A MAJOR PROJECT REPORT ON

# CNN Based Accurate Lung Cancer Classification

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**INFORMATION TECHNOLOGY**

***Submitted by***

**T. Bhumika (20B81A1208)**

**N. Deepika (20B81A1211)**

**K. Jeeva Bindhu (20B81A1219)**

***Under the esteemed guidance of***

**Dr. J. Sengathir**

Professor, IT Department

CVR College of Engineering

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CVR COLLEGE OF ENGINEERING**

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510
2023-2024

## DEPARTMENT OF INFORMATION TECHNOLOGY

## CERTIFICATE

This is to certify that the Project Report entitled **"CNN Based Accurate Lung Cancer Classification "** a bonafide work done and submitted by **T. Bhumika (20B81A1208), N. Deepika (20B81A1211), K. Jeeva Bindhu (20B81A1219)** during the academic year 2023-2024, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

| | |
|---|---|
| **INTERNAL GUIDE** | **HEAD OF THE DEPARTMENT** |
| **Dr. J. Sengathir** | **Dr. Bipin Bihari Jayasingh** |
| Professor, IT Department | HOD & Professor, IT Department |
| **PROJECT COORDINATOR** | **EXTERNAL EXAMINER** |
| **A. Seetharam Nagesh** | |
| Associate Professor, IT Department | |

## ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide **Dr. J. Sengathir,** Professor, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects, **Dr. J. Sengathir,** Professor of Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **A. Seetharam Nagesh,** Project Coordinator, Associate Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances. We also thank the **Project Review Committee Members** for their valuable suggestions.

# DECLARATION

We hereby declare that the project report entitled **"CNN Based Accurate Lung Cancer Classification "** is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Dr. J. Sengathir , Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

Signature of the Student
(T.Bhumika)
(20B81A1208)

Signature of the Student
(N.Deepika)
(20B81A1211)

Signature of the Student
(K.Jeeva Bindhu)
(20B81A1219)

# ABSTRACT

Lung cancer is a type of cancerous development that originates in the cells of the lungs. Although it is commonly associated with smoking, other factors like as genetics or pollution exposure can also cause it to develop in non-smokers. Predicting and Classifying lung cancer helps in early detection, improving treatment outcomes and potentially saving lives by allowing for timely intervention and management of the disease. Early detection can greatly improve a patient's quality of life and enhance the likelihood that their treatment will be successful. Convolutional Neural Networks (CNN), a type of deep learning, have become a highly promising technique for lung cancer prediction from medical imaging data in the past few years. We may be able to improve lung cancer diagnostic efficiency and accuracy by utilizing several CNN models, which could improve patient outcomes and survival rates. In this research, we will employ CNN models, such as VGG16, Resnet152, and Inception V3, to classify and detect lung cancer early on based on CT scan images. A number of performance indicators, including accuracy, precision, F1-score, recall, and support, will be used to assess the models. Our goal is to identify which lung cancer classification model is the most reliable and efficient by methodically examining the output from each model .In the end, this comparative analysis will help choose the best strategy for improving patient care and survival rates by offering insightful information about the pros and cons of each model. The results of this experiment have the potential to significantly improve the diagnostic accuracy of lung cancer and assist health care providers in making decisions.

# LIST OF TABLES

# LIST OF FIGURES

# TABLE OF CONTENTS

# 1. INTRODUCTION

A cancer that develops in the tissues of the lungs, usually in the cells lining the airways, is called lung cancer. It is one of the most prevalent and fatal forms of the disease worldwide, contributing significantly to the death toll from cancer. Due to uncontrolled cell development in the lungs, lung cancer typically affects both men and women. This seriously impairs one's ability to breathe in and out of the chest. The World Health Organization (WHO) states that the main causes of lung cancer are cigarette smoking and passive smoking. Compared to other cancers, the mortality rate from lung cancer is rising daily among both young and old people. Despite the availability of sophisticated medical facilities for precise diagnosis and efficient treatment, the death rate is still not well.Therefore, it is important to take preventative measures as soon as possible so that the disease's signs and effects can be identified early on for a more accurate diagnosis. These days, deep learning has a significant impact on the health care industry due to its high computational capacity for precise data processing and early disease prediction. It can forecast a patient's risk of developing lung cancer using common symptoms like age, sex, wheezing, shortness of breath, and pain in the arm, chest, or shoulder. The project's goal is to provide a model for accurate disease diagnosis and early prediction, which will enable a doctor to save a patient's life. One of the main causes of cancer is lung cancer. One of the main causes of death due to cancer in both men and women is lung cancer. In the majority of cases, the patient's early symptoms indicate the presence of lung cancer.

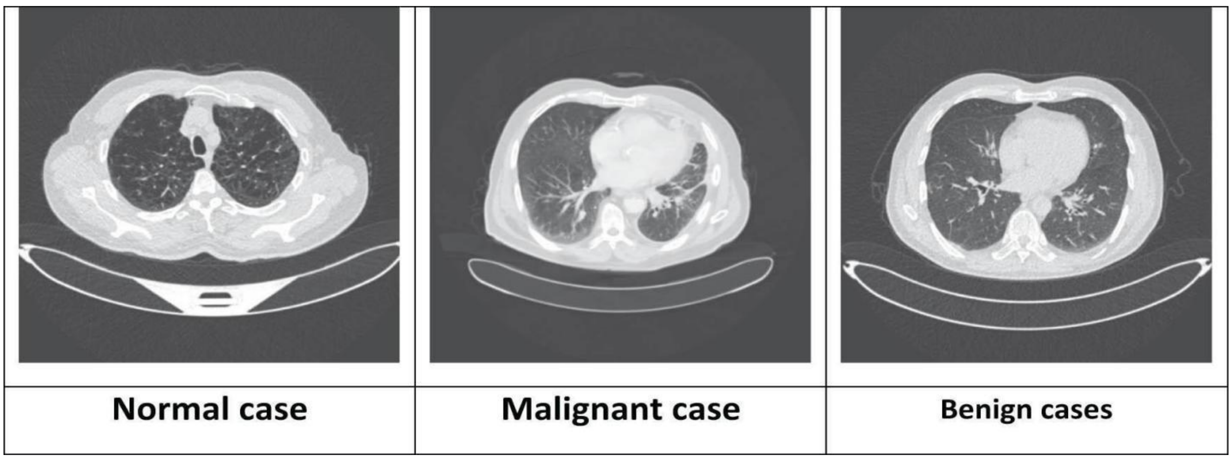


**Figure 1.1 Types of Lung Cancer cases**

**Normal Case:**

- A normal case refers to lung tissue that appears healthy and does not exhibit any signs of abnormal growth or tumor formation.
- In a normal lung, cells follow a regulated pattern of growth, division, and death, maintaining the normal structure and function of the organ.

**Malignant Case:**

- Malignant lung tumors are cancerous growths characterized by uncontrolled and abnormal cell division.
- These cancer cells have the potential to invade nearby tissues and, in the case of lung cancer, may spread to other organs through the bloodstream or lymphatic system.
- Early detection and classification of malignancies are crucial for determining the appropriate treatment strategy and improving the chances of successful outcomes.

**Benign Case:**

- Benign lung tumors are non-cancerous growths that do not spread to other parts of the body.
- These tumors have well-defined borders and do not invade surrounding tissues or organs.
- While benign tumors may grow, they usually do so at a slow pace and are often not life-threatening.
- They are considered localized abnormalities.

Deep learning is one of the fastest-growing topics in medical imaging, with rapidly emerging applications medical image-based and textural data modalities. With the help of deep learning based medical imaging tools, clinicians can detect and classify lung nodules more accurately and quickly. This project presents the recent development of deep learning based imaging techniques for early lung cancer detection. We are using deep learning architecture for image segmentation and object classification in images. Convolutional Neural Network is also known as CNN and ConvNet. CNN is useful for feature extraction and classification of objects in the images. In the realm of medical imaging, where Computed Tomography (CT) scans serve as crucial diagnostic tools. These neural networks, inspired by the details of human vision, excel at automatically extracting detailed patterns and features from the image datasets. The application of CNNs to the analysis of CT scans holds immense promise, as it seeks to unlock a deeper understanding of early stage lung cancer, thus enabling more accurate diagnoses. Hence this project makes use of CNNs to identify patterns within CT images of lung cancer. By training the neural network on various datasets, we aim to differentiate between benign and malignant nodules, ultimately leading to more accurate and early diagnoses.

## 1.1 MOTIVATION

In the olden days, lung cancer diagnosis relied heavily on manual interpretation of imaging studies and histopathological analysis of tissue samples, requiring extensive time and expertise. During this era, the urgent need to improve diagnostic accuracy and streamline treatment decisions in the face of a growing epidemic of lung cancer cases, particularly linked to tobacco smoking. However, there weren't enough automated techniques for identifying lung cancer.One such method is computer-assisted detection (CAD) systems, it involves basic algorithms to assist radiologists in identifying suspicious areas on imaging scans, such as chest X-rays. These early CAD systems often utilized rule-based approaches or simple image processing techniques to highlight potential abnormalities for further review by medical professionals. Another automated method used in the olden days was pattern recognition algorithms, which attempted to identify characteristic patterns associated with lung cancer in medical images. By using complex neural networks to automatically extract and analyze complex patterns from medical imaging data, deep learning techniques have greatly improved the classification of lung cancer. This has led to improved accuracy, earlier detection, and personalized treatment plans, which have ultimately improved patient outcomes and survival rates. In our study, we will use CNN models, such as VGG16, Resnet152, and Inception V3, to classify and detect lung cancer early on from CT scan images. The models will be assessed using a range of performance indicators, including support, recall, F1-score, accuracy, and precision. The model with the highest accuracy will be selected as the most suitable one for classifying lung cancer cases. We are going to categorize the individuals with all three of these types of lung cancer cases utilizing CT scan images. The significance of using Inception V3 lies in its ability to meticulously extract intricate features from lung imaging data, elevating diagnostic accuracy crucial for early cancer detection and treatment optimization. This advanced architecture empowers deep learning models to discern subtle patterns indicative of lung malignancies, ultimately enhancing patient outcomes and survival rates in lung cancer management.

## 1.2 PROBLEM STATEMENT

This problem is concentrated on addressing the critical need for accurate and early detection of lung cancer, a pervasive and deadly disease with significant global impact. By focusing on lung cancer classification, this project seeks to revolutionize the diagnostic process through the application of advanced computational methods, particularly deep learning algorithms, to analyze medical imaging data such as CT scans.

The main goal is to create a strong and reliable classification system that can accurately and efficiently discriminate between benign and malignant lung nodules. Lung cancer survival rates and patient outcomes are greatly increased by timely intervention and treatment, which is made possible by early identification. This project aims to automate the study of medical imaging data by utilizing deep learning methods, such as Convolutional Neural Networks (CNNs), which will enable the quick and accurate diagnosis of worrisome lung nodules.

Furthermore, this research attempts to find out how deep learning algorithms might be utilized for predicting tumor behavior, treatment response, and patient prognosis based on imaging features and molecular profiles. By integrating comprehensive data analysis techniques, it endeavors to provide clinicians with valuable insights into the underlying biology of lung cancer, facilitating more informed treatment decisions and ultimately improving patient outcomes.

# LITERATURE SURVEY

**Zinovev et al. [1] (2011) has proposed a strategy that identifies lung nodule classification with belief decision trees. Engineering in Medicine and Biology Society, EMBC, Annual International Conference of the IEEE, pp.4493–4498. https://doi.org/10.-1109/IEM-BS.2011.6091114**. It was composed of many features, including texture and conjecture, to provide a feature vector of 63 dimensions for the classification. A multiple-label classification algorithm based on belief decision trees was used in this work to predict the distribution of radiologists' opinions. The algorithm was applied to the National Cancer Institute (NCI) Lung Image Database Consortium (LIDC) dataset, which contains semantic annotations by up to four human radiologists for each of the 914 nodules. Additionally, it employs a novel distance-threshold curve technique to analyze multiple-label outcomes, and by calculating the area under this curve, it achieves 69% performance on the validation subset. The experiment's findings show that, in the absence of ground truth, several radiologists' diagnoses on lung CT scans indicate multiple-label classification algorithms. The average accuracy reported by the article is 69% overall. It provided poor precision.

**Roy et al.[2] (2015) proposed Classification of lung image and nodule detection using fuzzy inference system. The IEEE International Conference on Communication, Automation, and Computing, pp. 1204–1207. https://doi.org/10.1109/CCAA.2015.7148560**. Using a fuzzy inference method, Roy et al. proposed classifying lung images and detecting nodules. A method for locating nodules in the lung picture obtained from a CT scan has been proposed by Roy et al. The technique, which uses a fuzzy interference approach and an active contour model, was created for the purpose of detecting lung cancer nodules. This technique uses a grey transition to increase visual contrast. This effective technique involves enhancing the image, using the Active Contour Model to extract the region of interest, extracting spatial features from the segmented image, training the feature vectors, and using the Fuzzy Inference System to classify the test image. An image is binarized before to segmentation, and an active contour model is used to segment the resultant image. The cancer is categorized using fuzzy inference. The classifier is trained using extracted features, which include area, entropy, mean, correlation, main axis length, and minor axis length. This paper improves accuracy and helps radiologists examine lung CT scan images more precisely. When compared to Support Vector Machine, one of the most effective and well-liked methods currently in use, the performance of the suggested method performs better, with an accuracy of 94.12. The inability of this approach to differentiate between benign and malignant tumors is a drawback.

**N.V Ramana Murty and Prof. M.S. Prasad Babu [3] (2017) proposed A Critical Study of Classification Algorithms for Lung Cancer Disease Detection and Diagnosis International Journal of Computational Intelligence Research ISSN 0973-1873, Vol-13, Number 5 , pp. 1041-1048. http://www.ripublication.com.** N.V. Ramana Murty and Prof. M.S. Prasad Babu presented A Critical Analysis of Classification Algorithms for the Identification and Diagnosis of Lung Cancer Disease. Lung cancer prediction utilizing classification algorithms like RBF Neural Network, Multilayer Perceptron, Decision Tree, Naive Bayesian, and C4.5 (J48) method. In order to predict lung cancer, data from 32 instances of cancer and non-cancer patients were first gathered, with 57 attributes. These instances were then pre-processed and analyzed using classification algorithms. Later, 96 instances—86 of which were cancer patients and 10 of which were non-cancer patients—and 7130 attributes underwent the same process. The UCI Machine Learning Repository of Lung Cancer Patients and the Michigan Lung Cancer Patients data set provided the data sets used in this investigation. This paper's primary goals are to give consumers early warning and assess the classification algorithms' performance analysis utilizing WEKA Tool. This paper's low memory usage and short processing time offset its slower processing speed and lower accuracy.

**T. Atsushi et al. [4] (2017) proposed Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks BioMed Research International, pp. 1-6. https://doi.org/10.1155/2017/4067832.** The approach of studying lung malignancies shown in microscopic images has been proposed by T. Atsushi et al. utilizing a deep convolutional neural network (DCNN), a significant deep learning technology. Three convolutional layers, three pooling layers, and two fully linked layers make up the DCNN utilized for classification. The DCNN was trained with our original database and a graphics processing unit in evaluation studies. Prior to being gathered, photos were rotated, flipped, and filtered to avoid overfitting. Microscopic images were first cropped and resampled to produce images with a resolution of 256 × 256 pixels. The dataset contains the probabilities of three different forms of cancer: adenocarcinoma, squamous cell carcinoma, and small cell carcinoma. The developed model had a 71.1% total accuracy rate.

