A Project Report
on

Pulmonary and Lungs Nodule Classification Using Deep Learning

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

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

19WH1A0567	**Mogullapalli Akshita**
19WH1A05A2	**Challa Priyanka**
19WH1A05B9	**Nimmala Shriya Reddy**

under the esteemed guidance of

Ms. A. Kranthi
Assistant Professor

Department of Computer Science and Engineering
BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

June, 2023

DECLARATION

We hereby declare that the work presented in this project entitled **“Pulmonary and Lungs Nodule Classification Using Deep Learning**” submitted towards completion of Project Work in IV year of B.Tech., CSE at ‘BVRIT HYDERABAD College of Engineering For Women**’**, Hyderabad is an authentic record of our original work carried out under the guidance of Ms. A. Kranthi, Assistant Professor, Department of CSE.

M. Akshita
(19WH1A0567)

C. Priyanka
(19WH1A05A2)

N. Shriya Reddy
(19WH1A05B9)

BVRIT HYDERABAD
College of Engineering for Women
(NBA Accredited – EEE, ECE, CSE and IT)
(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)
Bachupally, Hyderabad – 500090

Department of Computer Science and Engineering

Certificate

This is to certify that the Project Work report on "**Pulmonary and Lungs Nodule Classification Using Deep Learning**" is a bonafide work carried out by Mogullapalli Akshita (19WH1A0567), Challa Priyanka (19WH1A05A2), Nimmala Shriya Reddy (19WH1A05B9) in the partial fulfillment for the award of B.Tech. degree in **Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Head of the Department
Dr. E. Venkateswara Reddy
Professor and HoD,
Department of CSE

Guide
Ms. A. Kranthi
Assistant Professor

External Examiner

Acknowledgements

We would like to express our sincere thanks to **Dr. K V N Sunitha, Principal**, **BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. E. Venkateswara Reddy, Professor & HOD**, Department of CSE, **BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Ms. A. Kranthi, Assistant Professor**, Department of CSE**, BVRIT HYDERABAD College of Engineering for Women** for her constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of **CSE** Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

Ms. M. Akshita
(19WH1A0567)

Ms. C. Priyanka
(19WH1A05A2)

Ms. N. Shriya Reddy
(19WH1A05B9)

Contents

ABSTRACT

The diagnosis of early stage lung cancer is challenging due to its asymptomatic nature. Examining the lung CT images to detect pulmonary nodules, especially the cell lung cancer lesions, is also tedious and prone to errors even by a specialist.

The traditional methods of classifying the lung CT images are not so accurate and can easily miss lead in the early stages. Mutation, unregulated tissue growth, refers to the occurrence of permanent changes in deoxyribonucleic acid (DNA) sequences.

Mutation can be due to external factors or inherited genetic abnormalities. The most common external factors are chemicals in tobacco smoke but numerous other carcinogens exist. Earlier we classify and predict the lung nodules, more chances to recover from lung cancer. A better approach to classify the CT images is using deep learning.

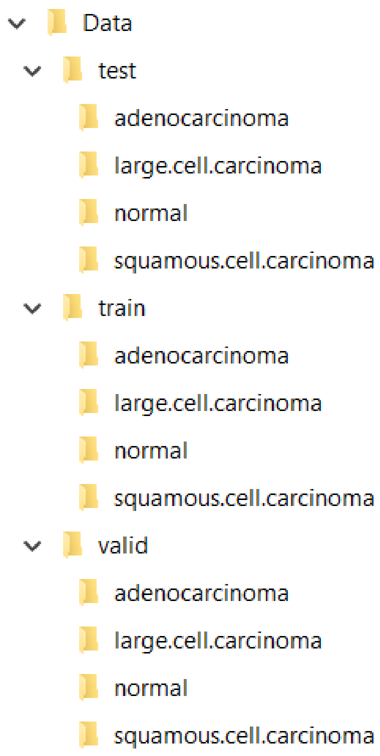
List of Figures

1. INTRODUCTION

1.1. Objectives: The task of identifying nodules in the CT images is time consuming when followed traditional methods. The objective of this project is identification of nodules accurately that will help detect cancer in the early stage and lead to save lives of people. The features are extracted and classified using various fully connected deep learning techniques. Using a publically available dataset of lung CT images consisting of cancerous and non-cancerous lungs, we would train a convolutional neural network to identify lung and pulmonary nodule, to classify the lung as cancerous or not and if cancerous then which type of lung cancer.

1.2. Methodology:

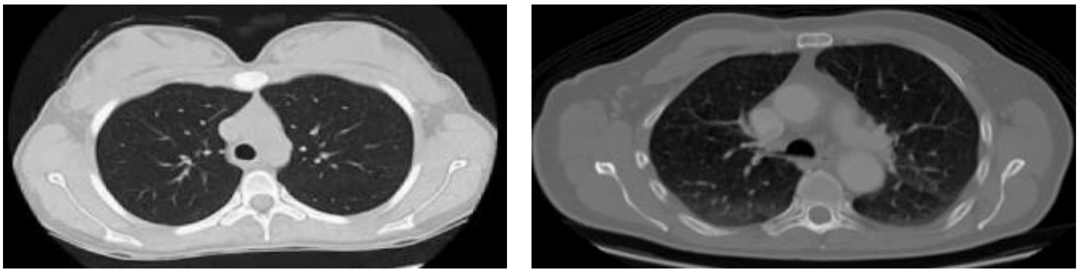
1.2.1. Dataset: Proper dataset is required for all classification classes during the training and the testing phase. The dataset for the experiment is downloaded from the Kaggle which contains 1000 different CT images. The images are split into 70% training set, 20% testing set and 10% validation set. Each having the images classified into 4 classes – normal, adenocarcinoma, large cell carcinoma, squamous cell carcinoma as seen in figure 1. The images are in PNG format as seen in figure 2.



- Data
  - test
    - adenocarcinoma
    - large.cell.carcinoma
    - normal
    - squamous.cell.carcinoma
  - train
    - adenocarcinoma
    - large.cell.carcinoma
    - normal
    - squamous.cell.carcinoma
  - valid
    - adenocarcinoma
    - large.cell.carcinoma
    - normal
    - squamous.cell.carcinoma

Fig 1: Dataset Files Architecture

Sample Images are as follows:



Normal Lung Cancerous Lung

Fig 2: Sample Images

Adenocarcinoma, Large Cell Carcinoma, and Squamous Cell Carcinoma are the different types of lung cancer.

Adenocarcinoma:

Lung adenocarcinoma is the most prevalent form of lung cancer, accounting for approximately 30 percent of all cases and around 40 percent of non-small cell lung cancer instances. This type of cancer is commonly found in various other types of cancers, such as breast, prostate, and colorectal. In the lungs, adenocarcinomas typically occur in the outer region, specifically in the glands responsible for mucus secretion and facilitating breathing. Symptoms may include coughing, hoarseness, weight loss, and weakness.

Large Cell Carcinoma:

Large-cell undifferentiated carcinoma is a rapidly growing and spreading type of lung cancer that can manifest anywhere within the lung. It accounts for approximately 10 to 15 percent of all cases of non-small cell lung cancer. Large-cell undifferentiated carcinoma is known for its aggressive nature, characterized by fast growth and rapid spread throughout the body.

Squamous Cell Carcinoma:

Squamous cell carcinoma predominantly occurs centrally in the lungs, typically at the junction where larger bronchi connect the trachea to the lung or in the main airway branches. It is responsible for approximately 30 percent of all non-small cell lung cancers and is strongly associated with smoking. Squamous cell lung cancer is characterized by the formation of malignant cells in the squamous cells lining the airways.

In summary, lung cancer encompasses different types, including adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. Understanding the characteristics and symptoms associated with each type is crucial for accurate diagnosis, appropriate treatment decisions, and improved patient outcomes.

1.2.2. The Proposed CNN Model: In this proposed system, Convolutional Neural Networks (CNNs) are chosen for the classification of lung CT images due to their inherent advantages in image analysis tasks. Convolutional Neural Networks excel in capturing and learning intricate visual features present in images, making them particularly suitable for analyzing lung CT images that contain complex structural patterns. The hierarchical nature of Convolutional Neural Networks allows them to automatically extract and represent these features at different levels, enabling effective classification.

VGG16, a widely adopted Convolutional Neural Network architecture, offers a simple yet powerful framework for image classification. Its stacked convolutional layers followed by fully connected layers allow for the capture of detailed features and the learning of discriminative representations. VGG16 has demonstrated remarkable performance in various computer vision challenges, making it a reliable choice for this project.

Similarly, InceptionV3, another prominent Convolutional Neural Network architecture, is specifically designed to handle diverse and complex image structures. By incorporating inception modules that perform parallel convolutional operations with different kernel sizes, InceptionV3 can capture both local and global features simultaneously. This characteristic makes it well-suited for accurately classifying lung CT images with varying abnormalities and patterns.

By leveraging the strengths of Convolutional Neural Networks, specifically VGG16 and InceptionV3, the proposed system aims to achieve precise classification of lung CT images into categories such as normal, adeno carcinoma, large cell carcinoma, and squamous cell carcinoma. The established success of these architectures in image classification tasks ensures the system's ability to effectively analyze and classify lung CT images based on their visual characteristics.

1.3. Organization of Project

The technique which is developed is taking input as a lung CT image and extracts the features and filters for the presence of pulmonary or lung nodules, these features extracted are classified based on models trainings and experiences. And the image is finally classified as

- Normal or Uninfected Lung
- Cancerous Lung
  - Adeno Carcinoma
  - Large Cell Carcinoma
  - Squamous Cell Carcinoma

2. LITERATURE REVIEW

Title: Pulmonary Nodule Classification Using Feature and Ensemble Learning-Based Fusion Techniques
Authors: Muhammad Muzammil, Imdad Ali, Ihsan Ul Haq, Amir A. Khaliq, Suheel Abdullah

Summary: In this paper, the author proposed ensemble learners based different fusion techniques, such as; fusion based on averaging of prediction score and fusion based on MAX-VOTE. Deep features were extracted from three state-of-the-art transferable DCNN models and the performance of LDA, SVM and AdaBoostM2 was evaluated for lungs nodule classification. The results showed that the SVM and AdaBoostM2 outperformed over LDA during classification of pulmonary nodule. The achieved classification score was 96.89%±0.25, 99.21%±0.10 and 97.70%±0.21 in terms of accuracy, AUC and specificity, respectively.

Title: LDNNET: Towards Robust Classification of Lung Nodule and Cancer using Lung Dense Neural Network
Authors: Ying Chen, Yerong Wang, Fei Hu, Longfeng Feng, Taohui Zhou, Cheng Zheng

Summary: In this paper, a new network structure of the LDNNET is proposed. LDNNET network is utilized on two datasets, LUNA16 and Kaggle DSB 2017. LDNNET network is robust for the reason that both in the classification of lung nodules and the classification of lung cancer have both achieved good performance. The LDNNET network can achieve higher accuracy than other existing algorithms for different pixel sizes of input lung CT images and different experimental targets. The accuracy of lung cancer classification on Kaggle DSB 2017 is 0.999480. It shows that LDNNET has a good robustness for detecting lung cancers and nodules.

