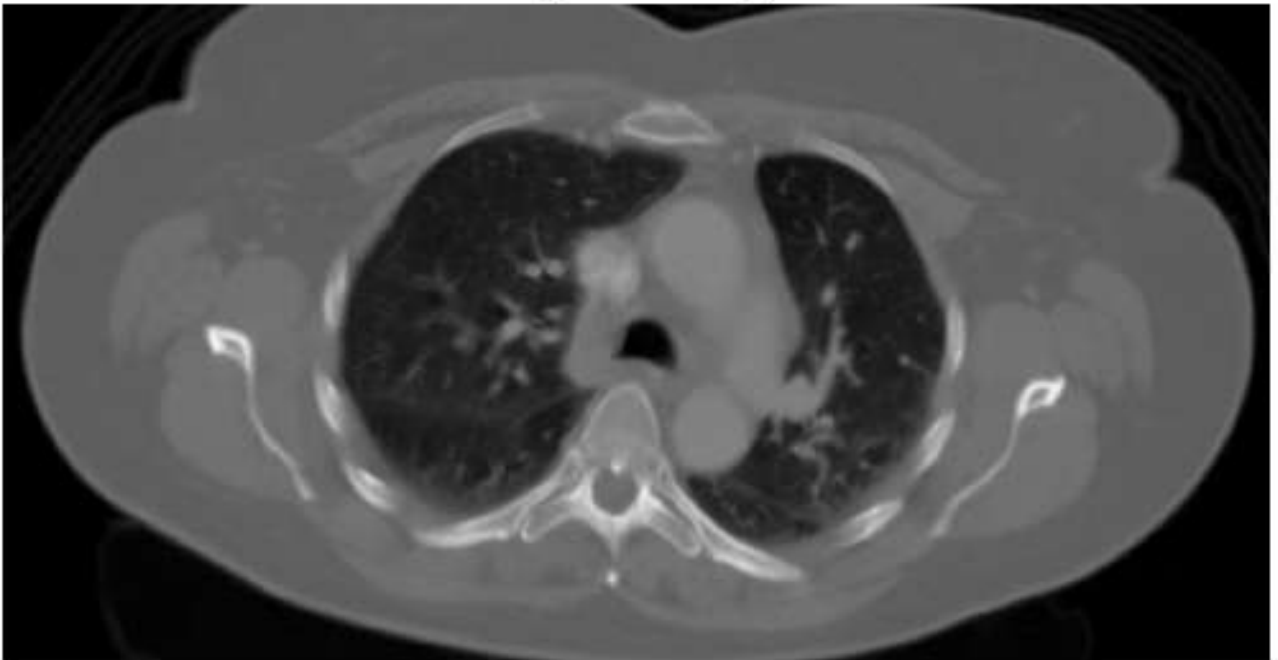


# Data Science Capstone

## Chest Cancer Detection Using Deep Learning on Chest CT-Scan Images

Original Image



**Lissan Liben**

**Professor Steve Chesney**

**Capstone**

## Chest Cancer Detection Using Deep Learning on Chest CT-Scan Images

This project focuses on building a deep learning model for the early detection of lung cancer using chest CT-Scan images. Lung cancer is one of the leading causes of cancer-related deaths worldwide, and early detection significantly increases survival rates. The goal of this project is to develop a Convolutional Neural Network (CNN) model capable of accurately classifying CT-Scan images into cancerous and non-cancerous categories. The dataset includes images labeled as either normal (non-cancerous) or representing different types of lung cancer, such as *adenocarcinoma*, *large cell carcinoma*, and *squamous cell carcinoma*.

### Project Scope:

- I. **Data Acquisition and Exploration:** The dataset is comprised of labeled chest CT-Scan images, divided into training, validation, and test sets. Initial exploration will focus on understanding image distributions and class labels, ensuring data quality and completeness.
- II. **Data Preprocessing:** Key steps include resizing, normalization, and augmentation of the images to standardize the input format and enhance the dataset for model training. A class imbalance strategy will be addressed if necessary.
- III. **Model Development:** A CNN model will be implemented to classify images into cancerous or non-cancerous categories. Transfer learning with pre-trained models may be employed to improve performance.
- IV. **Model Evaluation:** The model will be evaluated using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC, with a focus on minimizing false negatives.