**M. Saric et al.[5] (2019) proposed CNN based Method for Lung Cancer Detection in Whole Slide Histopathology Images 4th International Conference on Smart and Sustainable Technologies, pp. 1-4 . https://doi.org/10.23919/SpliTech.2019.8783041**. A totally automatic method for detecting lung cancer in whole slide photographs of lung tissue samples has been proposed by M. Saric et al. Convolutional neural networks are used for classification at the picture patch level (CNN). The performance of two CNN architectures, ResNet and VGG, is compared after training. The results obtained indicate that the CNN-based technique has the potential to assist pathologists in the identification of lung cancer. Using a receiver operating characteristics (ROC) plot, the output was compared. Patch level accuracy for VGG16 and ResNet50 was found to be 0.7541 and 0.7205, respectively. These are extremely low values. The authors clarified that the high pattern diversity across several slides was the cause of the models' low accuracy.

**Fang Zhou et al. [6] (2019) proposed Evaluate the Malignancy of Pulmonary Nodules Using The 3-D Deep Leaky Noisy-or Network. IEEE Trans. Neural Netw. Learn. Syst.30, pp. 3484- 3494, 2019. https://doi.org/10.1109/TNNLS.2019.2892409.** To overcome this difficulty, Fang Zhou et al. have proposed a 3-D deep neural network technique. There are two modules in the model. The first one is a 3D CNN model that is used in the first module to identify and categorize concerning lesions, or nodules, as either benign or cancerous. The second one determines the chance of lung cancer for the subject by taking the top five nodules based on the detection confidence, assessing their cancer probabilities, and combining them using a leaky noisy-OR gate. In the second module, a leaky noisy model is used to determine the cancer probability for each nodule that is found. A modified Unet design was used for the two modules.The 2017 Data Science Bowl competition produced the dataset. They received classification accuracy scores of 85.96 percent on the training set and 81.42 percent on the testing set, respectively. One shortcoming of the proposed effort is that no consideration is given to the growth rate of nodules.

**Rashidul Hasan et al. [7] (2019) proposed Lung cancer detection and classification based on image processing and statistical learning. J Emerg Trends Eng Appl Sci 11, https://hdl.handle.net/10520/ejc-sl_jeteas-v11-n6-a4** According to Rashidul Hasan et al., a study's goal should be to use image processing methods to identify lung cancer. Cancer patients'

CT-scanned lung images are obtained from the Kaggle Competition dataset. Area of interest is divided using image processing techniques such as preprocessing, segmentation, and feature extraction. Features from each image, such as area, perimeter, and entropy, are extracted in order to develop the algorithm. A doctor's recommended normal values are compared with the parameter values derived from these features. The comparison result indicates the presence of cancer noodles. To scan every picture and show the features and cancerous nodes, a graphical user interface is created. Accurate early identification of lung cancer is possible with the use of this technology. Specific traits were identified using a genetic method, and GLCM was utilized to retrieve the data. SVMs, or support vector machines, were employed to categorize lung cancer stages. To accurately identify whether or not the lungs are cancerous, an algorithm was developed.198 slices of CT images from the Kaggle dataset of cancer patients in various stages were used to evaluate the method. In this dataset, the accuracy of the suggested approach is 72.2%. The use of this approach has the drawback of providing much lesser precision.

**Radhanadh Patra et al.[8] (2020) proposed Prediction of lung cancer using machine learning classifier. International conference on computing science, communication and security, Computer science and communication security, pp 132-142.https://doi.org/10.1007/97-898115-6648-6_11.** A approach has been suggested by Radhanadh Patra et al. to analyze and categorize the existing lung cancer data in the UCI machine learning repository into two categories: benign and malignant. The input data is first pre-processed and transformed to binary form. Next, the data set is classified as cancerous or non-cancerous using a well-known classifier technique in the Weka tool. The comparative method shows that the suggested RBF classifier produced results with an impressive accuracy of 81.25% and is regarded as the most successful classifier method for the prediction of lung cancer data.There is a drawback: the accuracy is reduced.

**Monica Ramakrishnan et al. [9] (2022) proposed Automated Lung Cancer Nodule Detection, Santa Clara University, pp 1–37. https://doi.org/10.3390/s19173722.** A deep artificial neural network architecture, which combines CNN and RNN for fully-automated pulmonary nodule recognition in CT scans, was employed in Monica Ramakrishnan et al.'s technique for identifying lung cancer nodules using CT images. In our scenario, the architecture of the VGG16 convolutional neural network can be used to extract nodule information since it is trained to identify pixels across images. This work uses approaches from machine learning, data mining, and image processing to predict lung cancer nodules in patients at high risk. Our research will show that by utilizing these methods, we can significantly raise the sensitivity of lung nodule detection without raising the false positive rate. Thus, it included an inventive method of implementing CNN using the pre-trained VGG model for feature extraction and RNN for feature classification for identification of pulmonary nodules in lung cancer detection from the accessible LIDC/IDRI dataset, which consisted of about 1500 CT scans. The end-to-end strategy for detecting lung cancer nodules with 70% accuracy was created in this work by merging image processing and classification techniques. The accuracy of the suggested approach in this dataset is 75.6%. Utilizing this strategy has the drawback of offering much less accuracy.

# 2. REQUIREMENT SPECIFICATIONS

## 2.1 HARDWARE & SOFTWARE REQUIREMENTS

In order to perform CT scan-based lung cancer classification, the following essential hardware and software specifications are advised:

**Hardware Requirements:**

System: Pentium i3 Processer

Ram: 6GB

**Software Requirements:**

Operating System: Windows(7 or higher)

Coding Language: Python 3.10.9

Editors: Google Colab, VSCode

## 2.2 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

### Functional:

- The system shall be capable of collecting a diverse dataset of lung CT scans.
- The system shall implement the CNN models: InceptionV3,VGG 16, ResNet152.
- The system will use performance indicators including F1 score, accuracy, precision, and recall to assess the trained models.
- Based on the evaluation measures, the system need to choose the model that performs the best.
- The system should allow users to upload CT images and receive real-time predictions.

### Non-Functional:

- Performance: The system should be capable of handling large datasets and training models efficiently.
- Scalability: For the system to deal with potential future growth in the amount of data and users, the design must be scalable.
- Reliability: The system should be reliable, with minimal downtime and robust error handling.
- Security: The system must ensure the confidentiality and integrity of patient data.
- Maintainability: The code base and system architecture should be well-documented and easily maintainable.

## 2.3 TECHNOLOGY DESCRIPTION

### GOOGLE COLAB:

- Cloud-based Jupyter notebook service by Google. - Provides free access to GPUs and TPUs for deep learning.
- Enables collaboration and sharing of notebooks on Google Drive.
- Pre-installed libraries like TensorFlow and PyTorch.
- Popular for data analysis, machine learning, and sharing code with others.

### PANDAS:

- Data structures like DataFrame and Series are provided by this open-source Python library, which allows for the effective handling of structured data.
- Allows for reading and writing data in a variety of formats, including SQL databases, Excel, and CSV.
- Provides strong functions for aggregation, transformation, and cleansing of data.

**NUMPY:**

- A robust Python library for numerical computations is called NumPy (Numerical Python).
- Large, multi-dimensional arrays and matrices are supported, and a wide range of mathematical operations can be performed on these arrays.
- NumPy is an essential Python library for scientific computing that forms the basis for numerous additional libraries within the data science community.

# 3. DESIGN

## 3.1 USE CASE DIAGRAM

A use case diagram depicts interactions between people (actors) and system functionalities (use cases). Actors represent users or external systems, use cases represent specific functionalities, and relationships show how actors and use cases interact (for example, association, inclusion, and extend). The diagram provides a high-level picture of the system's behavior, which helps stakeholders understand needs and communicate effectively.

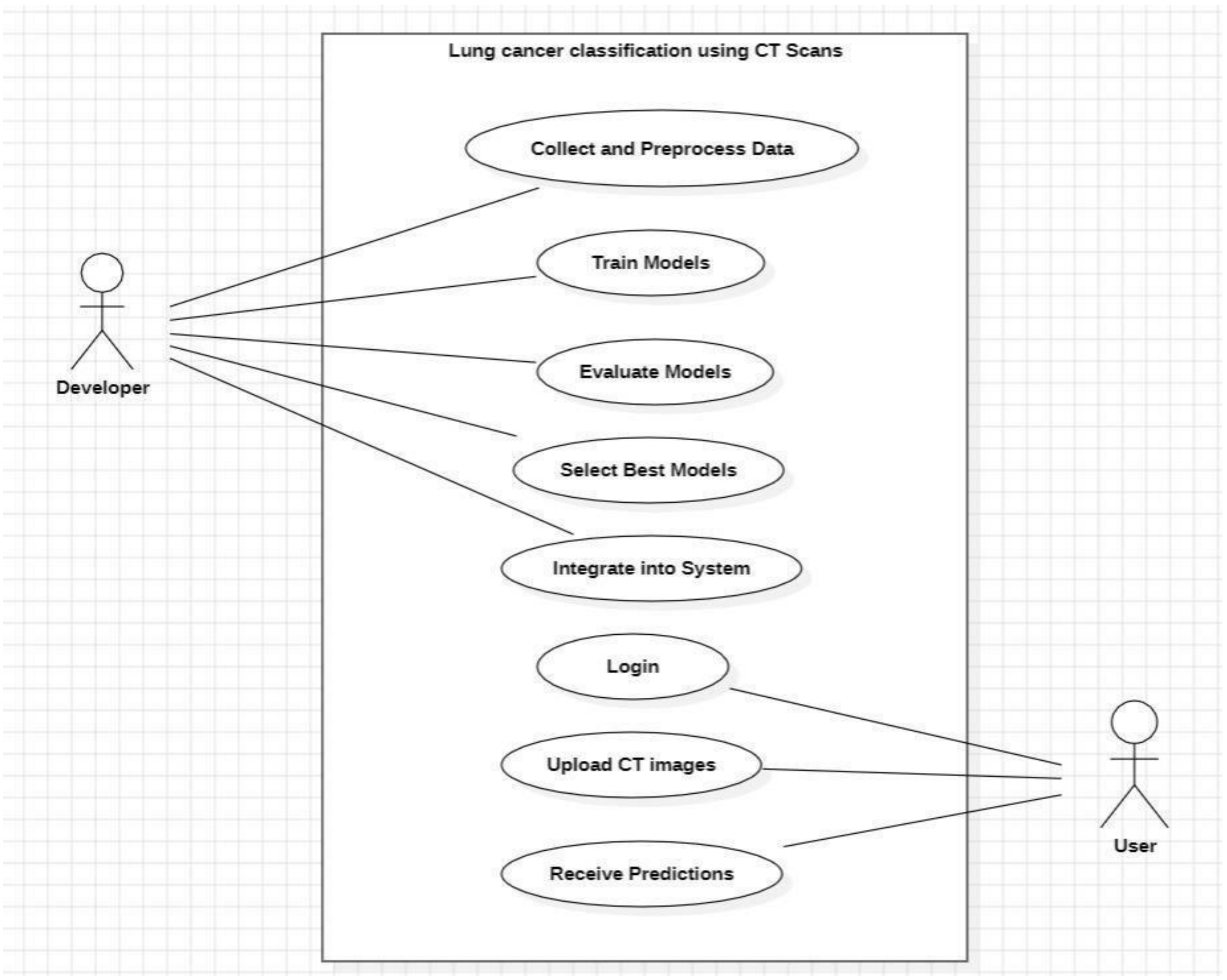


**Figure 3.1: A use case diagram illustrating interactions between a user and a developer.**

## 3.2 SEQUENCE DIAGRAM

A sequence diagram depicts the interactions between components of a system over time, as well as the messages that are transferred between them. Objects are represented as lifelines, and messages as arrows between them, indicating the order of communication. It is especially effective for visualizing the flow of control inside a system and understanding how items work together to complete tasks.

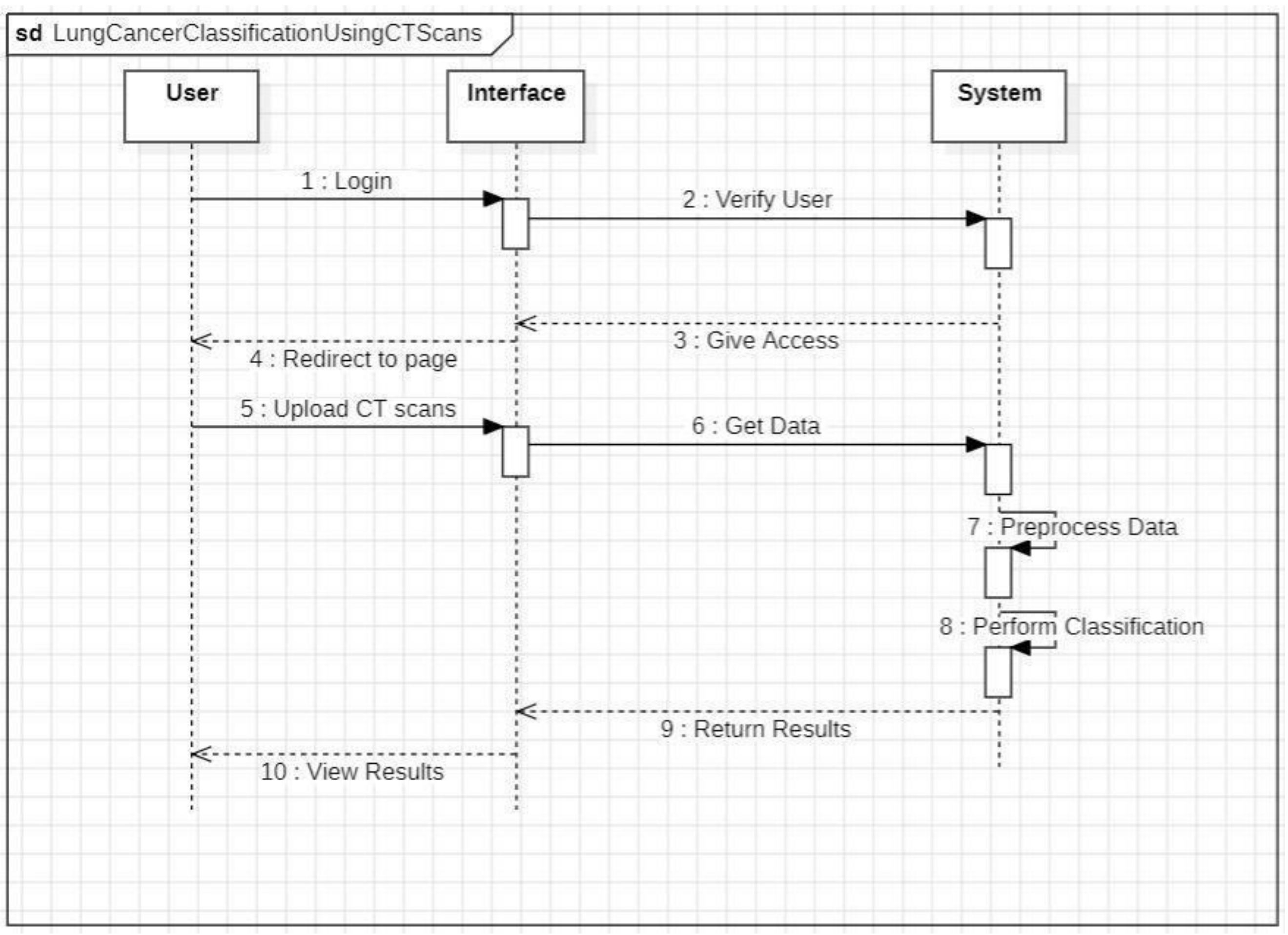


**Figure 3.2: Sequence. Diagram illustrating the interaction of items or components over a specified time sequence.**

## 3.3 ACTIVITY DIAGRAM

An activity diagram depicts the flow of activities within a system or process, indicating the order of actions or phases. It uses symbols such as actions, decisions, and forks to depict the transfer of power from one activity to another. Activity diagrams are especially effective for representing business processes, software workflows, and other sequential activity-based processes.