Title: Research on the Auxiliary Classification and Diagnosis of Lung Cancer Subtypes Based on Histopathological Images
Authors: Min Li, Xiaojian Ma, Chen Chen, Yushuai Yuan, Shuailei Zhang, Ziwei Yan, Cheng Chen, Fangfang Chen, Yujie Bai, Panyun Zhou, Xiaoyi Lv, Mingrui Ma

Summary: This paper was the first to apply the Relief-SVM algorithm to the classification of LC histopathological images. It included 121 histopathological images of LUSC, ASC and SCLC. The experimental results show that the Relief- SVM classification model achieves best classification performance regardless of whether it is compared with CNN models which are trained directly by using the histopathology image data set or compared with the fine-tuning pre-trained CNNs models and custom models. Relief-SVM has the best classification performance for identifying lung cancer subtypes, which provides excellent guidance in the classification of lung histopathological images.

Title: Multiscale Rotation-Invariant Convolutional Neural Networks for Lung Texture Classification

Authors: Qiangchang Wang, Yuanjie Zheng, Gongping Yang, Weidong Jin, Xinjian Chen, and Yilong Yin

Summary: Using high-resolution computed tomography, we propose a new multiscale rotation-invariant convolutional neural network (MRCNN) model for identifying lung tissue types. As MRCNN employs a Gabor-local binary pattern, an important property is its invariance to image scale and rotation in image analysis. Different numbers of scales S were used to measure the classification accuracy of Gabor LBP images. ILD refers to a wide variety of lung disorders caused by inflammation and fibrosis of the interstitium. Different ILD lung tissues have subtle texture changes that can be visualized using high-resolution computed tomography (HRCT) scans. The experimental results show that learning multi-scale information is effective for lung tissue classification. This indicates the benefit of having a multi- scale representation for high-level feature learning.

Title: Memory-Augmented Capsule Network for Adaptable Lung Nodule Classification

Authors: Aryan Mobiny, Pengyu Yuan, Pietro A. Cicalese , Supratik K. Moulik, Naveen Garg, Carol C. Wu , Kelvin Wong, Stephen T.Wong, Fellow, IEEE, Tian Cheng He, Hien V. Nguyen

Summary: This paper systematically evaluates the adaptability of deep networks. The author proposed a practical adaptive classifier called MEMCAP which is capable of taking a few annotated inputs from a new target domain to refine its decision making ability accordingly. To determine the maximum performance that a baseline model can achieve with all the available labelled data from each target domain, we also performed the transfer learning experiment The experimental results have demonstrated that when the data distribution changes, the proposed classifier adapts almost perfectly in the lung nodule classification task while popular deep networks performance decrease to chance with large domain shifts. MEMCAP can achieve 97.1% and 96.8% of its theoretical maximum performance.

Title: A Lightweight Multi-Section CNN for Lung Nodule Classification and Malignancy Estimation
Authors: Pranjal Sahu, Dantong Yu, Mallesham Dasari, Fei Hou, and Hong Qin

Summary: In this paper, the author introduced a novel Multi-section CNN for classifying lung nodules and estimating the probability of malignancy. The experiment results showed that our proposed model outperforms several state-of-the- art classification methods. To obtains a nodule's cross sections from multiple view angles and to encode the nodule's volumetric information into a compact representation by aggregating information from its different cross sections via a view pooling layer the lightweight, multiple view sampling based multi-section CNN architecture IS proposed. Multi-section CNN method does not require the tedious manual spatial annotation, is lightweight, and can be easily ported to Mobile devices, such as tablets and embedded system. The proposed method is evaluated and achieved a mean 93.18% accuracy.

Title: Bioimpedance Spectroscopy Measurement and Classification of Lung Tissue to Identify Pulmonary Nodules
Authors: Rasool Baghbani, Mohammad Behgam Shadmehr, Masoomeh Ashoorirad, Seyyedeh Fatemeh Molaeezadeh and Mohammad Hassan Moradi

Summary: This paper investigated by measuring the electrical bioimpedance of the healthy and tumoral lung tissues. For finding a deeply located nodule during the surgical procedure, first the bioimpedance of the A and B regions of the lung tissue (based on the pre-operative CT scan images) is measured by sweeping the surface of the lung by the bioimpedance probe and then the measured data are classified based on the proposed smart system. The whole procedure only takes 3s. Thus, it is possible to identify the exact location of the invisible and impalpable pulmonary nodules during surgical procedures with a safe, rapid, and user-friendly technique. Radiation, cost, and complications, such as pneumothorax, hemothorax, and pulmonary hemorrhage, are the drawbacks of these methods.

Title: Lung Nodule Malignancy Prediction in Sequential CT Scans: Summary of ISBI 2018 Challenge

Authors: Yoganand Balagurunathan, Andrew Beers, Michael McNitt-Gray, Member, IEEE, Lubomir Hadjiiski, Senior Member, IEEE, Sandy Napel, Life Member, Dmitry Goldgof, Fellow, Gustavo Perez, Pablo Arbelaez , Alireza Mehrtash, Tina Kapur, Ehwa Yang, Jung Won Moon, Gabriel Bernardino Perez, Ricard Delgado-Gonzalo, M. Mehdi Farhangi, Amir A. Amini , Fellow, IEEE, Renkun Ni, Xue Feng , Aditya Bagari, Kiran Vaidhya, Benjamin Veasey, Wiem Safta, Hichem Frigui, Joseph Enguehard, Ali Gholipour, Senior Member, IEEE, Laura Silvana Castillo, Laura Alexandra Daza, Paul Pinsky, Jayashree Kalpathy Cramer, Member, IEEE, and Keyvan Farahani

Summary: The National Lung Screening Trial (NLST) demonstrated that screening high risk people with low dose computed tomography (LDCT) reduces lung cancer specific mortality. National Lung Screening Trial (NLST) reported a higher cancer detection rate for the LDCT arm (24.2%) than for the chest X-ray arm (6.9%). It was a successful, community-wide effort that highlighted challenges in diagnosing malignant lung nodules with sequential LDCT scans.

Title: Self-Supervised Transfer Learning Based on Domain Adaptation for Benign- Malignant Lung Nodule Classification on Thoracic CT

Authors: Hong Huang, Member, IEEE, Ruoyu Wu, Yuan Li, and Chao Peng

Summary: In this paper, a new DL-based method called self-supervised transfer learning based on domain adaptation (SSTL-DA) is proposed for classifying benign and malignant lung nodules on thoracic CT. The SSTL-DA method tries to overcome the obstacle due to insufficient training samples by jointly exploiting labeled and unlabeled CT scans. Transfer was used to examine the lung cancer detection ability of pre-trained VGGNet-16 and VGGNet-19 models learning technology. Experimental results on the LIDC-IDRI benchmark dataset prove that the proposed SSTL-DA model performs better than several state-of-the-art approaches in terms of benign-malignant lung nodule classification.

Title: Knowledge-based Collaborative Deep Learning for Benign-Malignant Lung Nodule Classification on Chest CT

Authors: Yutong Xie, Yong Xia, Member, IEEE, Jianpeng Zhang, Yang Song, Member, IEEE, Dagan Feng, Fellow, IEEE, Michael Fulham, and Weidong Cai, Member, IEEE

Summary: In order to detect lung cancer early, chest CT is critical for identifying malignant lung nodules, which provide patients with the best chance of cure. Although large training data sets are lacking for the detection of malignant nodules, deep learning methods have recently been successfully applied to computer vision problems. As part of the knowledge-based collaborative (KBC) model, three types of

image patches are used to finetune three pre-trained ResNet-50 networks that describe the appearance of the nodules in each view. MV-KBC model is used to separate benign from malignant lung nodules on chest CT by taking into account the nodule appearance on nine view planes and nodule heterogeneity and by applying an adaptive weighting scheme so that our model can be trained end-to-end. The MV- KBC model had an accuracy of 91.60% for lung nodule classification.

Title: Modified Quality Threshold Clustering for Temporal Analysis and Classification of Lung Lesions

Authors: Stelmo Magalhaes Barros Netto, Joao Otavio Bandeira Diniz, Aristofanes Correa Silva, Anselmo Cardoso de Paiva, Rofolfo Acatauassú Nunes, and Marcelo Gattass

Summary: Methodology uses a modified version of quality threshold clustering algorithm to know the changes in lesions in a period of time and voxel of the lesion to a cluster. And additionally statistical features are extracted for classification for knowing the lesions is whether cancerous or not. Its main feature is the generation of the best number of clusters for a set of input data. The QT clustering algorithm only performs clustering of density values. The classification accuracy is 98.41%. The technique used in this methodology is the QT clustering algorithm. This clustering technique was specially modified for temporal analysis of the lesion density. This technique was applied to two databases of CT images, one for malignant lesions and the other for benign.

Title: Lung Sound Classification Using Co-Tuning and Stochastic Normalization

Authors: Truc Nguyen and Franz Pernkopf, Senior Member

Summary: In order to provide a more objective evaluation of the lung sound for the diagnosis of pulmonary diseases/conditions, computational lung sound analysis has been created. Recognition of normal and abnormal sounds is crucial for adventitious lung sound classification, while ternary chronic categorization is crucial for the classification of respiratory disorders. Less target task data is needed, faster training is possible, and typically greater performance is attained after fine-tuning the model on the target task. The best co-tuning system for 2-class lung sound classification achieves a better F-score (2.82%) compared to our previous work using a multi-input convolutional neural network.

Title: Evolutionary Cluster-Based Synthetic Oversampling Ensemble (ECO- Ensemble) for Imbalance Learning
Authors: Pin Lim, Student Member, IEEE, Chi Keong Goh, Senior Member, IEEE, and Kay Chen Tan, Fellow, IEEE

Summary: In this paper, the authors presented an ECO-Ensemble framework which showed significant improvement over state-of-the-art algorithms when evaluated over a total of 40 imbalance datasets. The framework consists of a novel synthetic data generation method (CSO) utilizing clustering methods to represent the oversampling dataspace. The parameters within CSO determining the oversampling dataspace are optimized using an EA. Leveraging on the characteristics of the EA, a set of diverse yet accurate ensemble members is created using a subset of the final population. These ensemble members are then combined to deliver relatively superior results over current state-of-the-art methods.