## Scope of the Project

This project focuses on developing a deep learning model to detect lung cancer using chest CT-Scan images. The primary objective is to classify images into cancerous and non-cancerous categories by leveraging Convolutional Neural Networks (CNNs). The scope includes data acquisition, preprocessing, model development, followed by evaluation and optimization of the model to achieve high accuracy in distinguishing different types of lung cancer, such as adenocarcinoma, squamous cell carcinoma, and large cell carcinoma. Additionally, transfer learning will be explored to enhance the model's performance; by addressing challenges such as imbalanced class distributions and complex image variability, this project aims to build a robust, scalable solution for lung cancer detection.

## Project Importance

This project is important because lung cancer is the leading cause of cancer-related deaths globally, with low survival rates when detected at advanced stages. Early detection significantly improves outcomes, but manual analysis of CT-Scan images is prone to error and time-consuming. Developing an automated detection model has the potential to assist radiologists in early diagnosis, providing faster, more accurate results. This not only reduces the workload for healthcare professionals but also offers life-saving potential by detecting lung cancer early, when it is more treatable. The project's use of deep learning to classify complex medical images represents a significant advancement in the integration of artificial intelligence (AI) in healthcare, with the potential to benefit both clinicians and patients.

## Background of the Problem Being Analyzed

Lung cancer detection remains challenging due to the subtle and diverse appearances of tumors in CT-Scan images. Even experienced radiologists can miss early signs of cancer, leading to delayed treatment. According to Giger (2018), AI-based diagnostic tools can enhance detection by providing an automated second opinion, significantly reducing the potential for human error. Deep learning techniques, especially CNNs, have demonstrated success in detecting patterns in medical images that are difficult for humans to identify (Litjens et al., 2017). These tools are particularly valuable in screening tasks where analyzing large volumes of images quickly is essential for early diagnosis. This project builds upon this foundation by applying CNN models to distinguish between various lung cancer types and normal CT-Scans.

## Dataset Description

The dataset consists of chest CT-Scan images labeled according to cancer type *adenocarcinoma*, *large cell carcinoma*, *squamous cell carcinoma* and *normal cases*. The images are divided into training, validation, and test sets, providing a structure for model development and evaluation. Each image has been standardized in terms of size and resolution, making it ready for deep learning model input. The dataset reflects real-world challenges, such as imbalanced class distribution (more normal images than cancerous ones) and the complexity of detecting small lesions within the lungs. Augmentation techniques will be applied to enhance dataset diversity and improve the generalization capabilities of the model. This dataset forms the foundation of the project, offering a robust set of images that represent common diagnostic challenges faced by radiologists.



## Table of Contents

Project Scope .....	7
Problem Description.....	7
Project Importance .....	9
Background .....	10
Data Set Description.....	11
Data Analytics Tools.....	12
Project Milestones.....	13
Completion History .....	<b>Error! Bookmark not defined.</b>
Lessons Learned .....	<b>Error! Bookmark not defined.</b>
Data Profiling and Preparation.....	24
Data Summary .....	24
Data Definition/Data Profile .....	31
Data Preparation.....	32
Data Visualizations.....	40
Descriptive Statistics.....	46
Data Visualization Definitions .....	49
Data Visualization 1 .....	59
Data Visualization 2.....	61
Data Modeling.....	63
Data Modeling Definitions .....	70
Data Model 1 .....	75
Data Model 2.....	77
Review of Data Models .....	79
Final Results.....	<b>Error! Bookmark not defined.</b>
Findings.....	83
Review of Success or Completion.....	94
Recommendations for Future Analysis.....	95
References.....	99

# Project Scope

## Problem Description

Lung cancer is the leading cause of cancer-related deaths globally, accounting for millions of deaths each year. Early detection significantly increases the chances of successful treatment, but diagnosing lung cancer through traditional means, such as chest X-rays or CT-Scans, can be challenging, time-consuming, and prone to human error. Radiologists are often required to examine hundreds of images to detect signs of cancer, leading to a high potential for misdiagnosis, especially in the early stages when symptoms are less obvious.

This project aims to address the problem of automating lung cancer detection using advanced machine learning techniques, specifically deep learning models. By leveraging a dataset of labeled chest CT-Scan images, the project seeks to develop a reliable Convolutional Neural Network (CNN) that can classify CT-Scans as either cancerous or non-cancerous. This will not only speed up the diagnosis process but also potentially reduce human error, providing a more consistent and accurate approach to identifying lung cancer at its earliest stages.

The problem lies in the ability to differentiate between various types of lung cancer adenocarcinoma, large cell carcinoma, squamous cell carcinoma and normal lung scans. The variability in tumor appearance, size, and location across different patients makes accurate classification challenging. This project addresses the need for an automated tool that can assist healthcare professionals by improving the speed and accuracy of cancer detection, ultimately leading to earlier interventions and better patient outcomes.

