfer for
Candidate Classification in Lung Nodule Detection",
IEEE Access, 2019

Publication

1%

4	Prajwal Rao, Nishal Ancelette Pereira, Raghuram Srinivasan. "Convolutional neural networks for lung cancer screening in computed tomography (CT) scans", 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016	1%
5	Submitted to Charotar University of Science And Technology	1%
6	Submitted to King Saud University	1%
7	Patrice Monkam, Shouliang Qi, Mingjie Xu, Haoming Li, Fangfang Han, Yueyang Teng, Wei Qian. "Ensemble learning of multiple-view 3D- CNNs model for micro-nodules identification in CT images", IEEE Access, 2018	1%