Title: Geometric Structural Ensemble Learning for Imbalanced Problems
Authors: Zonghai Zhu, Zhe Wang, Dongdong Li, Yujin Zhu, and Wenli Du

Summary: This paper analyzes the deficiency of the existing ensemble learning methods in dealing with imbalanced problems and proposes a GSE learning framework. The effectiveness and efficiency of the proposed GSE can be explained as, primarily assume that a section is divided into four parts. In the first part, the experiment settings including the used data sets, comparison groups, and basic settings are given. The classification results of all used algorithms are shown in the second part. The third part presents the training time of all used algorithms to reflect the efficiency of GSE. The time complexity of GSE is O(nd log(nmin)log(nmaj)). The experiments validate both effectiveness and efficiency. Thus, we can conclude that GSE is a not only effective but also fast algorithm in dealing with imbalanced problems.

Title: Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques
Authors: Geoffrey I. Webb, Member, IEEE, and Zijian Zheng

Summary: This paper has examined techniques for combining simple ensemble learning approaches with the aim of exploring the relationship between ensemble member diversity and ensemble error. The results strongly support the proposition that combining effective ensemble learning strategies is conducive to reducing test error. A specific hypothesis about this effect was examined—that combining ensemble learning strategies would increase diversity at the cost of a small increase in individual test error resulting in a tradeoff that reduced overall ensemble test error. A specific hypothesis about the effect was examined-that combining ensemble learning strategies would increase diversity at the cost of a slight increase in individual test error resulting in a tradeoff that reduced overall ensemble test error.

3. THEORETICAL ANALYSIS OF PROPOSED PROJECT

3.1. Requirements Gathering

3.1.1. Software Requirements

- Operating System: Windows
- Programming Language: Python
- Packages: Numpy, Matplotlib, Random, OpenCV, Tensorflow
- Software: Jupyter Notebook

3.1.2. Hardware Requirements

- Processor: Intel Core i5 or above
- RAM: 8GB

3.2. Technological Description

Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors.

This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels.

All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

Tensorflow

TensorFlow is an extensively used open-source software library that offers robust dataflow and differentiable programming capabilities. It is widely adopted for diverse technological applications, particularly in the field of machine learning, specifically neural networks. With its flexible design, TensorFlow caters to the needs of both research and production environments, making it a reliable choice for implementing cutting-edge technologies. Originally developed by the Google Brain team for internal purposes, TensorFlow was later released under the Apache 2.0 open-source license. This licensing choice expanded its accessibility, allowing a wider community to leverage its functionalities and contribute to its development.

For deep learning tasks, TensorFlow provides a comprehensive API called tensorflow.keras, which simplifies the process of building and training neural networks. This API encompasses a range of modules and functions tailored to meet the demands of constructing efficient and scalable deep learning models. The layers module offers a diverse set of layer types, including convolutional, dense, and pooling layers, enabling users to design complex network architectures. The Model module within tensorflow.keras empowers developers to create customized models by specifying the input and output layers, facilitating the construction of neural network topologies tailored to specific technological requirements. This flexibility allows for the incorporation of various architectures and configurations to achieve optimal performance.

TensorFlow also provides specialized modules for data preprocessing and augmentation, streamlining the preparation of input data for training. The ImageDataGenerator module efficiently loads and preprocesses image data, while the preprocess_input function ensures compatibility of input images with specific neural network architectures, optimizing the training process. To enhance training efficiency and effectiveness, TensorFlow offers a range of callback functions. These functions, such as ModelCheckpoint and EarlyStopping, enable automatic model saving during training and early termination based on specified conditions, respectively. This proactive approach ensures model integrity and enhances training workflow.

Furthermore, TensorFlow facilitates the utilization of pre-trained models, enabling transfer learning for improved efficiency and performance. The load_model function allows for the integration of pre-trained weights and architectures, serving as a solid foundation for training on new datasets or solving related tasks. This transfer learning capability saves valuable time and computational resources. Overall, TensorFlow, with its versatile technological capabilities and the streamlined workflow provided by the tensorflow.keras API, empowers developers and researchers to implement state-of-the-art machine learning technologies with ease. Its robust features make it a preferred choice for various technological applications, driving advancements in the field of deep learning.

Numpy

Numpy is a versatile package for array processing that offers a range of functionalities. It revolves around a high-performance multidimensional array object, providing efficient storage and manipulation of data. Numpy is widely used in scientific computing with Python, serving as a fundamental tool in this domain. Its key features include a powerful N-dimensional array object, advanced functions for broadcasting, support for integrating C/C++ and Fortran code, and capabilities for linear algebra, Fourier transforms, and random number generation. While Numpy is primarily known for its scientific applications, it can also serve as an effective container for generic data. This flexibility allows Numpy to seamlessly and rapidly integrate with various databases, enabling efficient data handling and processing.

Matplotlib

Matplotlib is a powerful 2D plotting library for Python that offers high-quality figures suitable for publication purposes. It supports a wide range of output formats and can be utilized in various environments, including Python scripts, IPython shells, Jupyter Notebooks, web application servers, and graphical user interface toolkits. Matplotlib aims to provide an intuitive and straightforward experience for simple plotting tasks while also offering advanced capabilities for complex visualizations.

With just a few lines of code, you can generate a diverse range of plots, histograms, power spectra, bar charts, error charts, scatter plots, and more. The pyplot module, especially when combined with IPython, provides a user-friendly interface similar to MATLAB. For advanced users, Matplotlib offers extensive customization options, allowing precise control over line styles, font properties, axes properties, and more through an object-oriented interface or MATLAB-like functions. To explore the capabilities of Matplotlib, you can refer to the sample plots and thumbnail gallery provided by the library.

OpenCV

The cv2 (OpenCV) library is a widely used computer vision library in machine learning projects. It provides a comprehensive set of functions and tools for image and video processing, analysis, and computer vision tasks. By importing the cv2 module, developers can leverage its capabilities to perform a variety of operations on visual data. One of the primary uses of cv2 in machine learning projects is image and video input/output. It allows loading and saving images and videos in different formats, providing flexibility in handling diverse datasets. The library supports various image file formats, such as JPEG, PNG, and BMP, as well as video formats like AVI and MP4.

Additionally, cv2 offers a wide range of image processing functions, including filtering, transformations, geometric operations, and color manipulation. These operations are instrumental in preprocessing visual data before feeding it into machine learning models. Examples of such operations include resizing images, applying filters, adjusting brightness and contrast, and extracting image features. Moreover, cv2 provides tools for object detection, image segmentation, and feature extraction, enabling advanced computer vision tasks. It includes pre-trained models and algorithms that can be utilized for tasks like face detection, object recognition, and motion analysis. These capabilities are particularly valuable in building machine learning applications that require visual understanding and interpretation.

Jupyter Notebook

The Jupyter Notebook is a web-based application that allows users to create and share documents containing live code, equations, visualizations, and text. It is an open-source project maintained by Project Jupyter. Originally derived from the IPython project, the Jupyter Notebook was created as a separate project to provide a flexible and interactive computing environment.

The name "Jupyter" is a combination of the core supported programming languages it initially supported: Julia, Python, and R. While it ships with the IPython kernel, which enables programming in Python, Jupyter supports a wide range of programming languages through various kernels. Currently, there are more than 100 kernels available, allowing users to write code in different languages within the Jupyter Notebook environment. The Jupyter Notebook has gained popularity due to its versatility and its ability to seamlessly combine code, visualizations, and explanatory text, making it a powerful tool for data analysis, research, and interactive programming.

4. DESIGN

4.1. Introduction:

In humans and the majority of other animals, the lungs serve as the main respiratory organs. Lungs are present in pairs as shown in figure 3. They perform the gas exchange process in the respiratory system, which involves obtaining oxygen from the airborne gases and delivering it to the bloodstream while releasing carbon dioxide from the bloodstream into the atmosphere. Lung cancer and pneumonia are two respiratory conditions that can harm the tissue of the lungs. Uncontrolled division of aberrant cells in a body part is the primary reason of cancer. Among all malignancies, cancer in the lung is placed 3rd in terms of importance. When gene mutations in the cells' DNA mutate and encourage unnatural grow, lung cancer, also recognized as lung carcinoma, is distinguished by malignant tumors.

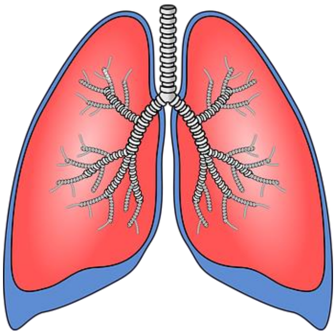


Fig 3: Lung Image

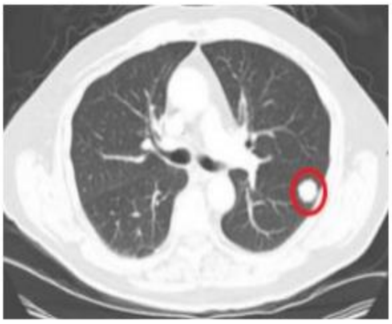


Fig 4: Lung CT Scan with a Nodule

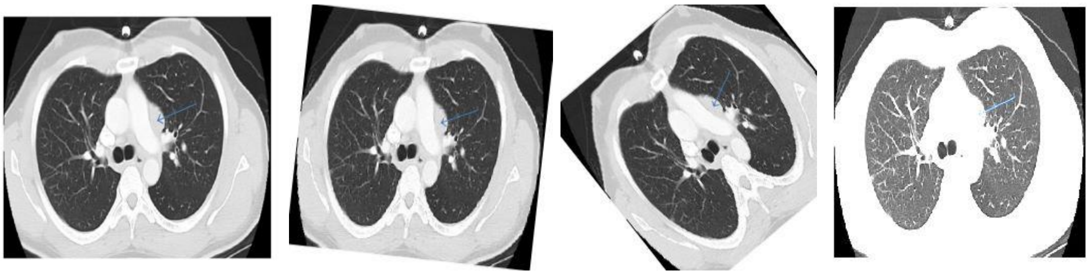
A region of the human body is covered in a sequence of cross-sectional images created by computed tomography (CT). These CT pictures can be analyzed to identify the lung nodules. Identify whether the CT scans are malignant or not. A lung nodule, often known as a "spot on the lungs" or a "coin lesion," is a tiny, circular-shaped tumor in the lung. Nodules are of size less than 3 cm in diameter. As seen in figure 4, a nodule could indeed develop anywhere in the lung.

The growth is referred to as a pulmonary mass and is cancerous if it is greater than that. Lung nodules are typically asymptomatic and can only be seen on computed tomography scans or conventional X-rays. Lung cancer is represented by the lung nodules. The histological investigation, which is invasive and time-consuming, has historically been required to diagnose the pathological type of lung cancer. This conventional method of classifying lung CT pictures is less precise since there aren't as many of them, which gives the radiologist less expertise.

The development of computer-aided systems improves CT scan categorization accuracy by eradicating all shortcomings in the conventional approach.
The target of this is to precisely identify nodules, which will aid in the early detection of cancer and help save lives. Different fully linked deep learning approaches are used to extract the features and categorize them. We would train a convolutional neural network to recognize lung and pulmonary nodule, to categorize the lung as malignant or not, and if cancerous, to determine the type of lung cancer using a publicly accessible dataset of lung CT images comprising of cancerous and non-cancerous lungs.