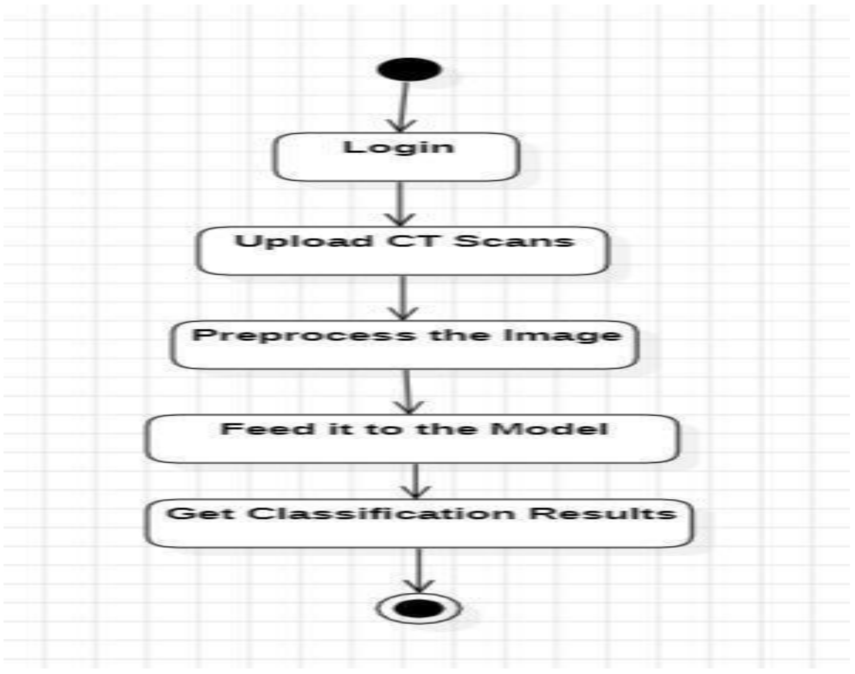


**Figure 3.3: Activity Diagram illustrating the flow of activities or processes within a system.**

## 3.4 ARCHITECTURAL DIAGRAM

For our system, we are using the lung cancer dataset called IQ-OTHNCCD Teaching Hospital/ National Centre for Cancer disease. Firstly, we will pre-process the dataset-resizing the images. We have to split the dataset into Training, Testing datasets. Training Set is fed to the following 3 models. We'll evaluate them with proper evaluation techniques and select the best model. The data/image is pre-processed and given to the final model and classification is done. After this, the results are returned and shown to the end user.

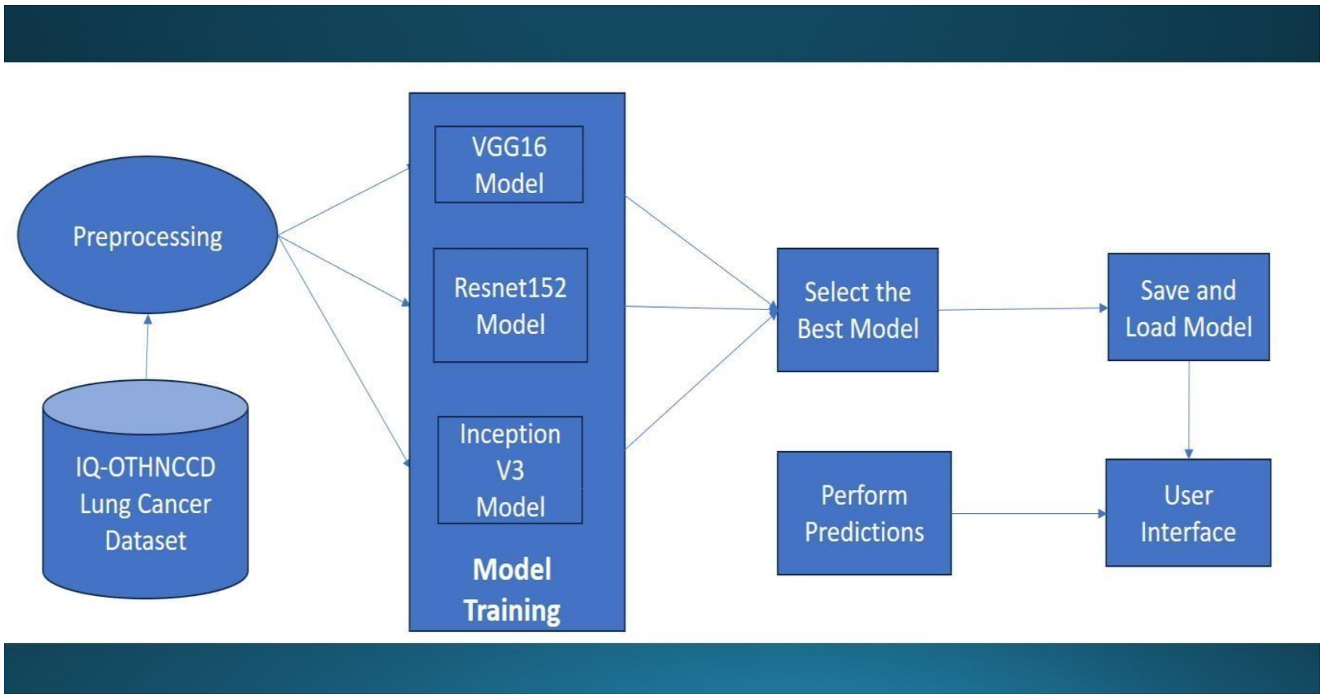


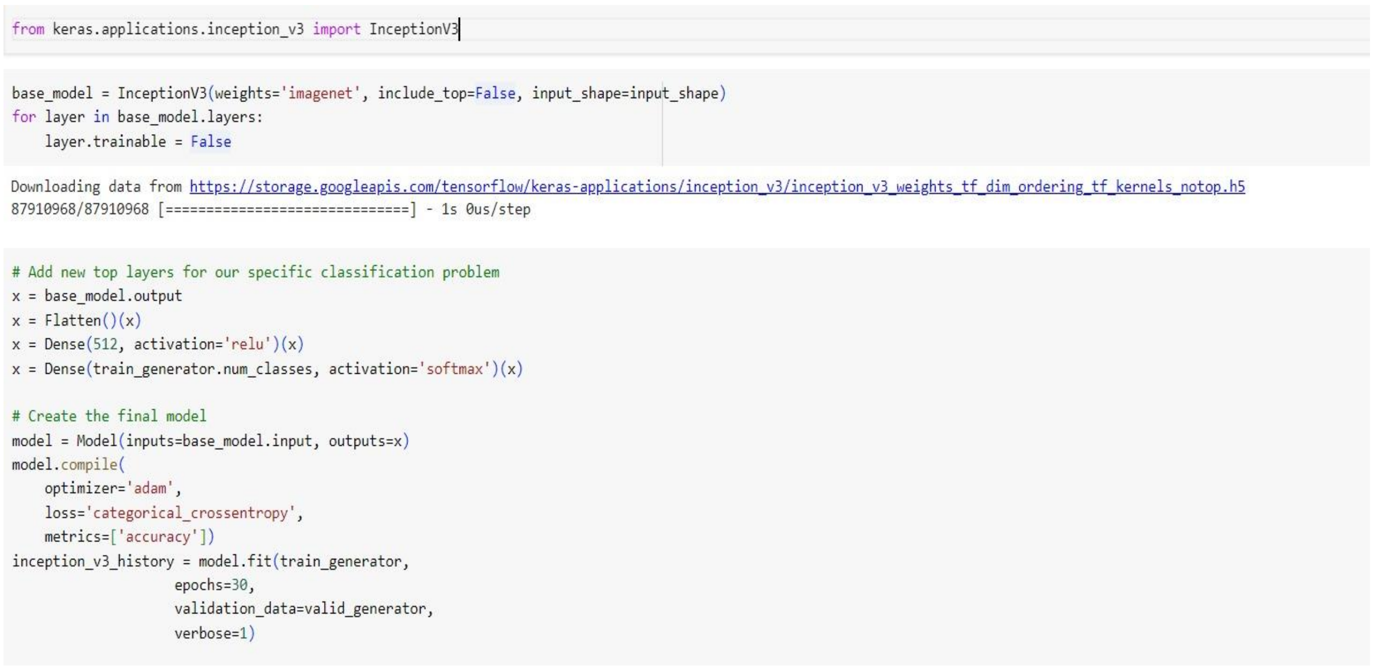
**Figure 3.4: Software Architectural Diagram to represent the high-level structure and components of a system.**

# 4. IMPLEMENTATION

## 4.1 ABOUT THE DATASET

This lung cancer dataset was obtained from the National Center for Cancer Diseases/Iraq-Oncology Teaching Hospital (IQ-OTH/NCCD). It was gathered in the fall of 2019 over a three-month period from the specialized hospitals. It includes CT scans of individuals with lung cancer across multiple phases, together with subjects in good health. In these two centers, radiologists and oncologists marked IQ-OTH/NCCD slides.1190 pictures in all, representing CT scan slices from 110 instances, are included in this dataset. The three groups of these cases are malignant, benign, and normal. Of these, 40 are classified as malignant instances, 15 as benign cases, and 55 as normal cases. Originally, the CT scans were gathered in DICOM format. The CT protocol consists of the following parameters: window width and center vary from 350 to 1200 HU and 120 kV and 1 mm slice thickness, respectively, were used for holding one's breath while fully inspired. Every picture was removed from the database before the analysis. The institutional review board of the cooperating medical centers gave their approval to the study. Every scan comprises many slices. There are between 80 and 200 of these slices, each of which represents a different aspect of angle on a human chest. The 110 instances differ in terms of living situation, age, gender, and level of schooling attained. A portion of them work for the Iraqi ministries of oil and transportation, whereas others are farmers and gainers.

## 4.2 IMPLEMENTATION OF INCEPTION V3



```
from keras.applications.inception_v3 import InceptionV3
```

```
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=input_shape)
for layer in base_model.layers:
    layer.trainable = False
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/inception_v3/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5
87910968/87910968 [==============================] - 1s 0us/step
```

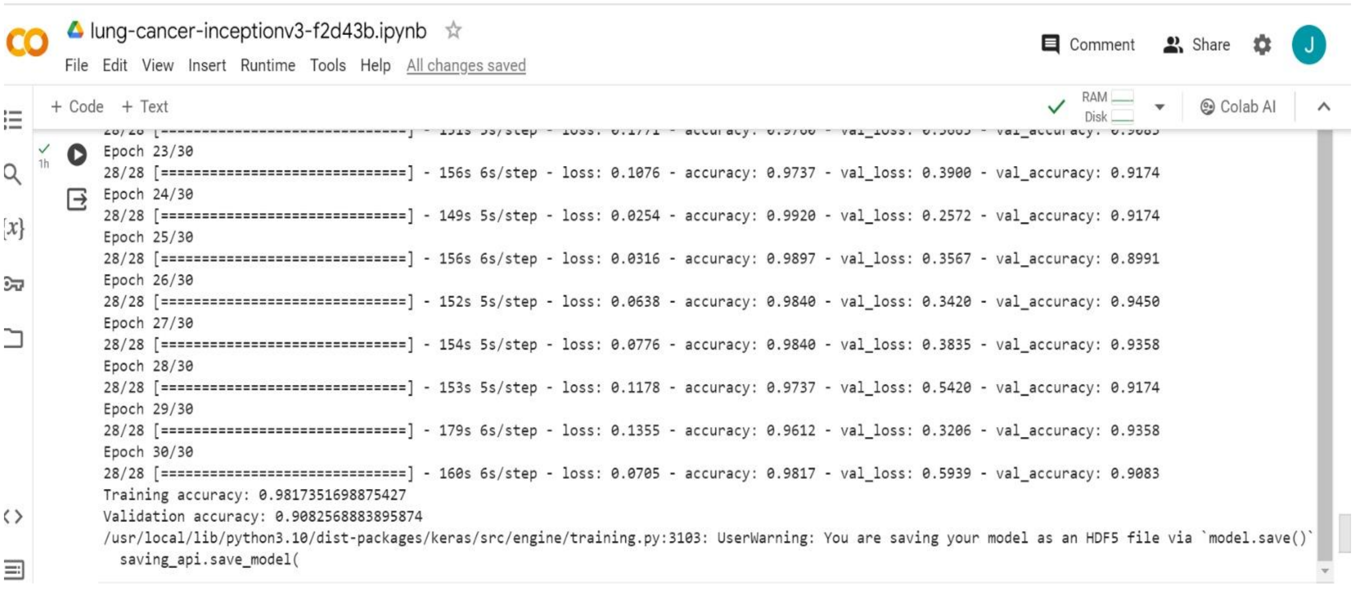
```
# Add new top layers for our specific classification problem
x = base_model.output
x = Flatten()(x)
x = Dense(512, activation='relu')(x)
x = Dense(train_generator.num_classes, activation='softmax')(x)

# Create the final model
model = Model(inputs=base_model.input, outputs=x)
model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy'])
inception_v3_history = model.fit(train_generator,
                    epochs=30,
                    validation_data=valid_generator,
                    verbose=1)
```

**Figure 4.2.1 code snippet of Inception v3**

For the lung cancer classification position, this algorithm makes use of the InceptionV3 pre-trained model, a deep convolutional neural network. Pretrained weights from the ImageNet dataset are initially introduced to the InceptionV3 model, ignoring the top classification layers, while providing the input shape—which has a shape of (224,224,3) as defined in the variable input_shape.To keep the previously learned weights and features, all layers in the basic InceptionV3 model are then set to non-trainable.

On top of the InceptionV3 basis, new top layers are added for the custom lung cancer classification task. First, the output of the base model is obtained, and the Flatten layer is used to flatten it into a one-dimensional tensor. Next, a Dense layer using ReLU and 512 units.In order to extract features from the flattened output, an activation function is applied. Eventually, a Dense layer is added to perform classification into the appropriate classes. It contains the number of classes indicated by train_generator.num_classes and a softmax activation function.