Deep Learning algorithms are utilized to forecast the nodules in the lung CT scans and nodules discovered are categorized into kinds of lung cancer. Adeno carcinoma originate in the cells that would typically release substances such as mucus. This type of lung cancer primarily affects people who smoke or ex - smokers. Any area of the lung can develop large cell carcinoma. It frequently multiplies and spreads, which might make treatment more challenging. Squamous cells, which are flat cells that line the lining of the lungs' airways, are where squamous cell carcinomas originate. They frequently occur in the center of the lungs, close to a major airway, and are frequently associated with a history of smoking.

Often situations arise where we are shortage of data and hence the model becomes less trustable. Data augmentation is the addition of new data artificially from an existing dataset to strengthen the data before giving it to a model. More the training data better the model learns. This includes from simple flipping the image horizontally, vertically to rotating the image and brightening it to create new images from the existing images as in figure 5. More number of combinations we choose, we get more images.



Lung Image Rotation range: 20 Rotation range: 180 Brightened

Fig 5: Data Augmentation Results for a Single Lung CT Scan

4.2. Architecture Diagram

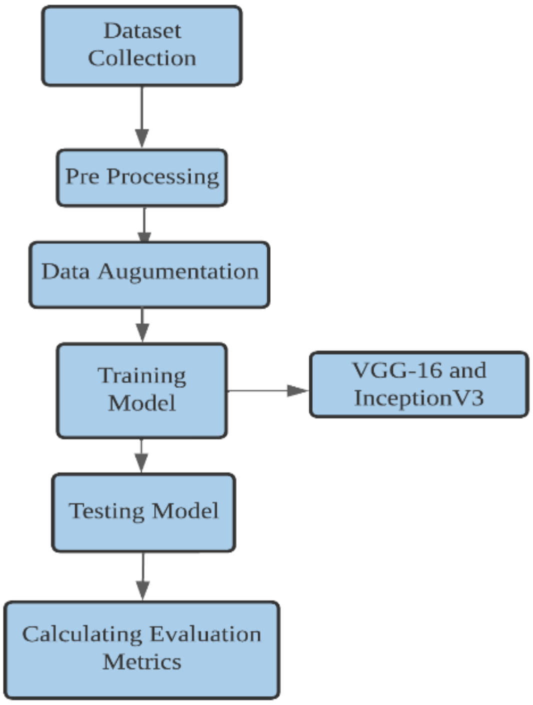
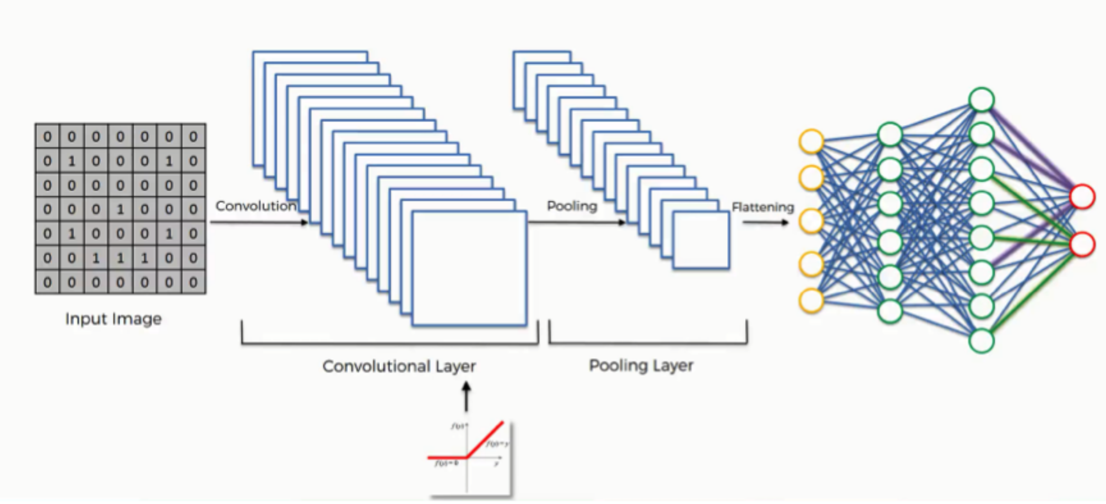


Fig 6: Architecture Design

The proposed model's architecture as in figure 6 can be broken down into six phases or six steps. Data gathering comes first, then data augmentation. Data is gathered from Kaggle [16]. It consists of 1000 images of type PNG taking up to 124 MB memory in which 613 images are used for training, 72 images for validating and 315 images for testing. This is done to increase the dataset in order to boost the model's performance since the more variations there are, the more experiences the model may get. InceptionV3 and VGG16 are utilized to prepare and train the prototype. Then the model is used to test before being made available to the public. Calculating the metrics to determine the degree to which the model findings may be trusted for accuracy.

4.3. Algorithm

4.3.1. VGG16: The acronym VGG stands for visual geometry group. Convolution neural networks (CNNs) are made up of hidden layers, input, and output. A Convolutional Neural Network (CNN) is a deep learning architecture specifically designed for processing and analyzing visual data such as images and videos. Inspired by the structure and functioning of the human visual cortex, CNNs excel at capturing spatial hierarchies and extracting meaningful features from raw pixel data. The key component of CNNs is the convolutional layer, which applies filters to input data, allowing the network to detect various patterns and features at different spatial scales. This hierarchical representation is further enhanced by pooling layers that reduce spatial dimensions while preserving important information. The combination of convolutional and pooling layers enables CNNs as in figure 7 to learn intricate visual representations, making them highly effective for tasks such as image classification, object detection, and image segmentation.



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Fig 7: Convolution neural network

VGG16, short for Visual Geometry Group 16, is a convolutional neural network (CNN) architecture that has made significant contributions to the field of computer vision. It was developed by the Visual Geometry Group at the University of Oxford and was introduced in the paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman in 2014.

One of the key features of VGG16 is its deep architecture, consisting of 16 convolutional layers. This deep structure allows the network to learn intricate and abstract features from images, enabling it to achieve impressive performance on various visual recognition tasks. VGG16's architecture is characterized by the repeated use of 3x3 convolutional filters with a stride of 1, which ensures that the spatial resolution is preserved throughout the network.

The VGG16 architecture can be divided into two main parts: the convolutional part and the fully connected part. The convolutional part consists of thirteen convolutional layers, each followed by a rectified linear unit (ReLU) activation function and a 2x2 max-pooling layer with a stride of 2 as in figure 8. The max-pooling layers reduce the spatial dimensions of the feature maps while preserving the most salient features. The fully connected part consists of three fully connected layers, followed by a softmax activation function for classification.

What sets VGG16 apart from previous CNN architectures is its simplicity. The authors chose to use smaller filter sizes (3x3) and deeper layers instead of larger filter sizes to achieve better performance. This design choice proved to be effective, as VGG16 achieved top results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 in multiple categories.

Despite its remarkable performance, VGG16 is computationally expensive and has a large number of parameters (approximately 138 million), which can make training and deploying the network resource-intensive. However, its architecture serves as a foundation for subsequent developments in the field of deep learning and has inspired various modifications and adaptations, such as VGG19 with 19 layers and the VGG-Face model for facial recognition.

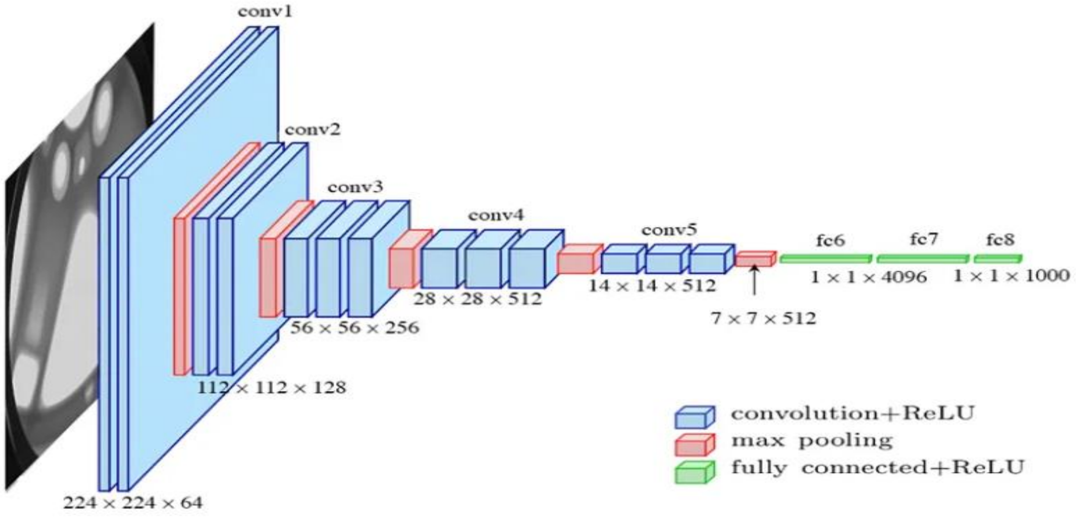


Fig 8: VGG16 Architecture

VGG16's contributions extend beyond its classification performance. The network's learned features have proven to be transferable to other computer vision tasks, such as object localization and detection. Researchers often use pre-trained versions of VGG16 as a starting point for fine-tuning or transfer learning on their specific visual recognition problems.

VGG16 is a deep convolutional neural network architecture that has significantly advanced the field of computer vision. With a focus on smaller filter sizes and deeper layers, has led to outstanding performance on various visual recognition tasks. Although it is computationally demanding, VGG16's impact and influence have been profound, paving the way for subsequent advancements in the field of deep learning.

VGG16's deep architecture allows it to learn highly abstract and discriminative features from medical images, including lung scans. This capability is crucial in detecting subtle patterns or anomalies that might indicate the presence of lung cancer. The network's ability to capture intricate details contributes to its effectiveness in medical image analysis. VGG16 has been pre-trained on a large-scale dataset, such as ImageNet, which consists of a wide range of objects and scenes.

This pre-training allows the network to acquire general visual representations that can be fine-tuned for specific tasks with smaller medical datasets. By leveraging the pre-trained weights, VGG16 can accelerate the training process and improve performance even when the medical dataset is limited. VGG16 has demonstrated strong performance in image classification tasks, achieving high accuracy rates on benchmark datasets. This robustness and accuracy are particularly desirable in medical fields, where accurate predictions are critical for accurate diagnoses and treatment decisions.

Convolution layer: The convolutional layer is considered an essential block of the CNN. Convolutional layers are good for feature extraction from images as they deal with the spatial redundancy by weight sharing. It contains a set of filters (or kernels), parameters of which are to be learned throughout the training. Each filter convolves with the image and creates an activation map. The output of the convolution operation is computed by convolving an input (I) with a number of filters.

Max pooling: Pooling is a down sampling operation that reduces the dimensionality of the feature map as in figure 9. Max Pooling is a convolution process where the Kernel extracts the maximum value of the area it convolves. It calculates the maximum value for each patch of the feature map.

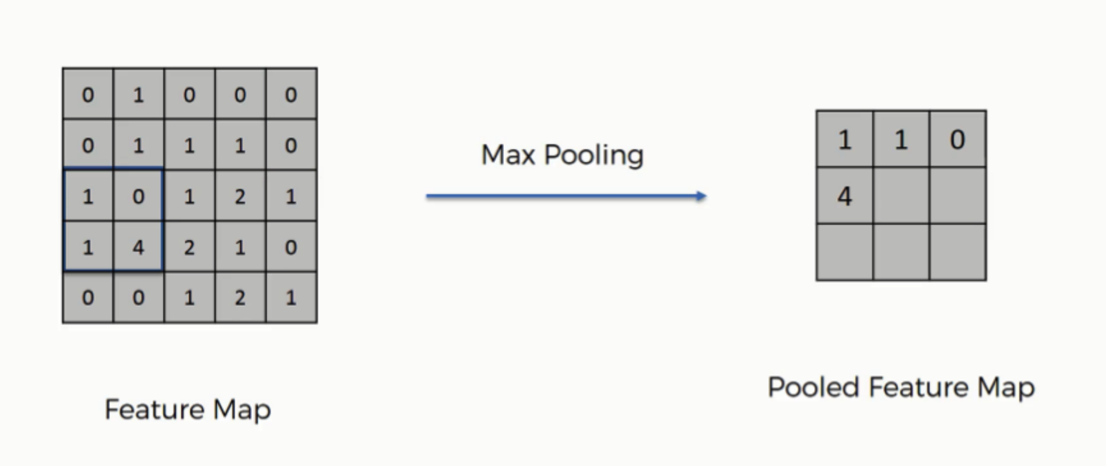


Fig 9: Max Pooling in CNN

ReLU: The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

Fully Connected: This is a simply, feed forward neural network. The **input** to the fully connected layer is the output from the final Pooling or Convolutional Layer as in figure 10. There can be one or more of these layers. Fully connected means that every node in the first layer is connected to every node in the second layer and so on.

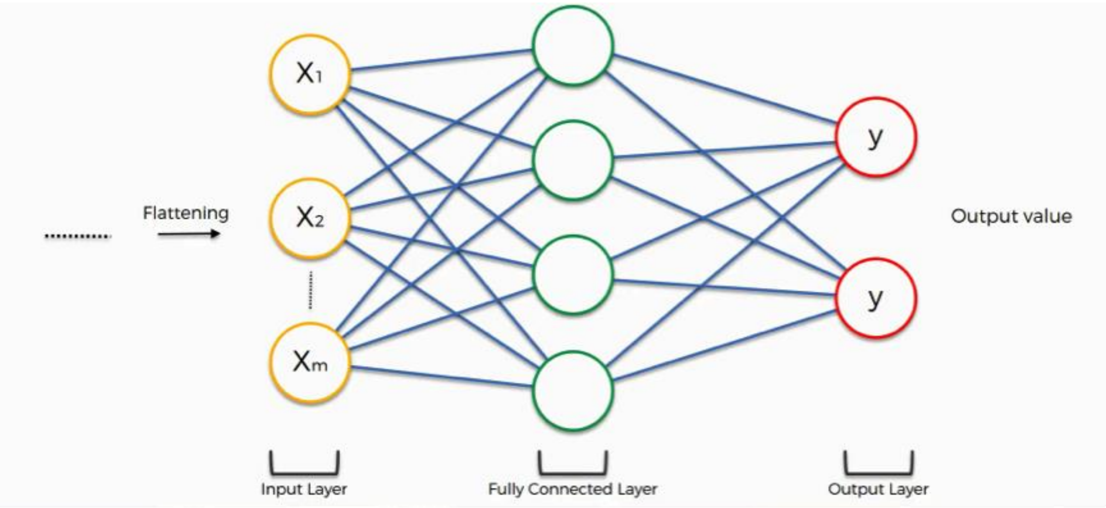


Fig 10: Fully Connected Layer

4.3.2. InceptionV3: Inception models avoid overfitting by employing many filters of various sizes on the same level. Inception models, parallel layers are used in place of deep layers as in figure 11, making the model wider rather than deeper. Version 3 of the Inception model is known as V3. The V3 model of inception technique has overcome a number of drawbacks in the naive model, such as spatial factorization into asymmetric convolutions, which reduces the number of model parameters and shortens processing time as in figure 11.

These deep learning methods are chosen among others based on their practicality and adaptability. After predicting the presence of a lung or pulmonary nodule, the models were eventually trained to classify the test image into different groups.

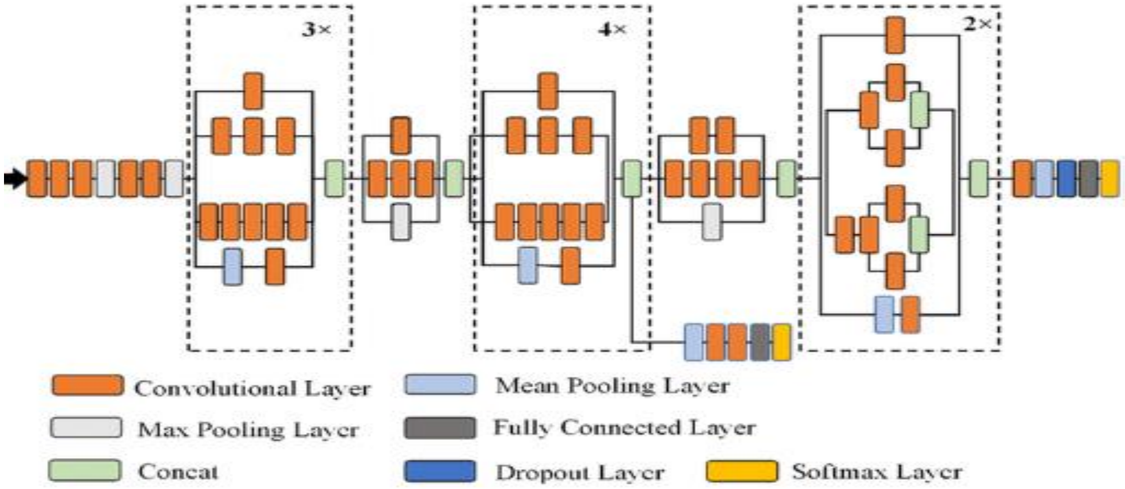


Fig 11: InceptionV3 Architecture

Naïve Form: Different sizes of convolutions are performed to capture different sizes of information in the Picture. 1x1, 3x3, 5x5 size filters are used in the naïve form shown below in figure 12.

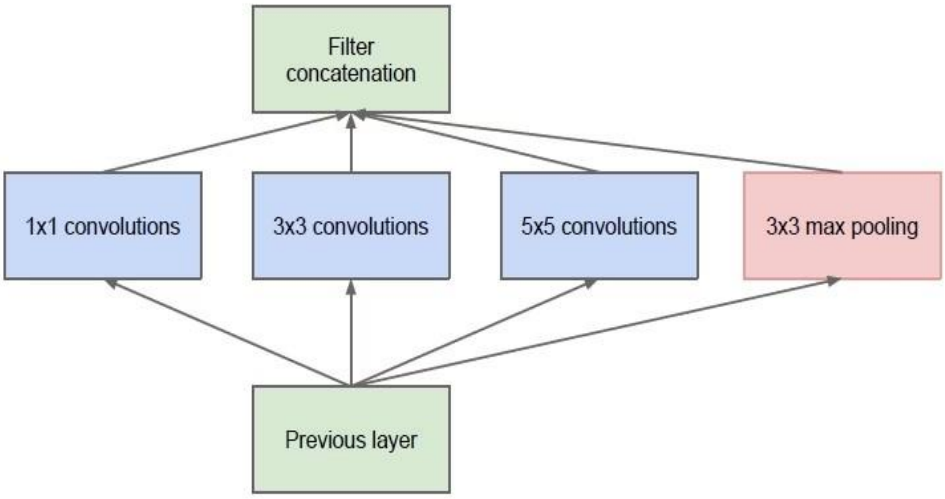


Fig 12: Naïve Form of Inception

A drawback of this naive form is that even the 5×5 convolutional layer is computationally pretty expensive i.e. time-consuming and requires high computational power. Hence 1×1 convolutional layers are added before each convolutional layer as shown in figure 13, which results in reduced dimensions of the network and faster computations.

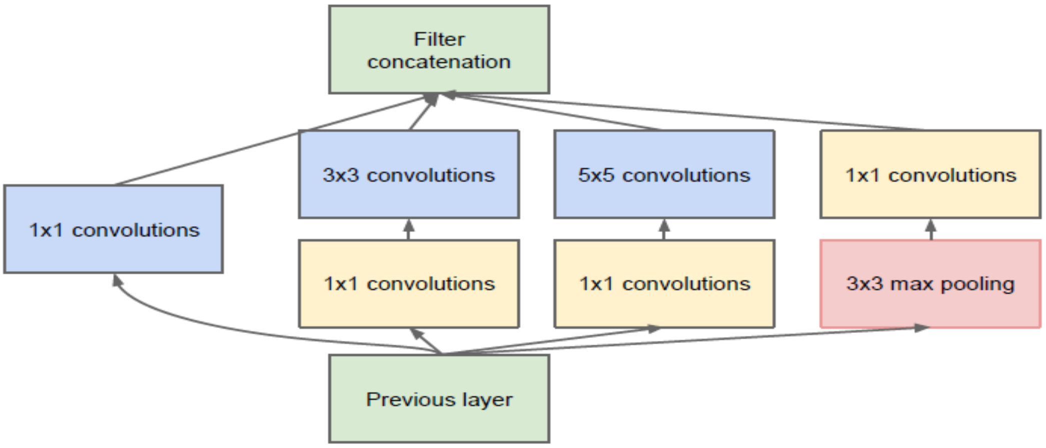


Fig 13: Modified Version of Naïve Form of Inception

InceptionV3 is a modified version of the naïve form. The modifications done are:

Factorization into Smaller Convolutions: 5×5 convolutional layer which is computationally expensive is replaced by two 3×3 convolutional layers as seen in figure 14. This reduces the number of parameters.