By defining the input and output layers, the Model class is used to build the final model. After that, it is compiled using the Adam optimizer, accuracy as the evaluation metric, and the categorical cross-entropy loss function (assuming a multiclass classification problem). The fit method is known as to validate the model's performance using the validation data provided by valid_generator, and then train the model for 30 epochs on the training data provided by train_generator. The method of training is verbose and displays the results as it goes.


lung-cancer-inceptionv3-f2d43b.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Colab AI

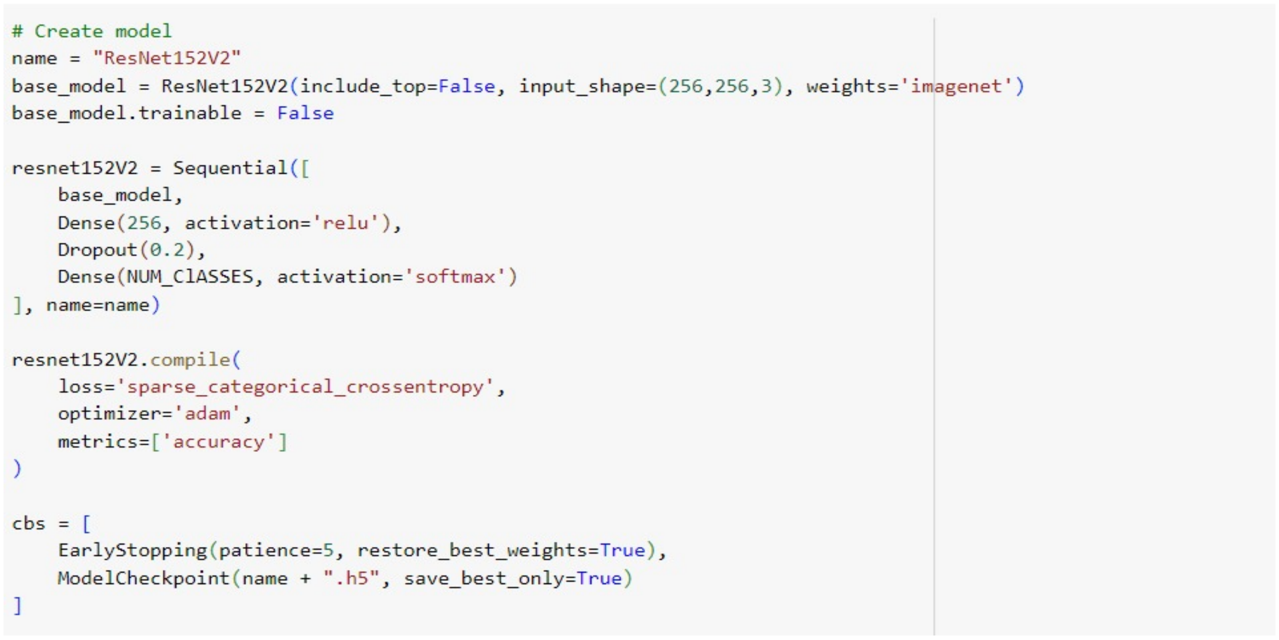
```
Epoch 23/30
28/28 [==============================] - 156s 6s/step - loss: 0.1076 - accuracy: 0.9737 - val_loss: 0.3900 - val_accuracy: 0.9174
Epoch 24/30
28/28 [==============================] - 149s 5s/step - loss: 0.0254 - accuracy: 0.9920 - val_loss: 0.2572 - val_accuracy: 0.9174
Epoch 25/30
28/28 [==============================] - 156s 6s/step - loss: 0.0316 - accuracy: 0.9897 - val_loss: 0.3567 - val_accuracy: 0.8991
Epoch 26/30
28/28 [==============================] - 152s 5s/step - loss: 0.0638 - accuracy: 0.9840 - val_loss: 0.3420 - val_accuracy: 0.9450
Epoch 27/30
28/28 [==============================] - 154s 5s/step - loss: 0.0776 - accuracy: 0.9840 - val_loss: 0.3835 - val_accuracy: 0.9358
Epoch 28/30
28/28 [==============================] - 153s 5s/step - loss: 0.1178 - accuracy: 0.9737 - val_loss: 0.5420 - val_accuracy: 0.9174
Epoch 29/30
28/28 [==============================] - 179s 6s/step - loss: 0.1355 - accuracy: 0.9612 - val_loss: 0.3206 - val_accuracy: 0.9358
Epoch 30/30
28/28 [==============================] - 160s 6s/step - loss: 0.0705 - accuracy: 0.9817 - val_loss: 0.5939 - val_accuracy: 0.9083
Training accuracy: 0.9817351698875427
Validation accuracy: 0.9082568883895874
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`
  saving_api.save_model(
```

**Figure 4.2.2 Training of Inception v3**

We obtained training accuracy of 0.98 and validation accuracy of 0.90 by training this model. These results indicate that the model performs well on training data and performs slightly worse yet still well on validation data.

## 4.3 IMPLEMENTATION OF RESNET152

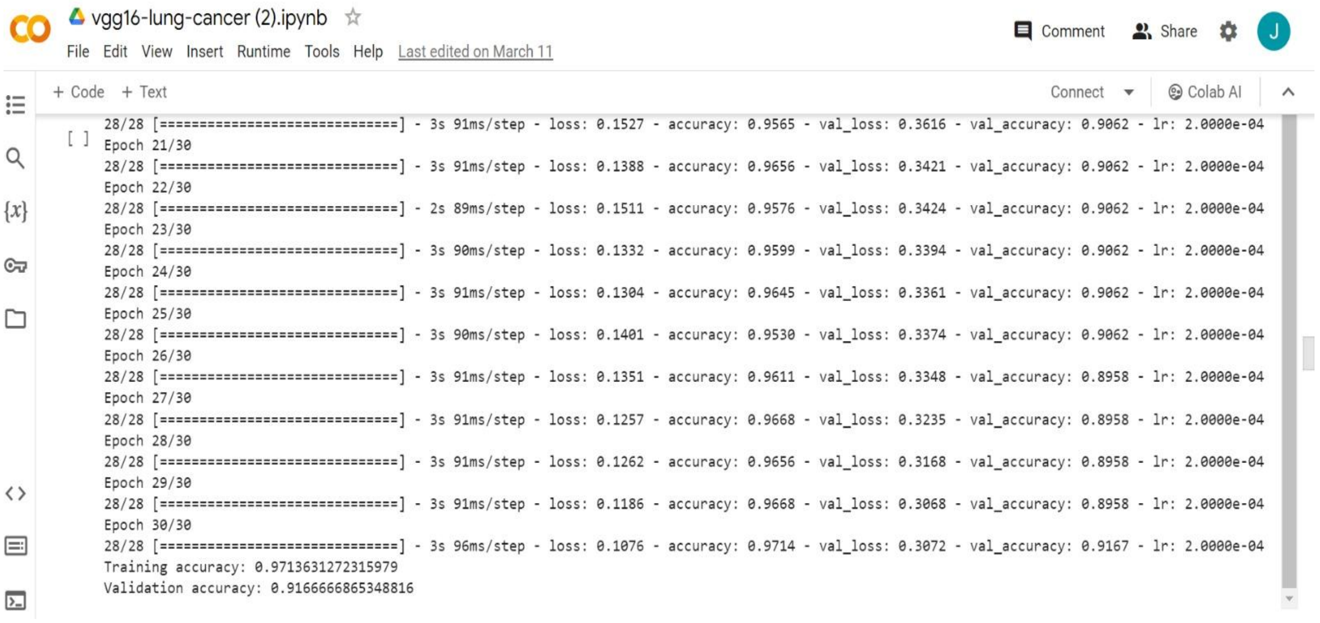
This code segment creates a convolutional neural network (CNN) model based on the ResNet152V2 architecture for lung image classification tasks. It first loads the ResNet152V2 architecture without its classification layers, setting the input shape to 256x256 pixels with 3 color channels (RGB) and initializing the model with pre-trained weights from the ImageNet dataset.



```
# Create model
name = "ResNet152V2"
base_model = ResNet152V2(include_top=False, input_shape=(256,256,3), weights='imagenet')
base_model.trainable = False

resnet152V2 = Sequential([
    base_model,
    Dense(256, activation='relu'),
    Dropout(0.2),
    Dense(NUM_ClASSES, activation='softmax')
], name=name)

resnet152V2.compile(
    loss='sparse_categorical_crossentropy',
    optimizer='adam',
    metrics=['accuracy']
)

cbs = [
    EarlyStopping(patience=5, restore_best_weights=True),
    ModelCheckpoint(name + ".h5", save_best_only=True)
]
```

**Figure 4.3.1 Code snippet of Resnet152**

The weights of the base model are then frozen to prevent updates during training. The main model is constructed using a Sequential container, where the base ResNet152V2 model is followed by a Dense layer with 256 units and ReLU activation for feature extraction, a Dropout layer with a dropout rate of 0.2 to reduce overfitting, and a final Dense layer with a softmax activation function for classification into NUM_CLASSES categories.The accuracy metric, Adam optimizer, and sparse categorical cross-entropy loss are used to assemble the model. Two callbacks are defined in addition:The EarlyStopping and ModelCheckpoint preserve the best-performing model in a file called "ResNet152V2.h5" and stop training if validation loss does not improve after five epochs.


vgg16-lung-cancer (2).ipynb
File Edit View Insert Runtime Tools Help Last edited on March 11
Comment Share
+ Code + Text
Connect
Colab AI

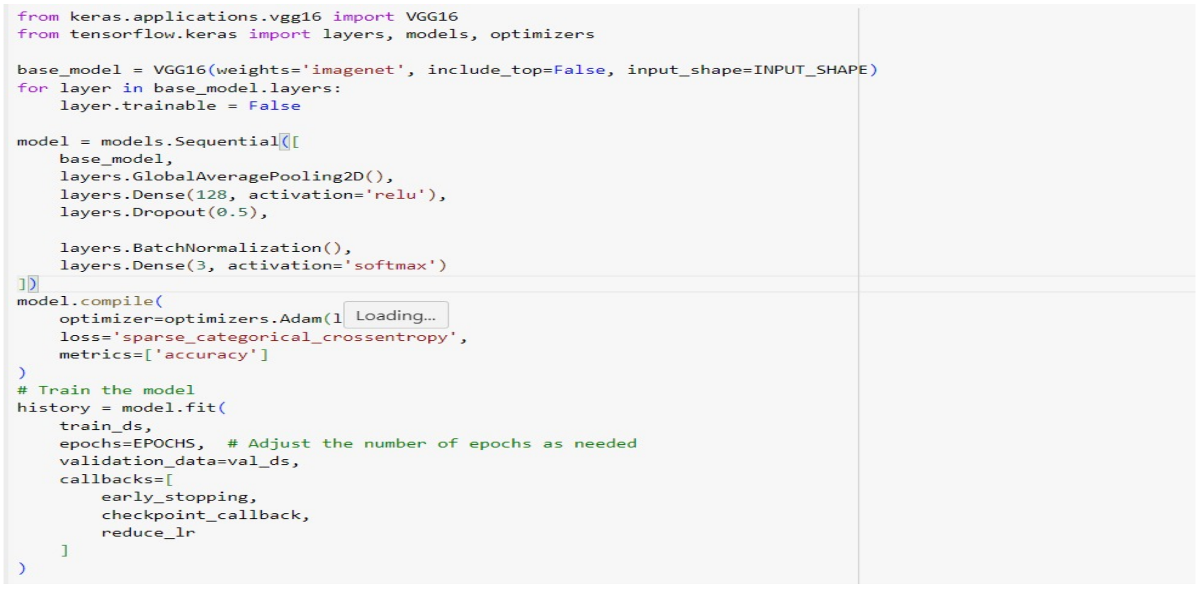
```
28/28 [==============================] - 3s 91ms/step - loss: 0.1527 - accuracy: 0.9565 - val_loss: 0.3616 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 21/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1388 - accuracy: 0.9656 - val_loss: 0.3421 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 22/30
28/28 [==============================] - 2s 89ms/step - loss: 0.1511 - accuracy: 0.9576 - val_loss: 0.3424 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 23/30
28/28 [==============================] - 3s 90ms/step - loss: 0.1332 - accuracy: 0.9599 - val_loss: 0.3394 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 24/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1304 - accuracy: 0.9645 - val_loss: 0.3361 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 25/30
28/28 [==============================] - 3s 90ms/step - loss: 0.1401 - accuracy: 0.9530 - val_loss: 0.3374 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 26/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1351 - accuracy: 0.9611 - val_loss: 0.3348 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 27/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1257 - accuracy: 0.9668 - val_loss: 0.3235 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 28/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1262 - accuracy: 0.9656 - val_loss: 0.3168 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 29/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1186 - accuracy: 0.9668 - val_loss: 0.3068 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 30/30
28/28 [==============================] - 3s 96ms/step - loss: 0.1076 - accuracy: 0.9714 - val_loss: 0.3072 - val_accuracy: 0.9167 - lr: 2.0000e-04
Training accuracy: 0.9713631272315979
Validation accuracy: 0.9166666865348816
```

**Figure 4.3.2 Training of Resnet152**

We obtained training accuracy of 0.97 and validation accuracy of 0.91 by training this model. These results indicate that the model performs well on training data and performs slightly worse yet still well on validation data.

## 4.4 IMPLEMENTATION OF VGG 16

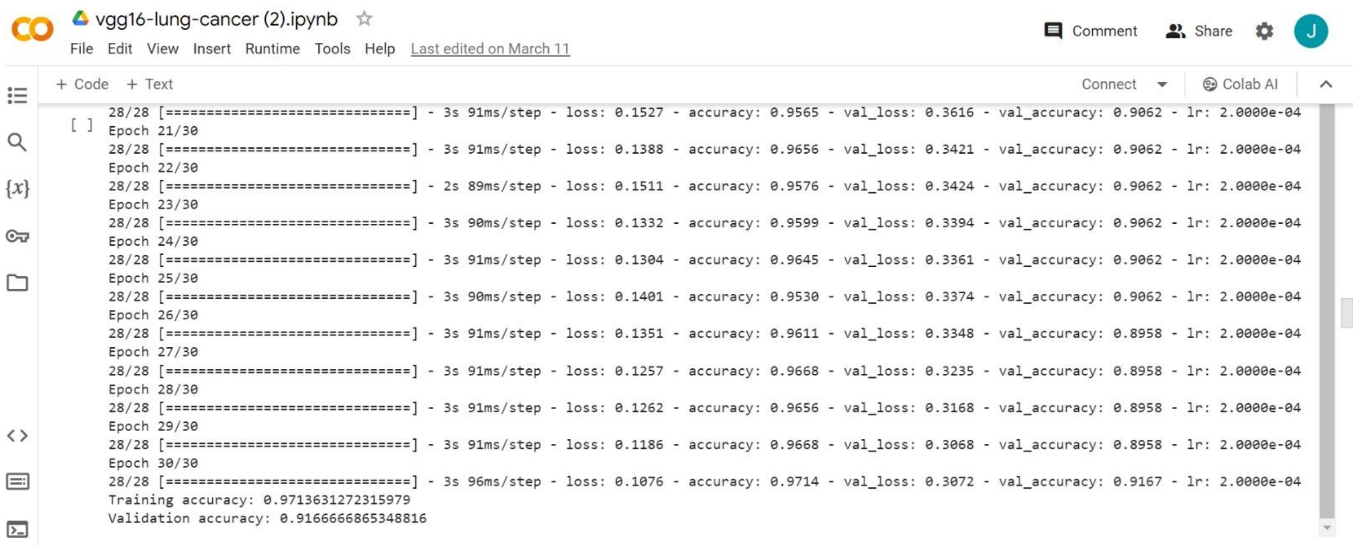
The pre-trained VGG16 model is used by this code to perform picture classification tasks. The first step is to load the VGG16 architecture without its upper classification layers, defining INPUT_SHAPE as the input shape, and initializing the model using ImageNet dataset pre-trained weights.To maintain the pretrained features, it then freezes the weights of every layer in the base VGG16 model. Next, a Sequential container is used to construct the main model.



```
from keras.applications.vgg16 import VGG16
from tensorflow.keras import layers, models, optimizers

base_model = VGG16(weights='imagenet', include_top=False, input_shape=INPUT_SHAPE)
for layer in base_model.layers:
    layer.trainable = False

model = models.Sequential([
    base_model,
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),

    layers.BatchNormalization(),
    layers.Dense(3, activation='softmax')
])
model.compile(
    optimizer=optimizers.Adam(l Loading...
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
# Train the model
history = model.fit(
    train_ds,
    epochs=EPOCHS,  # Adjust the number of epochs as needed
    validation_data=val_ds,
    callbacks=[
        early_stopping,
        checkpoint_callback,
        reduce_lr
    ]
)
```

**Figure 4.4.1 Code snippet of VGG 16**

This includes the base VGG16 model, a layer called GlobalAveragePooling2D to reduce spatial dimensions, a layer called Dense with 128 units and ReLU activation for feature extraction, a layer called Dropout with a dropout rate of 0.5 to mitigate overfitting, a layer called Batch Normalization for normalization, and a final Dense layer with a softmax activation function for multi-class classification into three categories. Using accuracy as the evaluation metric, sparse categorical cross-entropy loss, and a learning rate of 1e-4, the model is constructed using the Adam optimizer. The fit approach is used to provide training on the using the training dataset train_ds, verifying on the validation dataset val_ds, and indicating the number of epochs as EPOCHS. To track and enhance training performance, callbacks are used, such as early stopping, checkpointing, and lowering learning rate.


vgg16-lung-cancer (2).ipynb
File Edit View Insert Runtime Tools Help Last edited on March 11
Comment
Share
+ Code + Text
Connect
Colab AI

```
28/28 [==============================] - 3s 91ms/step - loss: 0.1527 - accuracy: 0.9565 - val_loss: 0.3616 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 21/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1388 - accuracy: 0.9656 - val_loss: 0.3421 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 22/30
28/28 [==============================] - 2s 89ms/step - loss: 0.1511 - accuracy: 0.9576 - val_loss: 0.3424 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 23/30
28/28 [==============================] - 3s 90ms/step - loss: 0.1332 - accuracy: 0.9599 - val_loss: 0.3394 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 24/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1304 - accuracy: 0.9645 - val_loss: 0.3361 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 25/30
28/28 [==============================] - 3s 90ms/step - loss: 0.1401 - accuracy: 0.9530 - val_loss: 0.3374 - val_accuracy: 0.9062 - lr: 2.0000e-04
Epoch 26/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1351 - accuracy: 0.9611 - val_loss: 0.3348 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 27/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1257 - accuracy: 0.9668 - val_loss: 0.3235 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 28/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1262 - accuracy: 0.9656 - val_loss: 0.3168 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 29/30
28/28 [==============================] - 3s 91ms/step - loss: 0.1186 - accuracy: 0.9668 - val_loss: 0.3068 - val_accuracy: 0.8958 - lr: 2.0000e-04
Epoch 30/30
28/28 [==============================] - 3s 96ms/step - loss: 0.1076 - accuracy: 0.9714 - val_loss: 0.3072 - val_accuracy: 0.9167 - lr: 2.0000e-04
Training accuracy: 0.9713631272315979
Validation accuracy: 0.9166666865348816
```

**Figure 4.4.2 Training of VGG 16**

We obtained training accuracy of 0.97 and validation accuracy of 0.91 by training this model. These results indicate that the model performs well on training data and performs slightly worse—yet still well—on validation data.