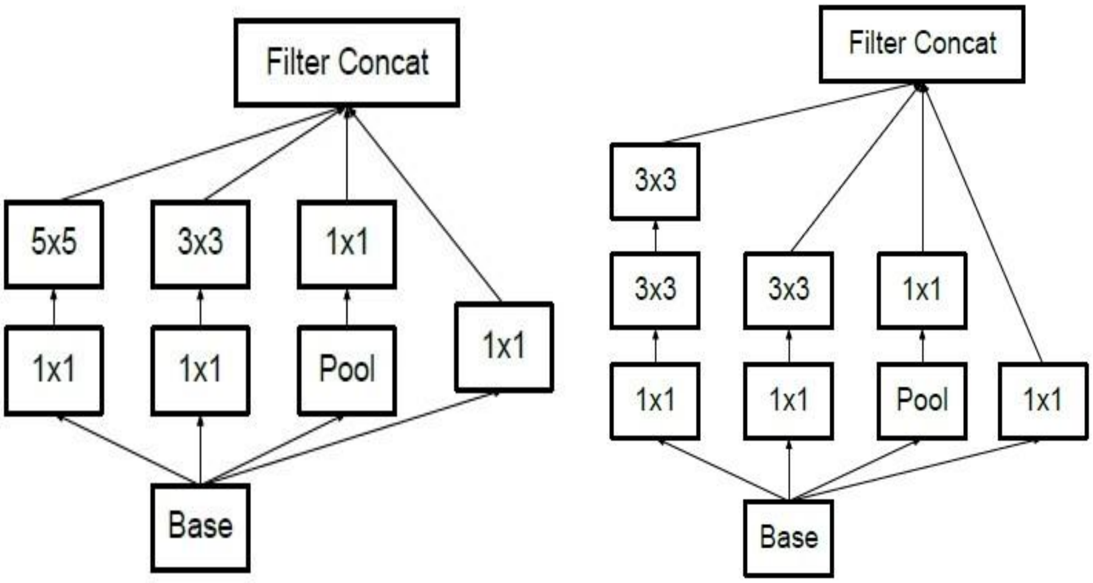


Fig 14: Factorization into Smaller Convolutions

Spatial Factorization into Asymmetric Convolutions: Asymmetric convolutions of the form nx1 can reduce the parameters further. Like replacing a 3x3 filter by 1x3 filter followed by a 3x1 filter as in figure 15.

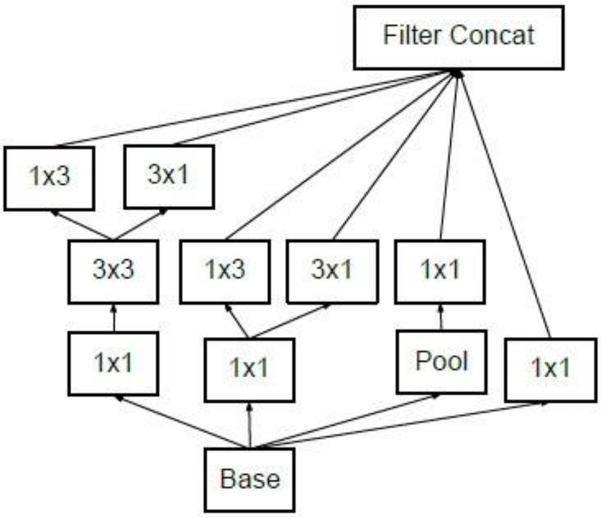


Fig 15: Spatial Factorization into Assymetric Convolutions

Utility of Auxiliary classifiers: The objective of using an Auxiliary classifier is to improve the convergence of very deep neural networks. The auxiliary classifier is mainly used to combat the vanishing gradient problem in very deep networks.

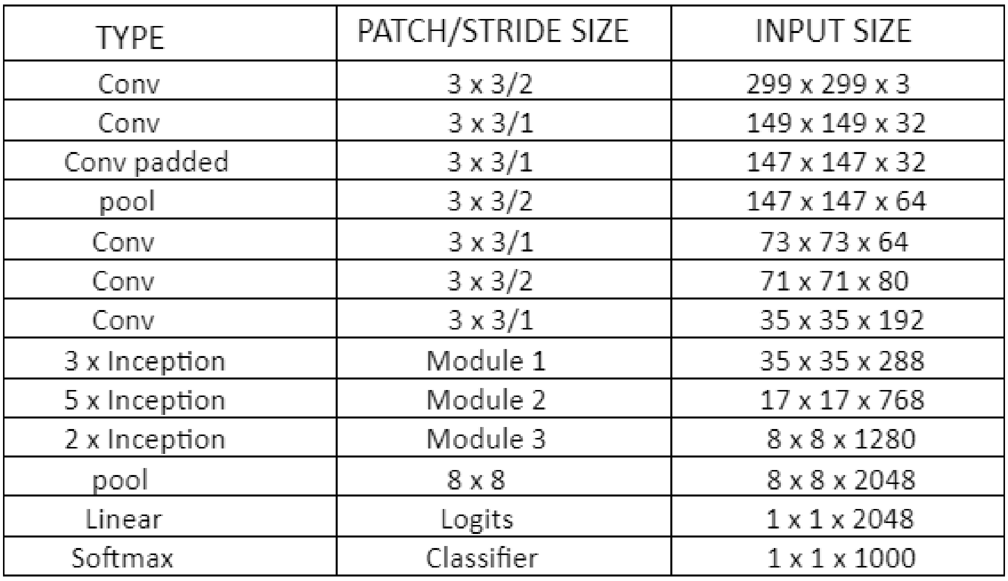
Efficient Grid Size Reduction: Traditionally max pooling and average pooling were used to reduce the grid size of the feature maps. In the inception V3 model, in order to reduce the grid size efficiently the activation dimension of the network filters is expanded.

The inceptionV3 model made up of 42 layers which are a bit higher than the previous inception V1 and V2 models. The various parallel processing layers combinations are named as model A, model B, model C and combinations of these models placed one after the other forms the overall inceptionV3. InceptionV3 is a popular convolutional neural network architecture that has gained significant attention in medical fields and image classification tasks.

It utilizes a unique "Inception module" that incorporates filters of different sizes (1x1, 3x3, 5x5) within a single layer. This allows the network to capture features at multiple scales and resolutions simultaneously. In medical fields, where images may contain important diagnostic information at various scales, InceptionV3's ability to extract multi-scale features can be advantageous for accurate analysis and classification. Despite its deep architecture with multiple layers, InceptionV3 is designed to be computationally efficient. It employs dimensionality reduction techniques, such as 1x1 convolutions, to reduce the number of parameters and computations required. This efficiency is particularly valuable in medical applications, where processing large medical datasets and performing real-time image analysis is crucial.

Medical images often exhibit variations in terms of image quality, patient position, lighting conditions, and other factors. InceptionV3's design with multiple parallel branches and pooling layers helps to capture and incorporate spatial and contextual information from different regions of an image. This robustness to variations enhances its ability to handle diverse medical imaging data. It has demonstrated excellent performance in various image classification challenges and medical applications.

The table describing each layer of the inceptionV3 is:



TYPE	PATCH/STRIDE SIZE	INPUT SIZE
Conv	3 x 3/2	299 x 299 x 3
Conv	3 x 3/1	149 x 149 x 32
Conv padded	3 x 3/1	147 x 147 x 32
pool	3 x 3/2	147 x 147 x 64
Conv	3 x 3/1	73 x 73 x 64
Conv	3 x 3/2	71 x 71 x 80
Conv	3 x 3/1	35 x 35 x 192
3 x Inception	Module 1	35 x 35 x 288
5 x Inception	Module 2	17 x 17 x 768
2 x Inception	Module 3	8 x 8 x 1280
pool	8 x 8	8 x 8 x 2048
Linear	Logits	1 x 1 x 2048
Softmax	Classifier	1 x 1 x 1000

Fig 16: Tabular Description of InceptionV3

The Inception layer is a fundamental component of the InceptionV3 convolutional neural network architecture. It is designed to capture features at multiple scales by employing various filter sizes within a single layer. The Inception layer consists of parallel convolutional branches that process the input feature maps in parallel and then concatenate their outputs. Each branch in the Inception layer performs convolutions with different filter sizes, including

1x1, 3x3, and 5x5 filters. This multi-scale approach allows the network to capture both fine-grained details and larger spatial context simultaneously.

Additionally, the Inception layer incorporates 1x1 convolutions for dimensionality reduction, which helps to reduce the computational complexity of the network. By leveraging these parallel branches with different filter sizes, the Inception layer in InceptionV3 facilitates the extraction of diverse and informative features, contributing to the network's robustness and performance in image classification and other visual recognition tasks.

The softmax layer is the final layer in the InceptionV3 convolutional neural network architecture, responsible for producing the probability distribution over multiple classes. It takes the output of the preceding layers and applies the softmax activation function. The softmax function transforms the input values into a probability distribution, ensuring that the values sum up to 1. Each value in the output represents the predicted probability of the input belonging to a specific class. In the case of InceptionV3, the softmax layer typically outputs probabilities for a predefined set of classes, such as different object categories in image classification tasks. The softmax layer's output allows for easy interpretation and decision-making based on the highest probability class. It enables the network to make predictions by assigning the input to the class with the highest probability. The softmax layer is crucial in obtaining the final predictions from InceptionV3 and plays a vital role in achieving accurate classification results in various visual recognition applications.

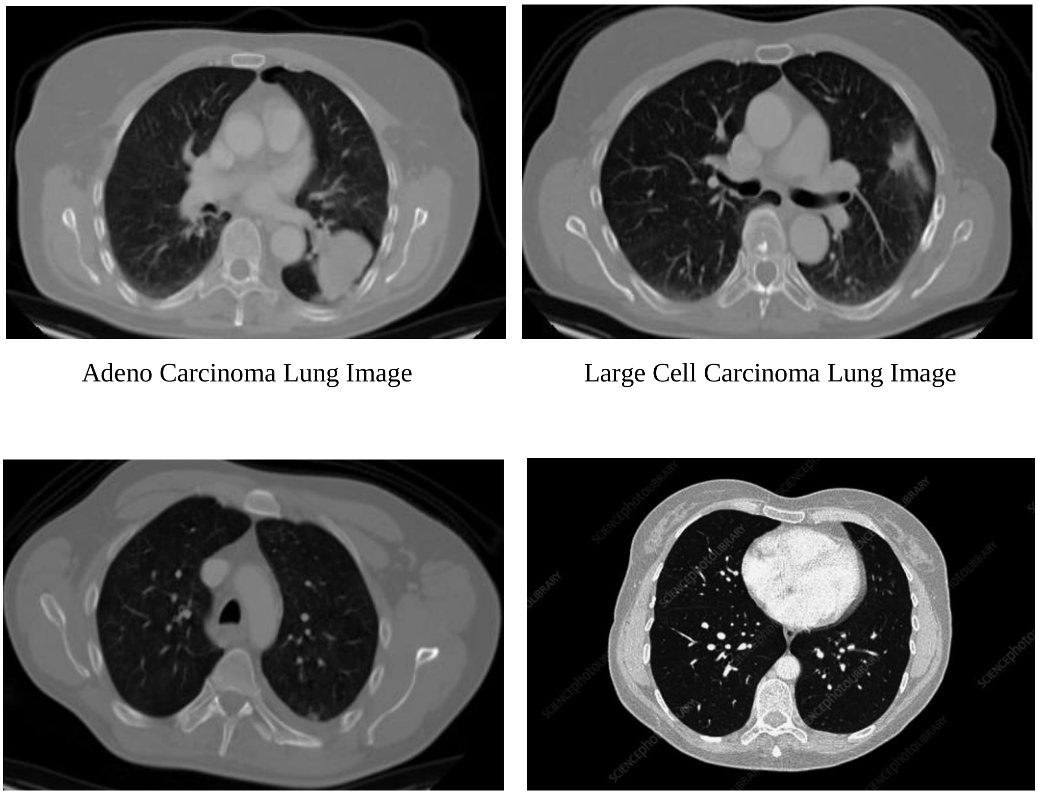
5. IMPLEMENTATION

5.1. Coding

Git link: https://github.com/MogullapalliAkshita/Pulmonary-and-Lungs-Nodule-Classification-Using-Deep-Learning

5.2. Input

The training dataset is obtained from Kaggle. A total of 1000 images are divided as training, testing and validating datasets. Each having sub folders of types of lung cancer images along with non infected lung CT scans. The training set consists of 613 scans in which 148 are healthy lungs, similary test set consists of 315 scans and 54 are normal lungs in that and 13 uninfected lungs in validation dataset of total 72 images. The png extension images are multiplied using data augmentation steps and are given to the model for training. A sample of the images used for training are shown in figure 17.



Adeno Carcinoma Lung Image

Large Cell Carcinoma Lung Image

Squamous Cell Carcinoma Lung Image

Normal Lung Image

Fig 17: Training Images of Various Classes

5.3. Evaluation Metrics

Accuracy is a commonly used evaluation metric to assess the performance of the classification model. Accuracy represents the proportion of correctly classified instances out of the total number of instances in the dataset. It provides a straightforward and intuitive measure of how well the model predicts the presence or absence of lung cancer.

To calculate accuracy as in figure 18, the model's predictions are compared to the ground truth labels of the lung cancer cases. The number of correctly predicted cases is divided by the total number of cases, yielding a value between 0 and 1, which is often expressed as a percentage. A higher accuracy score indicates a more accurate model in correctly identifying lung cancer cases.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Fig 18: Accuracy Formula

Where, TP stands for True Positives

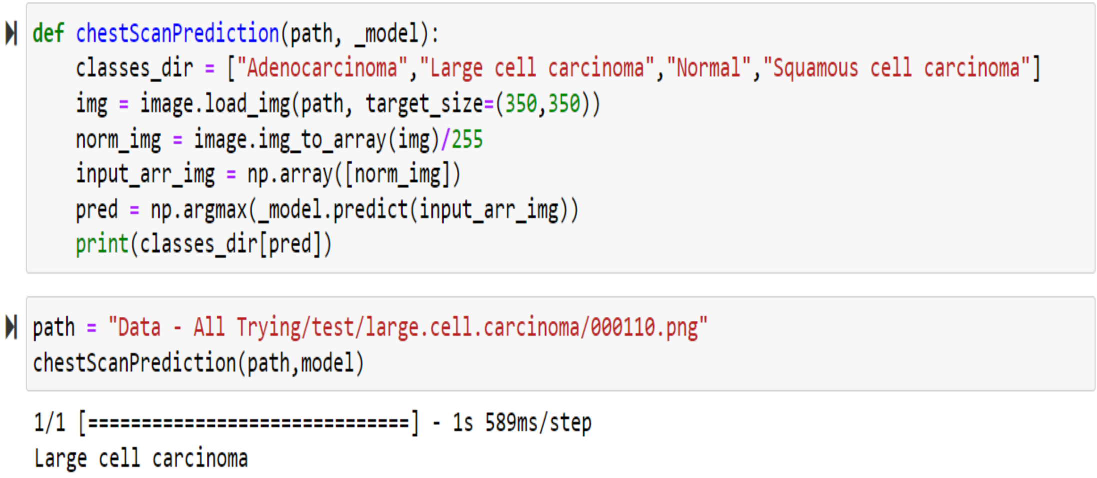
TN stands for True Negatives

FP stands for False Positives

FN stands for False Negatives

5.4. Results

Given the model shown in figure 19, takes an image and the model (VGG16 or InceptionV3) that should be used to predict, it classifies the given scan into its class – adenocarcinoma, large cell carcinoma, normal, squamous cell carcinoma.



```
def chestScanPrediction(path, _model):
    classes_dir = ["Adenocarcinoma","Large cell carcinoma","Normal","Squamous cell carcinoma"]
    img = image.load_img(path, target_size=(350,350))
    norm_img = image.img_to_array(img)/255
    input_arr_img = np.array([norm_img])
    pred = np.argmax(_model.predict(input_arr_img))
    print(classes_dir[pred])
```

```
path = "Data - All Trying/test/large.cell.carcinoma/000110.png"
chestScanPrediction(path,model)
```

```
1/1 [==============================] - 1s 589ms/step
Large cell carcinoma
```

Fig 19: Model Predicting the Image Class

VGG16 model's performance over the epochs is represented in figure 20.

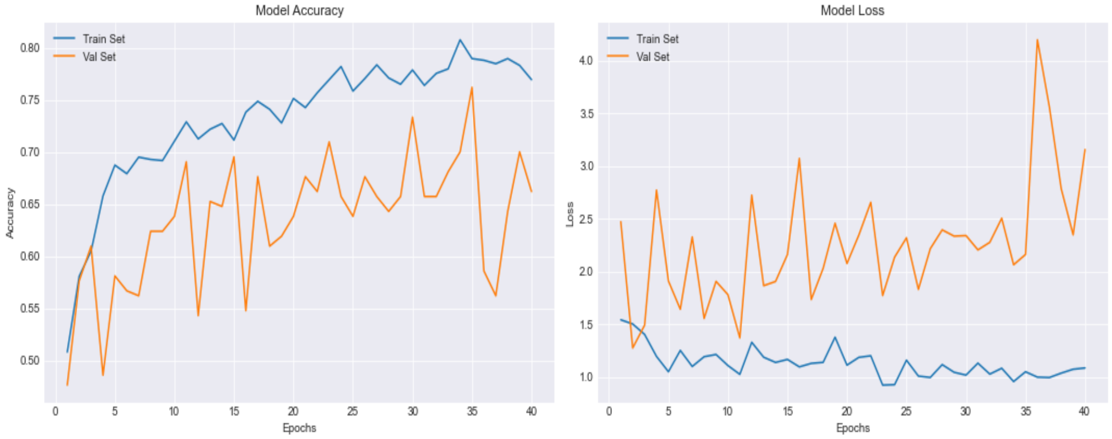


Fig 20: VGG16 Results

InceptionV3 model's performance over the epochs is represented in figure 21.

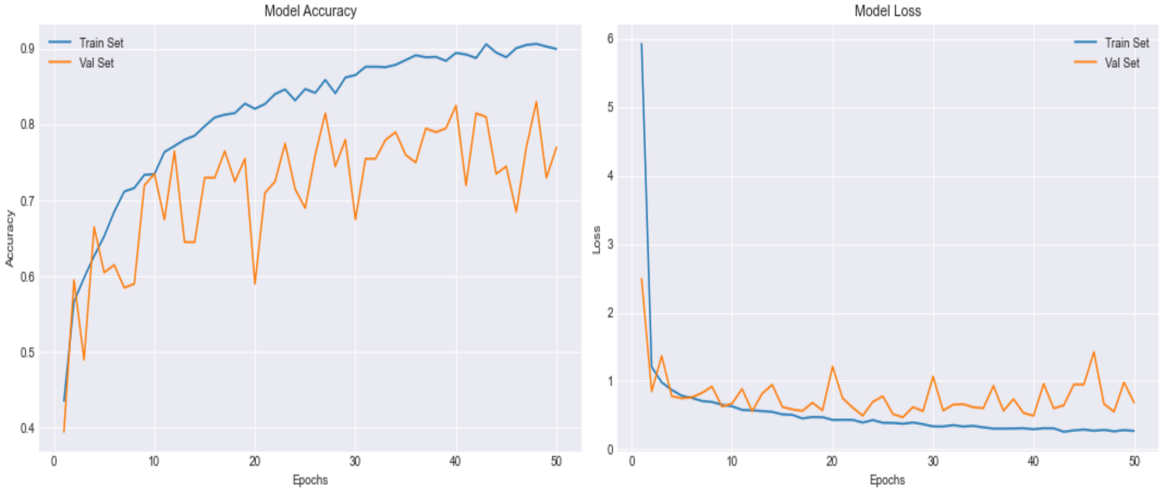


Fig 21: InceptionV3 Results

On the CT images produced through data augmentation, the models are trained and tested. The expected pulmonary and lung nodules are categorized by the models. InceptionV3 performs better than VGG16 in comparison as seen in figure 22.

Models (InceptionV3 and VGG16) provided with test images are categorized into normal, adeno, large, and squamous cell carcinoma classes. The accuracy of the models is compared in the graph. The results of these models can be used accurately to classify and predict lung cancer from CT scan images, this would help radiologists as well as surgeons to treat lung carcinoma in early stage. Predicting the existence of pulmonary nodules or lung nodules makes it easy to identify lung cancer. To make up for the lack of CT pictures, data augmentation is used, and the generated CTs, model are trained, validated, and tested using the produced CT images. The nodules are predicted with the aid of the models VGG-16 and InceptionV3. These computer aided systems help doctors in early diagnosis of lung cancer leading to early stage detection and faster recovery.

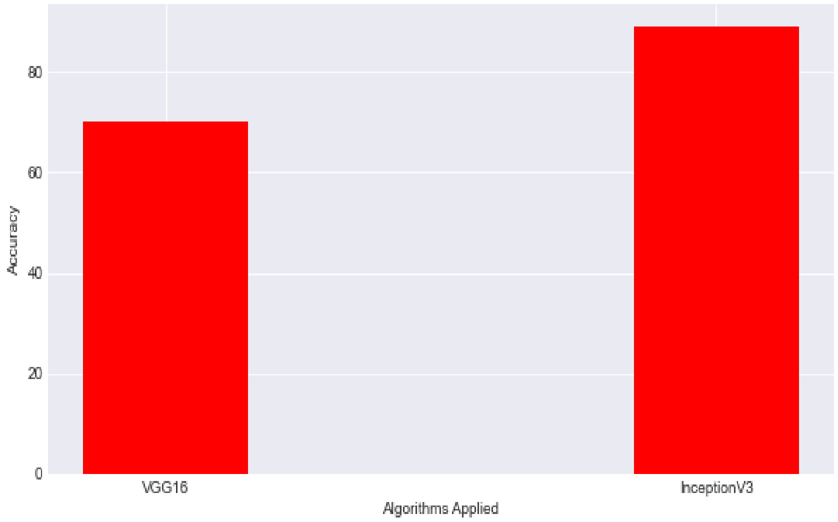


Fig 22: Comparation Between VGG16 and InceptionV3

InceptionV3 has performed better compared to VGG16. The shortcomings of the conventional prediction and categorization of CT scans are overcome by these models, which improve the accuracy of properly predicting the nodules. The anticipated nodules are further divided into adenocarcinoma, large cell carcinoma, and squamous cell carcinoma subtypes of lung cancer.