# 5. MODEL EVALUATION & RESULTS

In this chapter, we are going to discuss about evaluation of the models and perform classifications between them.

## 5.1 Evaluation based upon Inception-v3

In a deep learning training process, the image depicts a line graph with two lines indicating 'Train Accuracy' and 'Validation Accuracy' that extend multiple epochs. The "Epochs" x-axis, which goes from 0 to 30, indicates the number of iterations, or epochs, they are for the duration of training on the entire data set. The model's accuracy is represented by the "Accuracy" y-axis, which has a range of 0.65 to 1.00.The training accuracy is represented by the blue line, which begins at a lower value and rapidly increases to a high level above 0.95 before fluctuating slightly but mostly maintaining a high level for the remaining epochs.

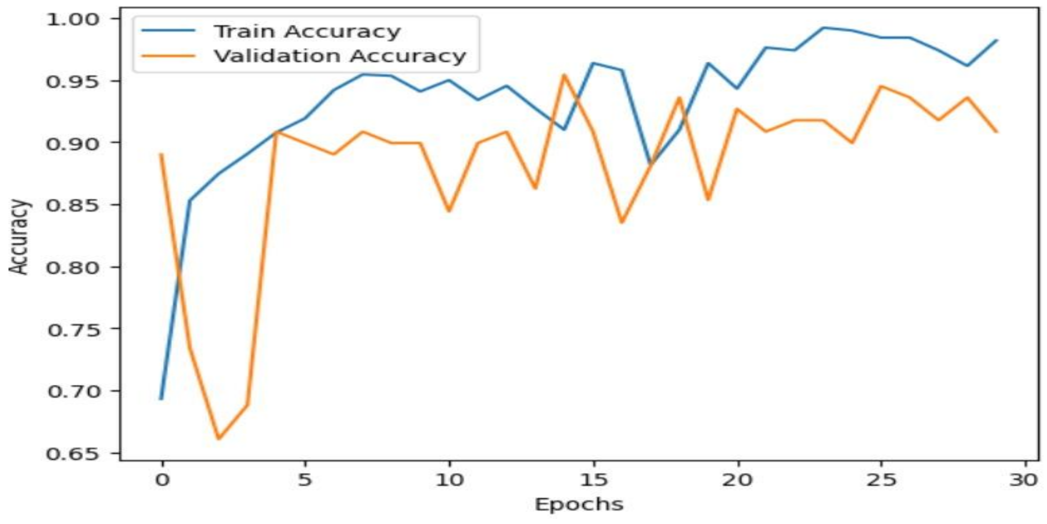


**Figure 5.1.1 Inception V3 Accuracy**

The validation accuracy is represented by the orange line, which likewise begins at a lower value, rises more gradually than the training accuracy, and then varies significantly, usually maintaining below the training accuracy. Peaks and troughs in the validation accuracy show that the model's compared to the training set, performance is less consistent on the validation set.

Given that the training accuracy is constantly higher than the validation accuracy, the graph suggests that the model is learning and getting better over time using the training data, but it also raises the possibility of some overfitting. When a model becomes overfit, it learns the training data, such as noise and details that don't transfer well to fresh, untainted data.

## 5.2 Evaluation based upon Resnet152

The "Training accuracy across Epochs for Each Fold" graph illustrates how a computer program's accuracy varies as it learns throughout a sequence of training epochs, or iterations. The graph shows the learning process for five different data subsets, or "folds" in cross-validation terminology. Cross-validation is a common approach used to evaluate the performance of a machine learning model on unobserved data.

The number of epochs is represented by the x-axis, which goes from 0 to minimal over 20. Accuracy is represented by the y-axis, which has values between 0.75 and 0.95.

Each of the graph's five lines corresponds to one of the five folds:

The first fold is represented by the blue line; the second by the orange line; the third by the green line; the fourth by the red line; and the fifth by the purple line.

Each line shows a different pattern of accuracy as the epochs increase. The blue line shows a sharp increase in accuracy early on, followed by a more gradual improvement. The other lines (orange,green, red, and purple) show various fluctuations in accuracy but generally hover around the 0.90 to 0.95 range after an initial increase.

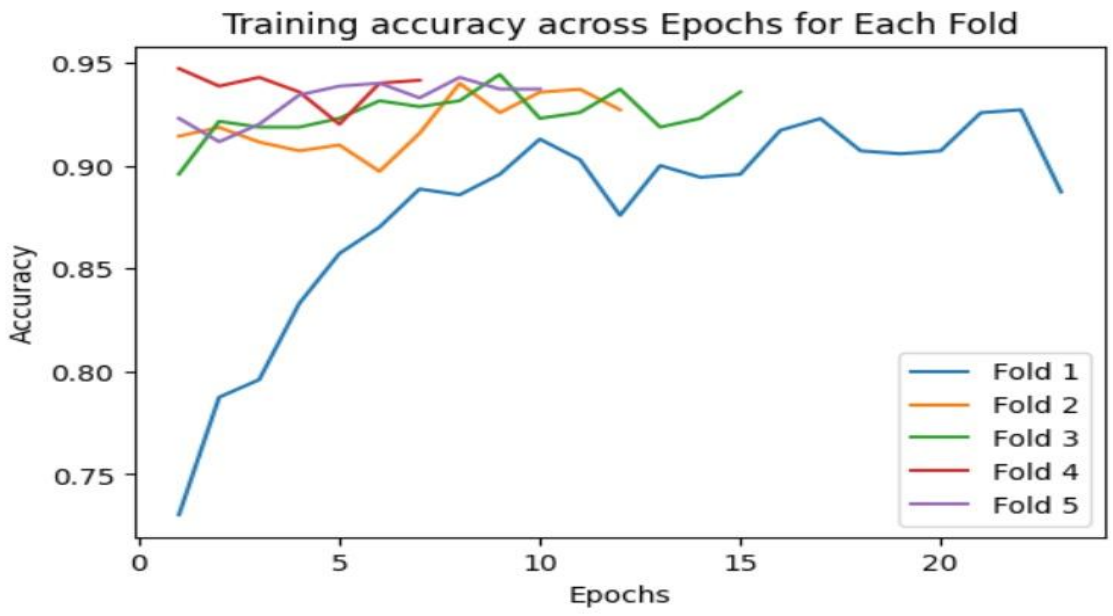


**Figure 5.2.1 Training Accuracy across Epochs for each fold**

Each of the graph's five lines corresponds to one of the five folds:

The first fold is represented by the blue line; the second by the orange line; the third by the green line; the fourth by the red line; and the fifth by the purple line.

Each line shows a different pattern of accuracy as the epochs increase. The blue line shows a sharp increase in accuracy early on, followed by a more gradual improvement. The other lines (orange,green, red, and purple) show various fluctuations in accuracy but generally hover around the 0.90 to 0.95 range after an initial increase.

The image is called "Training losses across Epochs for Each Fold," and it is a line graph. It displays, over several training epochs, the training loss of various folds of a machine learning model.The number of epochs is represented by the x-axis, which goes from 0 to little over 20. The loss is shown on the y-axis, which runs from 0 to 0.7 is a common loss scale, where lower values indicate better performance (i.e., less errors being made by the model).

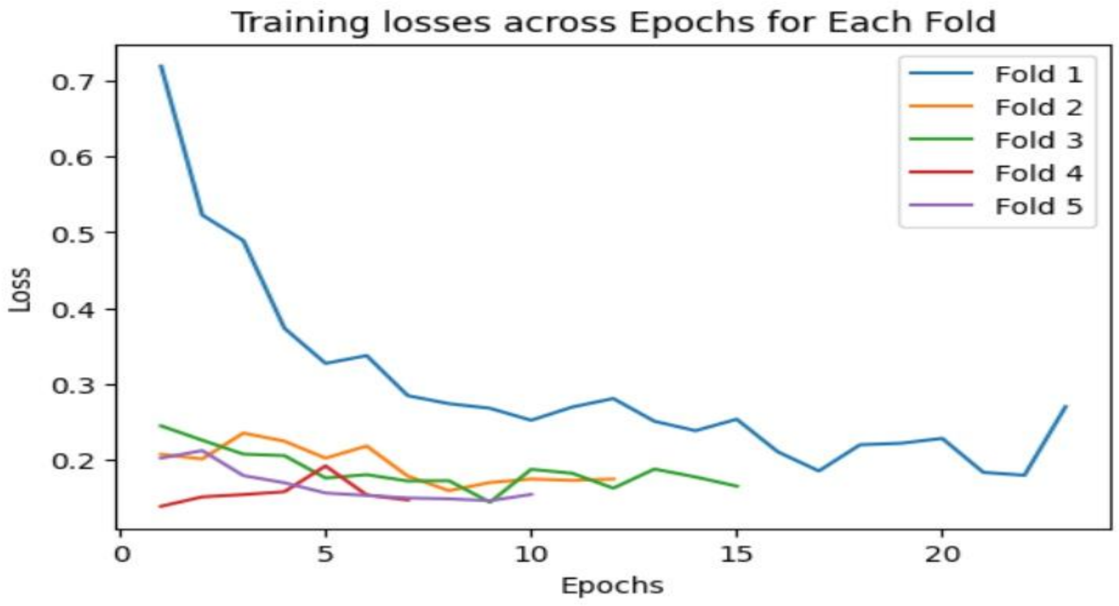


**Figure 5.2.2 Training Loss across Epochs for each fold**

As the number of epochs increases, the loss decreases on the graph, as it is expected as the model gains knowledge from the training set. The loss typically drops quite quickly at first as the model begins to identify patterns in the data. The pace of loss reduction usually slows down as training goes on showing that the model is beginning to approach a solution.

In k-fold cross-validation, where the dataset is divided into distinct subsets (folds) for training and validation, each fold displays a unique pattern of loss reduction. This aids in evaluating the generalizability and performance of the model.

According to the graph, Fold 1 begins with a loss that is significantly greater than the other folds', but by about epoch 5, it quickly converges to a loss level that is comparable. The other folds begin with a smaller loss values and also exhibit an overall downward trend, accompanied by occasional variations that might be brought about by the unique data in each fold or the random nature of the training process.

When everything is considered, the graph is helpful in comparing the training process between folds, which is useful in understanding the stability and consistency of the model's learning across various data subsets. Additionally, it can assist in determining whether a specific fold is acting abnormally, which may point to problems with the data or the fold's training process.

## 5.3 Evaluation based upon VGG16

This graph shows two lines that track how well a computer program is learning over time. The blue line shows the program's performance on the material it's been learning from, while the red line shows how well it does on new, similar material it hasn't seen before.

Both lines start at the left side of the graph, which is the beginning of the learning process, and move to the right side, which is the end of the learning period we're looking at. The bottom of the graph is the starting point for performance, and the top is the best performance possible.

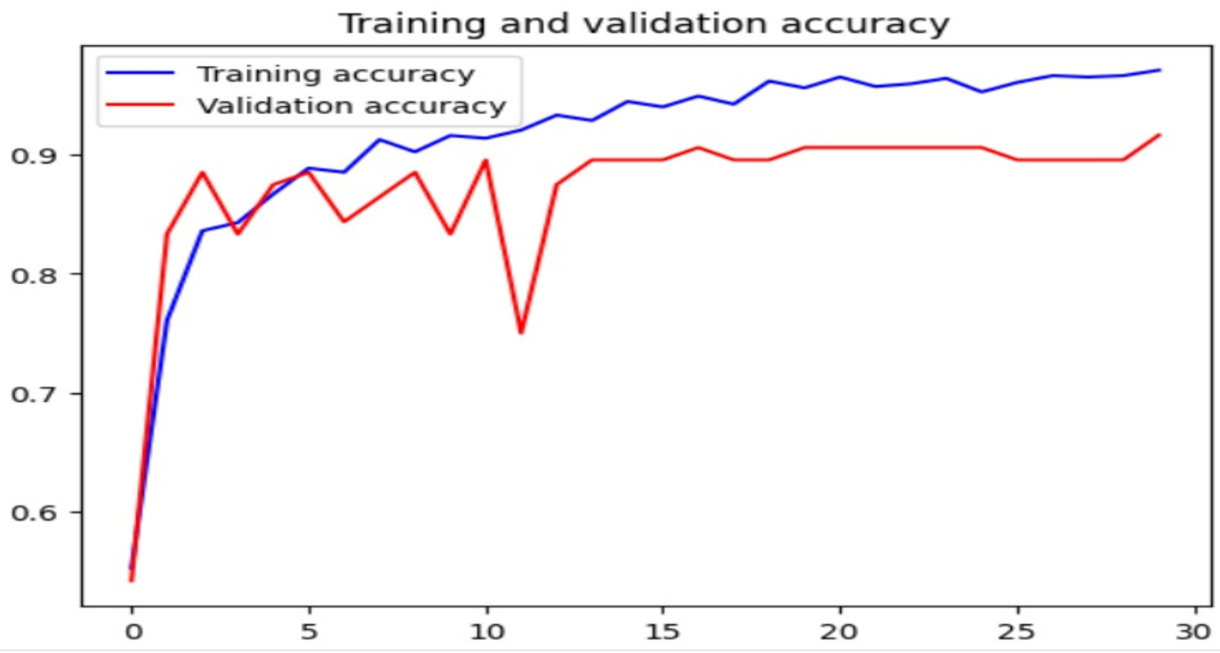


**Figure 5.3.1 Training and Validation Accuracies of VGG 16**

Two lines in this graph represent the learning capacity of a computer program over time. The red line indicates how well the program does on fresh, comparable content that it has never seen before, while the blue line displays the program's performance on the material it has been learning from.
Beginning at the left side of the graph, which represents the start of the learning process, both lines proceed to the right side, which represents the conclusion of the learning period under consideration. The lowest performance is represented at the bottom of the graph, and the highest performance is at the top.
The blue line rapidly increases and remains rather high, indicating that the program is utilizing the learning material effectively. The program's performance on new data is a little less stable, as indicated by the red line, which likewise rises but experiences some ups and downs. While the software is improving over time overall, there may be potential for improvement in terms of its ability to handle novel situations, as indicated by the ups and downs on the red line.

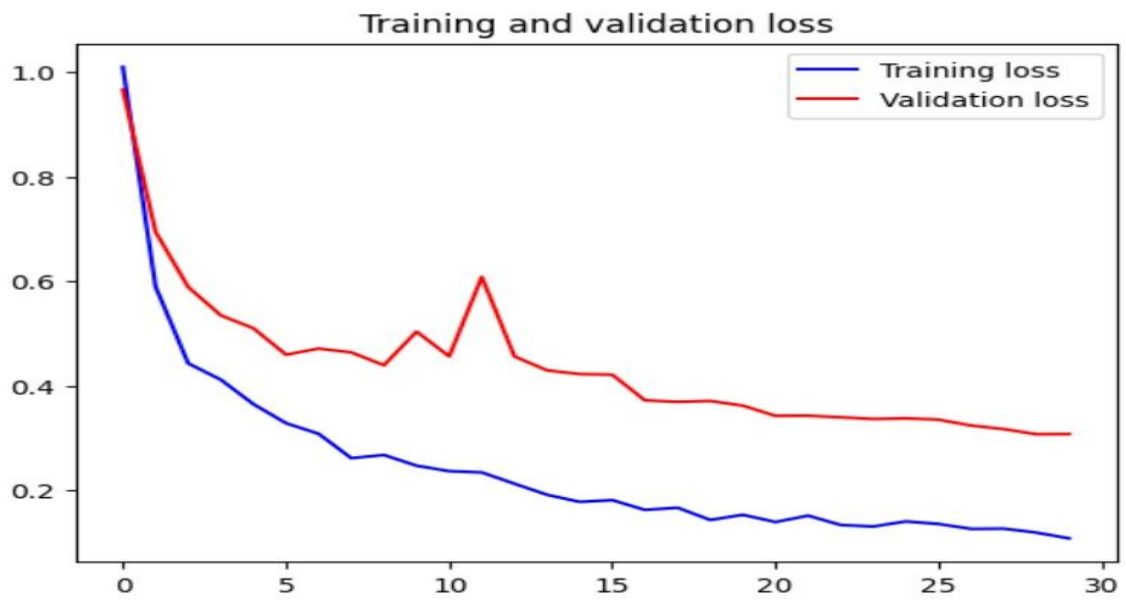


**Figure 5.3.2 Training and Validation Loss of VGG 16**

The presented graph, called "Training and validation loss," is a line chart with two lines that, over the course of several epochs (iterations) of training a machine learning model, reflect the loss values.

The graph's x-axis, which is labeled but not numbered, seems to show the number of epochs, extending from 0 to just under 30. The loss amount is represented by values labeled on the y-axis that range from 0 to 1.0. The graph consists of two lines:

The "Training loss," represented by the blue line, generally exhibits a declining trend as the number of epochs rises. This shows that as time goes on, the model learns from the training set and performs better in terms of loss. The "Validation loss," represented by the red line, likewise exhibits a declining trend but greater unpredictability. The model's performance on the validation set appears to be less consistent, as evidenced by the discernible spikes in the validation loss, especially around the 5th and 10th epochs.

Overall, learning is indicated by a decrease in both validation loss and training with time. To improve generalization, additional hyperparameter tuning of the model may be necessary, or variations in the validation loss may indicate overfitting or underfitting. The graph simply shows the relative changes in loss over time; it does not provide information on the model's absolute performance.

## ACCURACY

Accuracy is one parameter used to evaluate classification methods. Our model's accuracy is expressed as a fraction of its predictions. Accuracy equals the number of correct predictions multiplied by the total number of guesses. Accuracy is simply stated using Confusion matrix words like True Positive, True Negative, False Positive, and False Negative. Our model's accuracy is expressed as a fraction of its predictions.

**PRECISION**

Precision in machine learning refers to how effectively a model recognizes positive examples among those it predicts as positive. It calculates the percentage of real positive predictions to all positive predictions to assess the model's ability to avoid false positives. High precision indicates that the model produces fewer false positives, making it appropriate for activities where such errors are costly or undesirable, such as medical diagnoses.

**RECALL**

The proportion of data samples that a machine learning model correctly identifies as belonging to a class of interest (the "positive class") out of the total samples for that class. Recall assesses the completeness of positive predictions.

**F1-SCORE**

It is a machine learning evaluation statistic that assesses model accuracy. It combines a model's precision and recall scores. The accuracy measure calculates how often a model produced a valid prediction over the full dataset.

## Model-wise Accuracies for each set:

| | Training Set | Testing Set | Validation Set |
|---|---|---|---|
| VGG16 | 0.97 | 0.92 | 0.91 |
| Resnet152 | 0.96 | 0.9 | 0.93 |
| InceptionV3 | 0.98 | 0.90 | 0.90 |

**Table-6.1: Model wise accuracies for each set**

The above table compares the performance of three different convolutional neural network architectures: The VGG16, Resnet152, and InceptionV3 on the Training, Testing, and Validation sets are compared in the above table. Since the numbers range from 0 to 1, accuracy is probably

the metric used to assess performance. On the Training Set, 0.92 on the Testing Set, and 0.91 on the Validation Set, VGG16's accuracy is measured. Resnet152 has the lowest accuracy on the Testing Set (0.90), the lowest accuracy on the Training Set (0.96), and the lowest accuracy on the Validation Set (0.93) .With 0.98 accuracy on the Training Set, InceptionV3 exceeds Resnet152 with 0.90 accuracy on the Testing and Validation Sets. According to these results, VGG16 marginally exceeds InceptionV3 in terms of generalizing to previously unknown data, whereas InceptionV3 performs best on the Training Set.

**Classification report of InceptionV3:**

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Normal Case | 0.87 | 0.91 | 0.93 | 71 |
| Malignant Case | 0.94 | 0.90 | 0.92 | 56 |
| Bengin Case | 0.87 | 0.88 | 0.91 | 42 |

**Table- 6.2: Classification report of Inception V3**

The above table is evaluated based on four metrics: precision, recall, F1-score, and support. The precision, recall, F1-Score, and support value for the Normal Case are 0.87, 0.91, 0.93, and 71, respectively, showing the number of true cases in the dataset. The model obtains an accuracy of

0.94, a slightly lower recall of 0.90, an F1-Score of 0.92, and the Support metric indicates that there are 56 cases of the Malignant Case in the dataset. The F1-Score for the Benign Case is 0.91, its precision and recall are 0.87 and 0.88, respectively, and its support is 42. Overall, the table shows that the classification model performs well in all three categories, with the Malignant Case showing very high precision and all categories showing constant F1-Scores above 0.90, indicating an accurate balance between recall and precision in the model's predictions.

**Classification report of Resnet152:**

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Normal Case | 0.83 | 0.89 | 0.86 | 49 |
| Malignant Case | 0.89 | 0.96 | 0.91 | 73 |
| Bengin Case | 0.83 | 0.84 | 0.90 | 78 |

**Table-6.3: Classification report of Resnet152**

The above table measures: Precision, Recall, F1-Score, and Support values are measured in the above table. We obtained a precision of 0.83 for the Normal Case and a high recall of 0.96 (i.e., it accurately identified 96% of all actual malignant cases) for the Malignant Case. The harmonic mean of the F1-Score is the Benign Case has a high F1Score of 0.90, indicating an effective balance between precision and recall for benign predictions. This metric balances both precision and recall. Last but not least, the Support column shows how many instances of each class there are in the dataset,with 49 instances for Normal Case, 73 for Malignant Case, and 78 for Benign Case. These values collectively suggest that the model has a robust performance across the three classes, with a particularly strong ability to identify malignant cases.

**Classification report of VGG16:**

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Normal Case | 0.83 | 0.89 | 0.86 | 49 |
| Malignant Case | 0.90 | 0.99 | 0.94 | 82 |
| Bengin Case | 0.80 | 0.85 | 0.90 | 89 |

**Table-6.4: Classification report of VGG16.**

The above table is categorized into three classes: Normal Case, Malignant Case, and Benign Case There are four measures for each class: F1-Score, Precision, Recall, and Support. The precision, recall, and F1-Score for the Normal Case are 0.83, 0.89, and 0.86, respectively, with 49 support points. With 82 occurrences, the Malignant Case has a greater precision of 0.90, a recall of 0.99, and an F1-Score of 0.94. The Benign Case, which has the maximum support count of 89 instances, displays a precision of 0.80, recall of 0.85, and an F1-Score of 0.90. According to these measures, the model performs somewhat worse on Normal and Benign Cases and is most successful at accurately and reliably identifying Malignant Cases. The support values indicate the number of instances for each class that were used to compute these metrics.

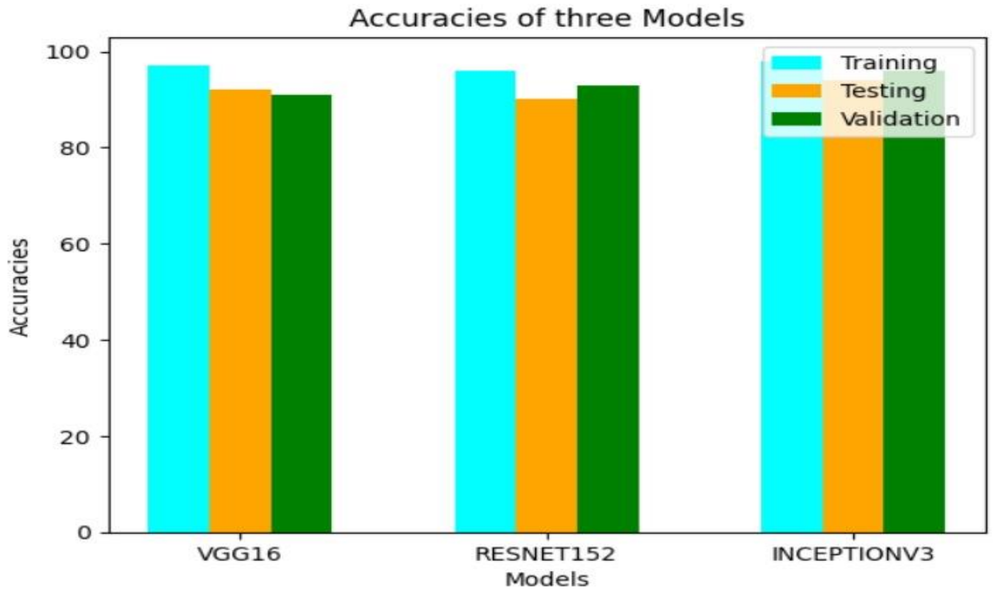


**Figure 6.1 Accuracies of three models**

The above image shows a bar chart titled "Accuracies of three Models." It compares the Models: VGG16, RESNET152, and INCEPTIONV3, across three different datasets: Training, Testing, and Validation.

Each model has three bars representing the accuracy percentages for each dataset type. The colors represent:

Cyan: Training accuracy

Orange: Testing accuracy

Green: Validation accuracy

From the chart, we can observe that all three models perform similarly on the training dataset, with very high accuracy, close to or at 100%. For the testing dataset, the accuracy is slightly lower for each model, which is common as models tend to perform better on previously seen data (training data) than on new acquired data (testing data).The validation accuracy is also high for each model, indicating that the models generalize well to unseen data.

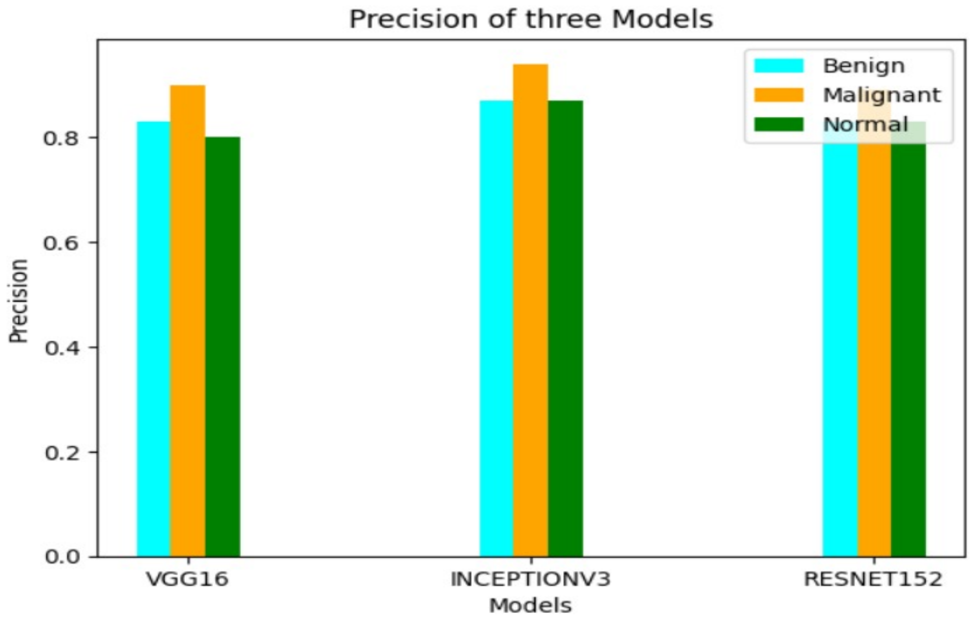


**Figure 6.2 Precision of three models**

The image shows a bar chart titled "Precision of three Models." It compares the precision of three different machine learning models: VGG16, INCEPTIONV3, and RESNET152, for classifying three different categories: Benign, Malignant, and Normal.

Each model has three bars representing the precision values for each category.

The colors represent: - Cyan: Benign precision

- Orange: Malignant precision

- Green: Normal precision

Precision in this context refers to the percentage of genuine positive predictions to the total number of positive predictions (true positives plus false positives). It is a measure of a model's accuracy in classifying a class correctly. From the chart, we can observe that all three models have relatively high precision across all categories, with precision values ranging from around 0.6 to just below 0.9. The INCEPTIONV3 model appears to have slightly higher precision for the Benign and

Normal categories compared to the other two models, while the RESNET152 model seems to have a marginally higher precision for the Malignant category.

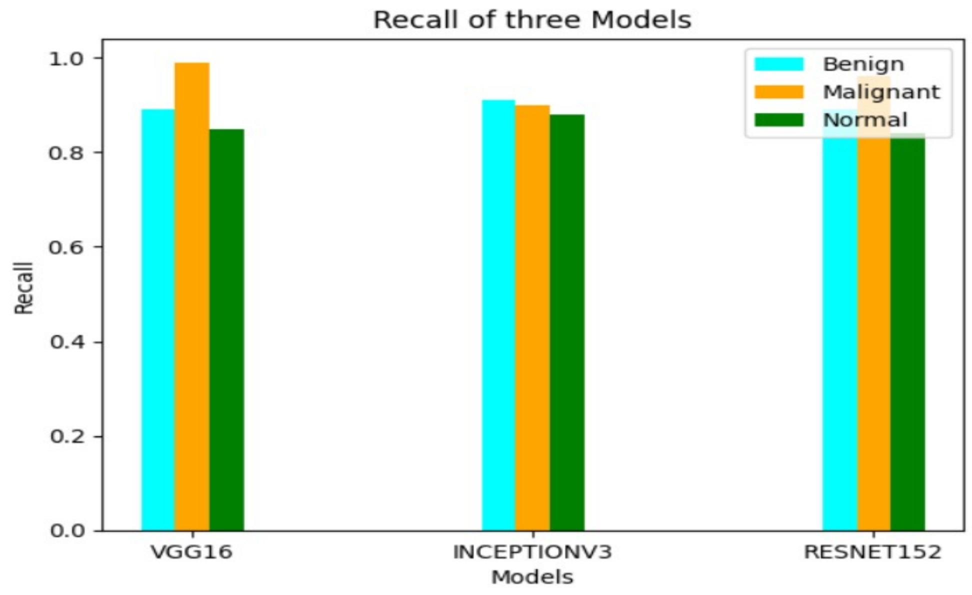


**Figure 6.3 Recall of three models**

The image shows a bar chart titled "Recall of three Models." It compares the recall of three different machine learning models: VGG16, INCEPTIONV3, and RESNET152, for classifying three different categories: Benign, Malignant, and Normal.