6. CONCLUSION AND FUTURE SCOPE

Predicting the existence of pulmonary nodules or lung nodules makes it easy to identify lung cancer. To make up for the lack of CT pictures, data augmentation is used, and the generated CTs, model are trained, validated, and tested using the produced CT images. The nodules are predicted with the aid of the models VGG-16 and InceptionV3. These computer aided systems help doctors in early diagnosis of lung cancer leading to early stage detection and faster recovery. InceptionV3 has performed better compared to VGG16. The shortcomings of the conventional prediction and categorization of CT scans are overcome by these models, which improve the accuracy of properly predicting the nodules. The anticipated nodules are further divided into adenocarcinoma, large cell carcinoma, and squamous cell carcinoma subtypes of lung cancer. Future work can be done on huge datasets with various other available deep learning techniques.

REFERENCES

[1] Muhammad Muzammil, Imdad Ali, Ihsan Ul Haq, Amir A. Khaliq, And Suheel Abdullah, “Pulmonary Nodule Classification Using Feature and Ensemble Learning-Based Fusion Techniques”, IEEE Access, vol. 9, pages. 113415 - 113427, 05 August 2021

[2] Ying Chen , Yerong Wang , Fei Hu, Longfeng Feng,Taohui Zhou, And Cheng Zheng,“LDNNET: Towards Robust Classification of Lung Nodule and Cancer Using Lung Dense Neural Network”, IEEE Access, vol. 9, pages. 50301 - 50320, 24 March 2021

[3] Min Li, Xiaojian Ma, Chen Chen, Yushuai Yuan, Shuailei Zhang, Ziwei Yan, Cheng Chen, Fangfang Chen, Yujie Bai, Panyun Zhou, Xiaoyi Lv, And Mingrui Ma, “Research on the Auxiliary Classification and Diagnosis of Lung Cancer Subtypes Based on Histopathological Images”, IEEE Access, vol. 9, pages. 53687 - 53707, 05 April 2021

[4] Qiangchang Wang , Yuanjie Zheng, Gongping Yang , Weidong Jin, Xinjian Chen, and Yilong Yin, “Multiscale Rotation-Invariant Convolutional Neural Networks for Lung Texture Classification”, IEEE journal of biomedical and health informatics, vol. 22, no. 1, January 2018

[5] Aryan Mobiny , Pengyu Yuan, Pietro A. Cicalese , Supratik K. Moulik, Naveen Garg, Carol C. Wu ,Kelvin Wong, Stephen T. Wong, Fellow, IEEE, Tian Cheng He , and Hien V. Nguyen, “Memory-Augmented Capsule Network for Adaptable Lung Nodule Classification”, IEEE transactions on medical imaging, vol. 40, no. 10, October 2021

[6] Pranjal Sahu , Dantong Yu, Mallesham Dasari , Fei Hou , and Hong Qin, “A Lightweight Multi-Section CNN for Lung Nodule Classification and Malignancy Estimation”, IEEE journal of biomedical and health informatics, vol. 23, no. 3, may 2019

[7] Rasool Baghbani, Mohammad Behgam Shadmehr, Masoomeh Ashoorirad, Seyyedeh Fatemeh Molaeezadeh and Mohammad Hassan Moradi, “Bioimpedance Spectroscopy Measurement and Classification of Lung Tissue to Identify Pulmonary Nodules”, IEEE transactions on instrumentation and measurement, vol. 70, 2021

[8] Yoganand Balagurunathan , Andrew Beers, Michael McNitt-Gray , Member, IEEE, Lubomir Hadjiiski, Senior Member, IEEE, Sandy Napel , Life Member, IEEE Dmitry Goldgof , Fellow, IEEE, Gustavo Perez „ “Lung Nodule Malignancy Prediction in Sequential CT Scans: Summary of ISBI 2018 Challenge”, IEEE transactions on medical imaging, vol. 40, no. 12, December 2021

[9] Hong Huang , Member, IEEE, Ruoyu Wu, Yuan Li, and Chao Peng, “Self-Supervised Transfer Learning Based on Domain Adaptation for Benign-Malignant Lung Nodule Classification on Thoracic CT”, IEEE journal of biomedical and health informatics, vol. 26, no. 8, august 2022

[10] Yutong Xie, Yong Xia , Member, IEEE, Jianpeng Zhang, Yang Song, Member, IEEE, Dagan Feng , Fellow, IEEE, Michael Fulham, and Weidong Cai , Member, IEEE, “Knowledge-based Collaborative Deep Learning for Benign-Malignant Lung Nodule Classification on Chest CT”, IEEE transactions on medical imaging, vol. 38, no. 4, april 2019

[11] Stelmo Magalhaes Barros Netto , Joao Otavio Bandeira Diniz, Aristofanes Correa Silva, Anselmo Cardoso de Paiva , Rofolfo Acatauassu Nunes, and Marcelo Gattass, “Modified Quality Threshold Clustering for Temporal Analysis and Classification of Lung Lesions”, IEEE transactions on image processing, vol. 28, no. 4, april 2019

[12] Truc Nguyen and Franz Pernkopf , Senior Member, IEEE, “Lung Sound Classification Using Co-Tuning and Stochastic Normalization”, IEEE transactions on biomedical engineering, vol. 69, no. 9, september 2022

[13] Pin Lim, Student Member, IEEE, Chi Keong Goh, Senior Member, IEEE, and Kay Chen Tan, Fellow, IEEE, “Evolutionary Cluster-Based Synthetic Oversampling Ensemble (ECO-Ensemble) for Imbalance Learning”, IEEE transactions on cybernetics, vol. 47, no. 9, september 2017

[14] Zonghai Zhu, Zhe Wang , Dongdong Li, Yujin Zhu, and Wenli Du, “Geometric Structural Ensemble Learning for Imbalanced Problems”, IEEE transactions on cybernetics, vol. 50, no. 4, april 2020

[15] Geoffrey I. Webb, Member, IEEE, and Zijian Zheng, “Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques”, IEEE transactions on knowledge and data engineering, vol.16,no.8, august 2004

[16] https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images

APPENDIX

Sample Data Augmentation Code

```
datagen = keras.preprocessing.image.ImageDataGenerator(rotation_range=20,
fill_mode='nearest')

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_normal",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/normal",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))

datagen = keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=180,
    width_shift_range=0.2,
    height_shift_range=0.2,
)

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_normal",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/normal",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))
```

```
datagen = keras.preprocessing.image.ImageDataGenerator(rotation_range=20,
fill_mode='nearest')

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_adeno",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/adenocarcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))

datagen = keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=180,
    width_shift_range=0.2,
    height_shift_range=0.2,
)

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_adeno",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/adenocarcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))
```

```
datagen = keras.preprocessing.image.ImageDataGenerator(rotation_range=20, fill_mode='nearest')

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_large",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/large.cell.carcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))

datagen = keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=180,
    width_shift_range=0.2,
    height_shift_range=0.2,
)

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_large",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/large.cell.carcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))

datagen = keras.preprocessing.image.ImageDataGenerator(rotation_range=20, fill_mode='nearest')
```

```
dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_small",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/squamous.cell.carcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))

datagen = keras.preprocessing.image.ImageDataGenerator(
    rescale=1./255,
    rotation_range=180,
    width_shift_range=0.2,
    height_shift_range=0.2,
)

dir_It = datagen.flow_from_directory(
    "Data - All Trying/test_small",
    batch_size=1,
    save_to_dir="Data - All Trying/test_preprocessed/squamous.cell.carcinoma",
    save_prefix="",
    save_format='png',
)
for _ in range(len(dir_It)):
    img, label = dir_It.next()
    plt.imshow((img[0] * 255).astype(np.uint8))
```

Sample Training Code:

```
train_datagen = ImageDataGenerator(rescale = 1.0/255.0,
                                   horizontal_flip = True,
                                   fill_mode = 'nearest',
                                   zoom_range=0.2,
                                   shear_range = 0.2,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   rotation_range=0.4)

train_data = train_datagen.flow_from_directory(train_path,
                                batch_size = batchsize,
                                target_size = (350,350),
                                class_mode = 'categorical')
val_datagen = ImageDataGenerator(rescale = 1.0/255.0)
val_data = val_datagen.flow_from_directory(val_path,
                                batch_size = batchsize,
                                target_size = (350,350),
                                class_mode = 'categorical')
test_datagen = ImageDataGenerator(rescale = 1.0/255.0)
test_data = test_datagen.flow_from_directory(test_path,
                                batch_size = batchsize,
                                target_size = (350,350),
                                class_mode = 'categorical')
```

VGG16:

```
base_model = VGG16(
  weights='imagenet',
  include_top=False,
  input_shape=(350,350,3)
)
NUM_CLASSES = 4
vgg_model = Sequential()
vgg_model.add(base_model)
vgg_model.add(layers.Flatten())
vgg_model.add(layers.Dropout(0.25))
vgg_model.add(layers.Dense(NUM_CLASSES, activation='sigmoid'))

vgg_model.layers[0].trainable = False
vgg_model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)
vgg_model.summary()
```

InceptionV3:

```
base_model = InceptionV3(input_shape = (350, 350, 3),
                         include_top = False,
                         weights = 'imagenet')

for layer in base_model.layers:
    layer.trainable = False
x = layers.Flatten()(base_model.output)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dropout(0.2)(x)

# Add a final sigmoid layer with 4 node for classification output
x = layers.Dense(4, activation='sigmoid')(x)

model_incep = tf.keras.models.Model(base_model.input, x)

model_incep.compile(optimizer =
tensorflow.keras.optimizers.RMSprop(learning_rate=0.0001),
                    loss = 'categorical_crossentropy',
                    metrics = ['accuracy'])
```

Sample Testing Code:

```
from matplotlib import image as imaging
def chestScanPrediction(path, _model):
    i = imaging.imread(path)
    plt.imshow(i)
    plt.show()
    classes_dir = ["Adenocarcinoma","Large cell carcinoma","Normal","Squamous cell
carcinoma"]
    img = image.load_img(path, target_size=(350,350))
    norm_img = image.img_to_array(img)/255
    input_arr_img = np.array([norm_img])
    pred = np.argmax(_model.predict(input_arr_img))
    print(classes_dir[pred])

path = "Data - All Trying/test/large.cell.carcinoma/000126.png"
chestScanPrediction(path,model)
```