Each model has three bars representing the recall values for each category.

The colors represent:

- Cyan: Benign recall

- Orange: Malignant recall

- Green: Normal recall. Recall, which is a synonym for sensitivity, quantifies the percentage of real positives that the model properly identifies. It is computed by dividing the total number of actual positives true positives + false negatives by the number of genuine positive predictions. From the chart, we can observe that all three models have high recall values for all categories, with values close to or at 1.0, indicating that the models are highly capable of identifying all actual positive cases for each category.

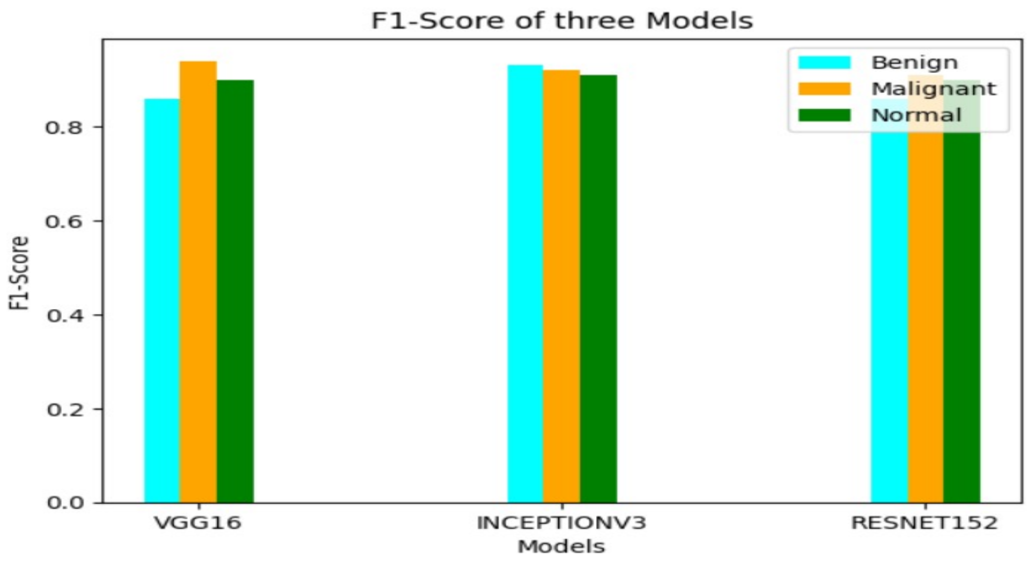


**Figure 6.4 F1-Score of three models**

The image shows a bar chart titled "F1-Score of three Models". There are three groups of bars, each representing a different model: VGG16, INCEPTIONV3, and RESNET152. Within each group, there are three bars colored differently to represent the F1-Score for different classifications: Benign (orange), Malignant (green), and Normal (blue).

The F1-Score is a measure used in statistics to gauge the accuracy of a test, considering both the precision and the recall to compute the score. It is particularly useful for uneven class distribution, as it takes into account both false positives and false negatives.

From the chart, we can observe the following:

- All three models perform similarly on the Malignant and Normal classifications, with the F1-Scores being very close to each other.

- For the Benign classification, the VGG16 model appears to have a slightly lower F1-Score compared to the other two models.

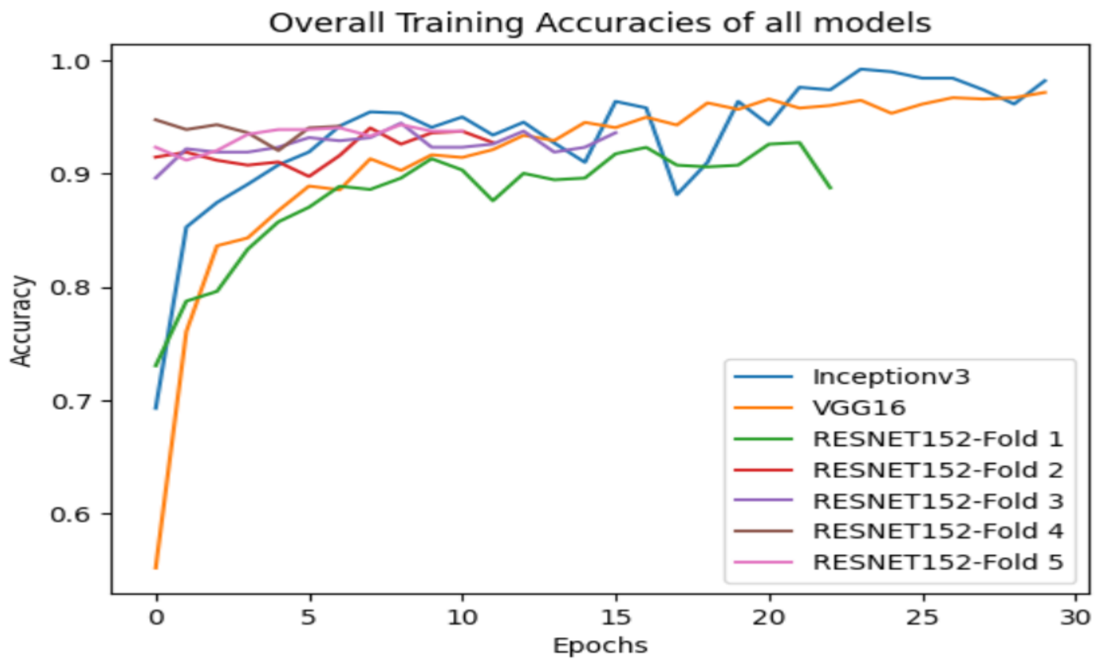


**Figure 6.5 Overall Accuracies of all models**

The image above is a line graph named "Overall Training Accuracies of All Models". The x-axis reflects the number of epochs (0-30), while the y-axis denotes accuracy (0.6-1.0). The graph depicts the accuracy of different models as they are trained across multiple epochs.

The graph contains six lines, each showing the accuracy of a distinct model or fold over time.

- Inception V3 (blue line)
- VGG16 (Orange line)
-RESNET152, Fold 1 (green line)
-RESNET152-Fold 2 (red line)

-RESNET152-Fold 3 (purple line)

-RESNET152-Fold 4 (brown line)

-RESNET152-Fold 5 (pink line)

Each line depicts the evolution of model accuracy as training progresses over epochs.

Each line shows the progression of model accuracy as training continues through epochs.-

The lines for the RESNET152 model are distinguished by "folds," which most likely suggest that the data was divided into multiple subsets (folds) for cross-validation purposes, a typical approach to ensure that the model's performance is consistent across different parts of the data.The graph shows the following:

- All models begin with varying accuracies at epoch 0, but quickly improve over the next few epochs.- The Inceptionv3 model (blue line) exhibits a rapid gain in accuracy and maintains it throughout the training period.- The VGG16 model (orange line) likewise improves, but appears to plateau earlier and with lesser accuracy than Inceptionv3.

The RESNET152 models' accuracy varies among folds, with certain folds outperforming others. This variation may be related to changes in data subsets used for each fold. By the end of the 30 epochs, all models' accuracies appear to have stabilized, with Inceptionv3 remaining the most accurate.

Overall, the graph compares the learning performance of various models or the same model with different data subsets over time, which is critical for evaluating how well the models learn from the data and how quickly they converge to a stable accuracy.

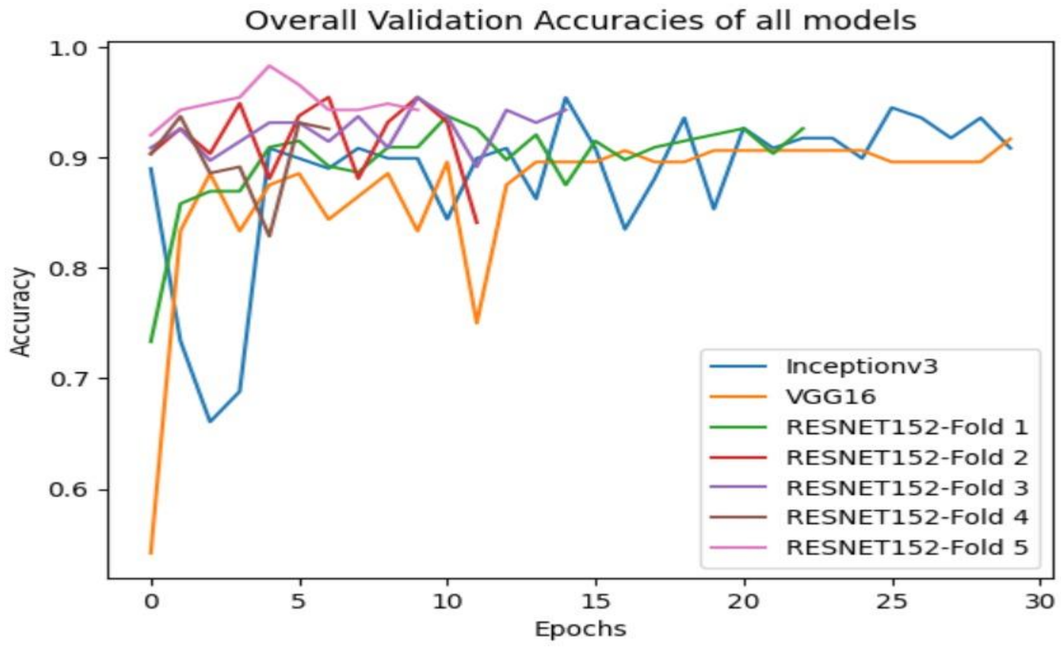


**Figure 6.6 Overall Validation Accuracies of all models**

The above image is a line graph titled "Overall Validation Accuracies of all models." It plots the accuracy of different deep learning models over a number of training epochs. The x-axis represents the number of epochs, ranging from 0 to 30. The y-axis represents accuracy, ranging from 0.6 to 1.0, which is a typical scale for accuracy where 1.0 represents 100% accuracy.

The graph shows fluctuations in accuracy for each model as training progresses. This is common in training deep learning models, as the model weights are adjusted to minimize the loss function, which can lead to temporary decreases in validation accuracy.

From the graph, we can observe that all models start with varying degrees of accuracy at epoch 0, and as training progresses, their accuracies generally improve. However, there are some dips and peaks, which could be due to overfitting, learning rate adjustments, or other factors affecting the training process.

The InceptionV3 model appears to perform consistently well, maintaining a high accuracy throughout the training process. The VGG16 model shows more volatility but also reaches high accuracy levels. The RESNET152 models, represented by different folds, show varying Performance, with some folds performing better than others. Overall, the graph is useful for

Comparing the performance of different models or different cross-validation folds of the same model over time, which can help in selecting the best model for a given task based on its validation accuracy.

# CONCLUSION

In conclusion, the development of predictive models for lung cancer holds immense promise in the realm of healthcare. We can greatly increase our capacity to identify those who are at a high risk of getting this fatal disease by utilizing deep learning. A powerful approach for predicting lung cancer is the use of deep learning models, such as Inception V3, ResNet-152, and VGG-16. These models, which were first created for image recognition applications, have demonstrated amazing promise in the accurate and dependable analysis of medical imaging data.

Deep learning models like Inception V3, ResNet-152, and VGG-16 excel in automatically learning and extracting meaningful patterns from lung images, which traditional methods might miss. They can discern subtle features indicative of lung cancer presence, even in complex and noisy images. After training these three models on the lung cancer dataset, the Inception v3 model has shown to work better compared to Resnet152 and VGG16 by giving a highest accuracy of 98%. Resnet has given an accuracy of 96% and VGG16 has given an accuracy of 97%.
One of the significant advantages observed from these models is their ability to handle large amounts of data efficiently. These models may learn from a wide range of instances by training on various datasets that contain thousands of lung images. This improves their capacity to generalize and make precise predictions on fresh, unknown data. The implementation of such predictive models in clinical practice can revolutionize the landscape of lung cancer prevention and management. Early detection facilitated by these models can lead to timely interventions, thereby improving patient outcomes and potentially reducing the economic burden associated with advanced-stage cancer treatment.

Moving forward, continued research and innovation in predictive analytics will be crucial for refining and optimizing lung cancer prediction models. Collaboration between healthcare professionals, researchers, and technology experts will be essential to ensure the translation of these advancements into real-world clinical practice.

# FUTURE ENHANCEMENTS

1.**Multi-modal Data Fusion**: Combining clinical data (the population, smoking history, genetic markers) with information from several imaging modalities (such as CT, MRI, and PET scans) can give a more complete picture of the illness. Predictive accuracy can be enhanced and the complexity of lung cancer better captured by deep learning models that can handle multi-modal input.

2. **Temporal Analysis**: Incorporating longitudinal data over time can enhance predictive models by capturing disease progression and treatment responses. Recurrent neural networks (RNNs) or attention mechanisms can be employed to analyse sequential imaging data and patient records, allowing for dynamic predictions and personalized treatment strategies.

3. **Uncertainty Estimation**: When presented with ambiguous data or forecasts, deep learning models frequently exhibit a lack of robustness. By offering uncertainty estimates, methods like Monte Carlo dropout or Bayesian neural networks help physicians make better decisions and enhance the interpretability of models.

4. **Interpretability**: Improving techniques to clarify deep learning models' predictions might boost acceptability and confidence in healthcare settings. Saliency maps, model-agnostic interpretation techniques, and attention mechanisms are a few examples of techniques that can be used to identify regions of interest in images and reveal the characteristics that influence predictions.

5. **Integration with Electronic Health Records (EHR):** Incorporating information from EHR, such as laboratory test results, medication history, and comorbidities, can provide valuable context for predicting lung cancer risk and prognosis. Deep learning models capable of processing heterogeneous EHR data alongside medical imaging could offer more holistic patient assessments.

6. **Collaborative Research and Data Sharing**: Collaborative Research and Data Sharing: With access to larger and more varied datasets, research institutions working together and data sharing efforts can support the creation of strong deep learning models. Federated learning techniques can allow for model training over several sites while maintaining the security and privacy of data.

# REFERENCES

[1] Zinovev et al. (2011) Probabilistic lung nodule classification with belief decision trees. Engineering in Medicine and Biology Society, EMBC, Annual International Conference of the IEEE, pp. 4493–4498. https://doi.org/10.1109/IEMBS.2011.6091114 .

[2] Roy et al. (2015) A Classification of lung image and nodule detection using fuzzy inference system. IEEE International Conference on Computing, Communication & Automation, pp. 1204– 1207.https://doi.org/10.1109/CCAA.2015.7148560.

[3] N.V Ramana Murty and Prof. M.S. Prasad Babu (2017), A Critical Study of Classification Algorithms for Lung Cancer Disease Detection and Diagnosis. International Journal of Computational Intelligence Research, vol. 13, no. 5, pp. 1041-1048. https://api.semanti-cscholar.org/CorpusID:38800795.

[4] Atsushi et al. (2017) Automated Classification of Lung Cancer Types from Cytological Images Using Deep Convolutional Neural Networks. BioMed Research International. pp. 1-6. https://doi.org/10.1155/2017/4067832.

[5] M. Saric et al. (2019) CNN based Method for Lung Cancer Detection in Whole Slide Histopathology Images, 4th International Conference on Smart and Sustainable Technologies, pp. 1-4. https://doi.org/10.23919/SpliTech.2019.8783041.

[6] Fang Zhou et al (2019) Evaluate the Malignancy of Pulmonary Nodules Using the 3-D Deep Leaky Noisy-or Network. IEEE Trans. Neural Netw. Learn. Syst.30, pp. 3484–3494. https://doi.org/10.1109/TNNLS.2019.2892409.

[7] Hasan et al. (2019) Lung cancer detection and classification based on image processing and statistical learning. J Emerg Trends Eng Appl Sci 11.https://hdl.handle.-net/10520/ejcsl_jeteasv11-n6-a4.

[8] Radhanath patra et al. (2020) Prediction of lung cancer using machine learning classifier. International conference on computing science, communication and security, computer science and communication security, pp 132–142.https://do i.org/10.1007/978-981-1566486_11.

[9] Ramakrishnan et al. (2022) Automated Lung Cancer Nodule Detection, Santa Clara University, pp 1–37. https://doi.org/10.3390/s19173722.

[10] Ruchita Tekade and K. Rajeswari(2018) Lung Cancer Detection and Classification Using Deep Learning, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1- 5.https://doi.org/10.1109/ICCUBEA.2018.8697352 .

[11] Ahmad T, Shahnajparvin M (2020) Lung cancer detection using CT image based on 3D convolutional neural network, pp. 35–48.https://doi.org/10.4236/jcc.2020.83004.

[12] Schreuder et al. (2021). Combining pulmonary and cardiac computed tomography biomarkers for disease-specific risk modelling in lung cancer screening. European Respiratory Journal, 58(3). https://doi.org/10.1183/13993003.03386-2020.

[13] Agarwal et al.(2021) Lung cancer detection and classification based on alexnet CNN. In 2021 6th international conference on communication and electronics systems (ICCES), IEEE, pp. 13901397. https://doi.org/10.3390/s22124426.

[14]Saleh et al. (2021) Lung cancer medical images classification using hybrid CNN-SVM. International Journal of Advances in Intelligent Informatics, 7(2), 151-162. https://doi.org/10.26555/ijain.v7i2.317.

[15] Faruqui et al. (2021) LungNet: A hybrid deep-CNN model for lung cancer diagnosis using CT and wearable sensor-based medical IoT data. Computers in Biology and Medicine, 139. https://doi.org/10.1016/j.compbiomed.2021.104961.

[16] Kasinathan et al. (2019) Automated 3-D lung tumor detection and classification by an active contour model and CNN classifier. Expert Systems with Applications, 134, 112-119. https://doi.org/10.1016/j.eswa.2019.05.041.

[17] Naseer, I et al. (2022) Performance analysis of state-of-the-art cnn architectures for luna16. Sensors, 22(12). https://doi.org/10.3390/s22124426.

[18] Ausawalaithong et al. (2018) Automatic lung cancer prediction from chest X-ray images using the deep learning approach. In 2018 11th biomedical engineering international conference (BMEiCON) (pp. 1-5). IEEE. https://doi.org/10.48550/arXiv.1808.10858.

[19] Sajja et al. (2019) Lung Cancer Detection Based on CT Scan Images by Using Deep Transfer Learning. Traitement du Signal, 36(4), 339-344. https://doi.org/10.18280/ts.360406.

[20] Ali et al. (2022) An efficient U-Net framework for lung nodule detection using densely connected convolutions, J.Supercomput, vol.78, no-2,pp.1602–1623.https://doi.org/10.1007/s11227-021-03845-x.

# APPENDIX A -ABBREVIATIONS

- VGG – Visual Geometry Group
- CNN – Convolutional Neural Network
- ResNet – Residual Network
- CT Scan – Computed Tomography
- GUI – Graphical User Interface
- IQ-OTH/NCCD - Iraq-Oncology Teaching Hospital/ National Centre for Cancer disease
- ReLU – Rectified Linear Unit

# APPENDIX B – SOFTWARE INSTALLATION PROCEDURE

## DATASET:

The dataset is collected from Kaggle and nearly 1190 images of each class are trained and classified.

## BENIGN CASE:

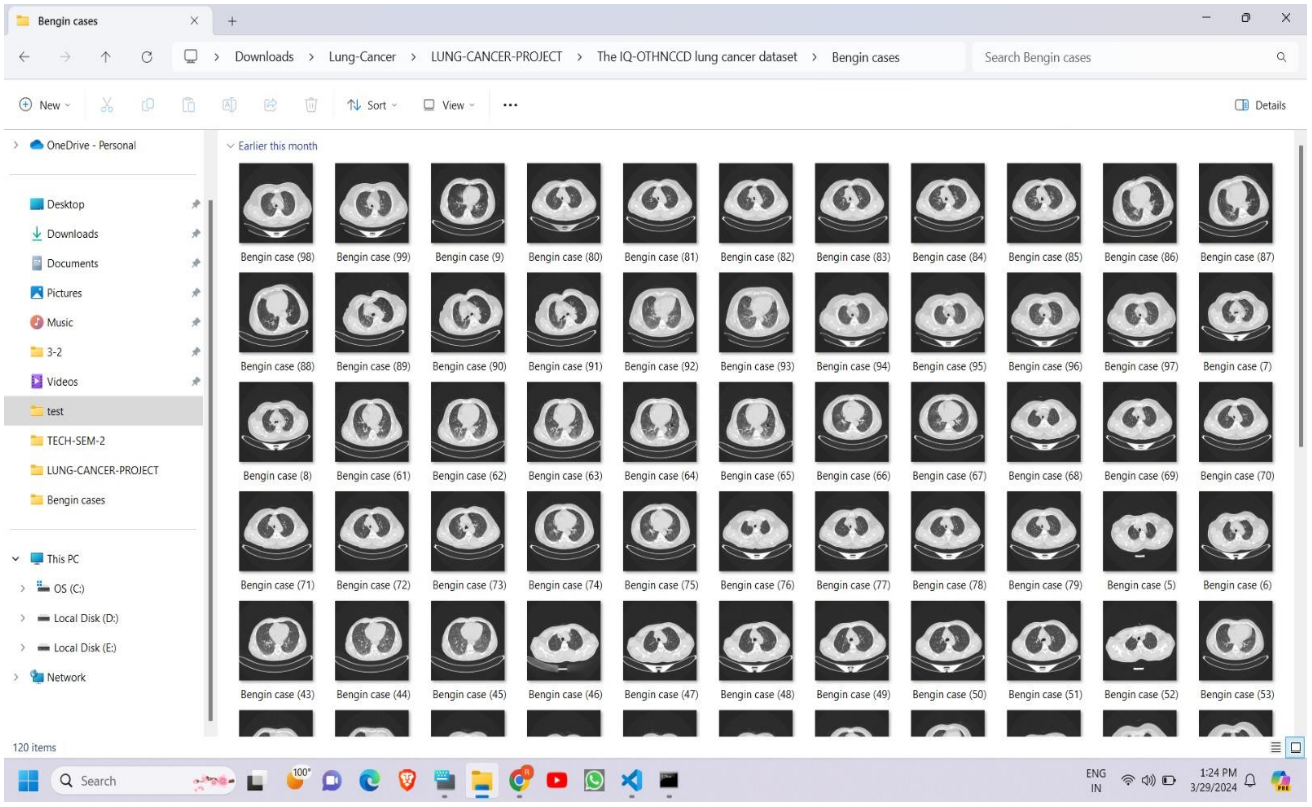


**Figure 7.1 Benign Case**

## MALIGNANT CASE:

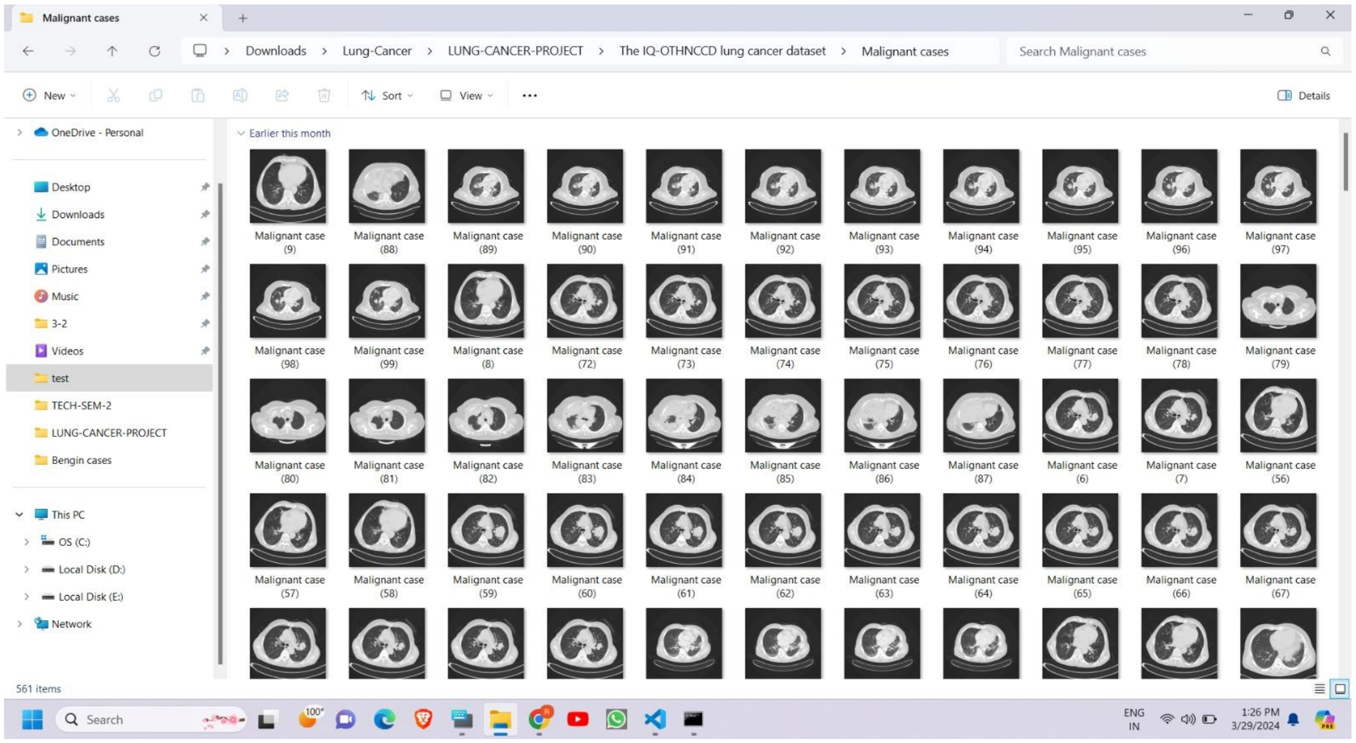


**Figure 7.2 Malignant case**

## NORMAL CASE:

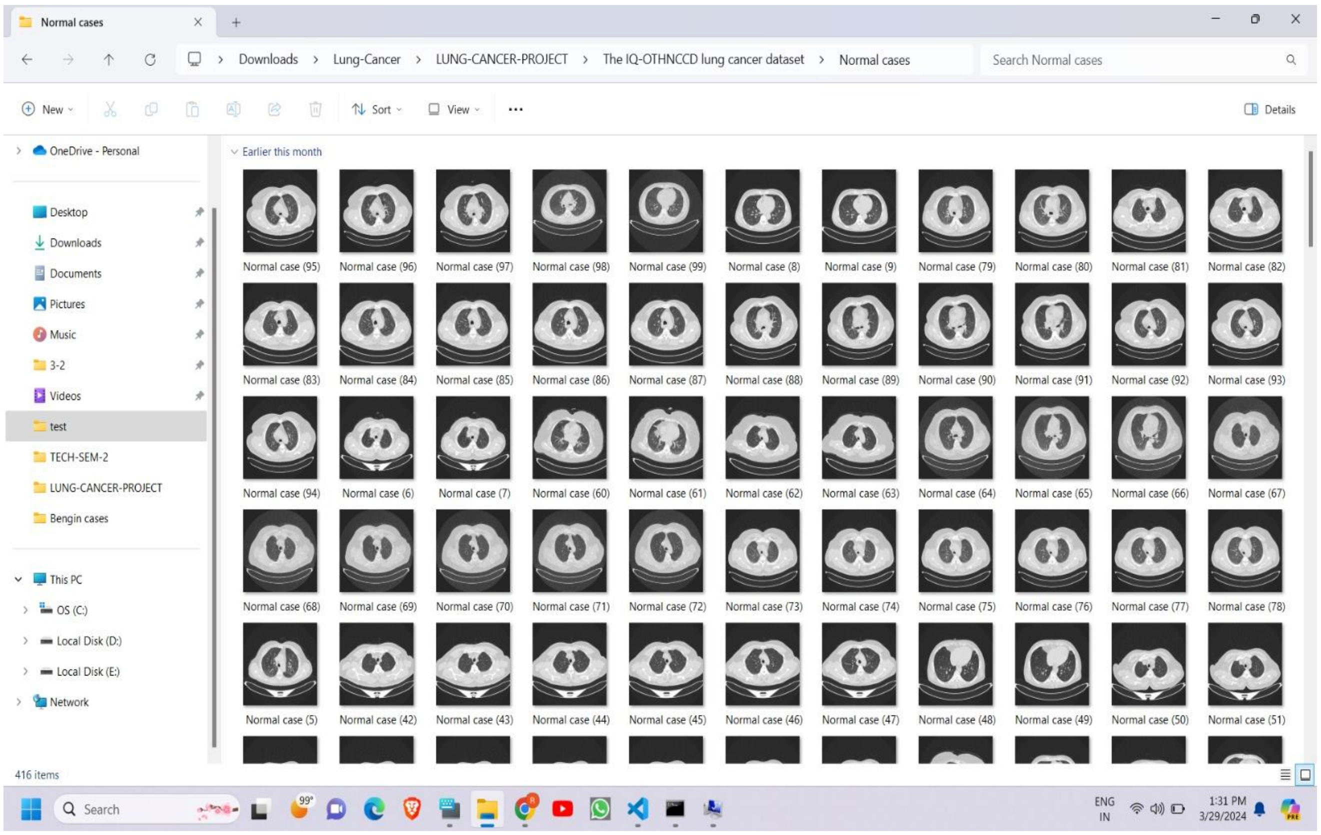


**Figure 7.3 Normal case**

# APPENDIX C – SOFTWARE USAGE PROCESS

**Create a New Notebook:**

From the Google colab, you can create a new notebook by clicking on the "New" button and selecting "Python" or any other available kernel depending on your project requirements.

This action opens a new notebook interface where you can write and execute Python code, create markdown cells for documentation, and add visualizations or other media.

**Write and Execute Code:**

Within the notebook interface, you can write Python code in individual cells and execute them by pressing Shift + Enter or clicking the "Run" button. Colab allows you to execute code interactively, viewing the output directly below each code cell. You can also edit and re-run cells as needed.

**Import Libraries and Data:**

To perform data analysis or machine learning tasks, you can import relevant Python libraries such as NumPy, Pandas, Matplotlib, and scikit-learn.

Additionally, you can import datasets or load data from external sources directly into your Colab notebook for analysis or modeling.

**Model Development and Evaluation:**

Using Colab Notebook, you can develop machine learning models and evaluate their performance using libraries like scikit-learn or TensorFlow.

You can train models, tune hyperparameters, and evaluate model performance metrics such as accuracy, precision, recall, and F1-score.

**Documentation and Reporting:**

Throughout the software usage process, you can document your workflow, analysis steps, and results using markdown cells within the Colab notebook.

You can add explanations, comments, and visualizations to create a comprehensive and interactive report documenting your project findings.

**Sharing and Collaboration:**

Once your analysis is complete, you can share your Colab notebook with colleagues or collaborators by exporting it to various formats such as HTML, PDF, or Markdown.

You can also share notebooks directly via email or by hosting them on platforms like GitHub, GitLab for collaborative editing and sharing.