## Importance of the Problem

Lung cancer remains the leading cause of cancer-related deaths worldwide, with millions of new cases diagnosed each year. The survival rate for lung cancer is significantly higher when detected early, but unfortunately, most cases are diagnosed at advanced stages when treatment options are limited, and the prognosis is poor. Early detection is critical, as it can dramatically increase the likelihood of successful treatment and extend patient survival.

However, identifying lung cancer in its early stages is particularly challenging. Radiologists must meticulously analyze chest CT-Scan images to detect subtle signs of cancer, which can vary widely depending on the type of cancer and its location in the lungs. This process is time-consuming and susceptible to human error due to the sheer volume of images that must be reviewed. Misdiagnoses or delays in diagnosis can lead to worse patient outcomes, with advanced cancers being more difficult to treat effectively.

Given these challenges, automating lung cancer detection using deep learning models holds significant promise. A reliable Convolutional Neural Network (CNN) can help alleviate the burden on radiologists by providing faster, more accurate preliminary screenings, enabling medical professionals to focus on more complex cases. By developing an automated tool that consistently identifies cancerous CT-Scans with high accuracy, the model can potentially save lives through earlier diagnosis and more timely interventions.

This problem is vital because it addresses the need for scalable, efficient, and accurate diagnostic tools in a healthcare system that faces increasing demands, especially in the context of cancer detection and treatment.

The data analytics problem that I am analyzing is the development of an automated system for detecting lung cancer from chest CT-Scan images using deep learning techniques. Specifically, the challenge involves building a model that can accurately classify images as either cancerous or non-cancerous, with the goal of identifying different types of lung cancer early and reliably. By leveraging data analytics and machine learning, this solution aims to improve diagnostic accuracy, reduce the workload on medical professionals, and facilitate timely treatment for patients, ultimately improving survival rates and healthcare outcomes.

## Project Importance

This project was selected due to the critical need for early and accurate lung cancer detection, which can dramatically improve patient survival rates. Lung cancer is the leading cause of cancer-related deaths worldwide, with a 5-year survival rate of only 19% across all stages. However, early-stage detection increases the survival rate to nearly 60% (American Cancer Society, 2023). Traditional diagnostic methods, such as manual analysis of CT-Scan images, are time-intensive and prone to human error, especially given the volume of scans that radiologists must process daily. By using machine learning, specifically deep learning models like Convolutional Neural Networks (CNNs), this project aims to automate and improve the detection process, helping to catch cancer earlier when treatment is most effective. Automated tools powered by artificial intelligence (AI) can assist radiologists by acting as a second opinion, reducing diagnostic errors and speeding up the diagnostic process (Giger, 2018).

The importance of this project extends beyond improving individual patient outcomes. It addresses a broader healthcare challenge by introducing a scalable and efficient solution that can benefit healthcare systems globally, particularly in regions with a shortage of trained

radiologists. The beneficiaries of this project include not only patients, who will receive faster and more accurate diagnoses, but also healthcare professionals, who will have access to advanced tools that reduce their workload and improve their diagnostic capabilities.

Furthermore, this technology can aid in medical research by providing large-scale, automated analysis of imaging data, potentially uncovering new insights into the patterns of lung cancer. As the field of AI in healthcare continues to evolve, this project contributes to the growing body of work demonstrating how AI-driven diagnostics can revolutionize medical practice and patient care (Hosny, Parmar, Quackenbush, Schwartz, & Aerts, 2018).

## Background

Lung cancer is one of the deadliest cancers globally, accounting for approximately 18% of all cancer-related deaths (World Health Organization, 2020). Early detection through imaging, particularly using chest CT-Scans, has become one of the primary methods for identifying lung cancer in its early stages, significantly improving patient outcomes (National Lung Screening Trial Research Team, 2011). However, the manual interpretation of CT-Scans by radiologists is both labor-intensive and prone to diagnostic errors, particularly when dealing with large volumes of scans in busy clinical environments. Studies show that even experienced radiologists can miss early-stage lung cancer lesions due to their small size or subtle appearance (Balata et al., 2021). To address these limitations, artificial intelligence (AI) and machine learning have emerged as valuable tools for assisting radiologists in detecting cancer more accurately and efficiently. By developing deep learning models that can automatically analyze medical images, healthcare providers can reduce the burden on radiologists while simultaneously improving diagnostic precision.

In recent years, Convolutional Neural Networks (CNNs), a class of deep learning algorithms, have demonstrated remarkable success in medical imaging tasks, including cancer detection (Litjens et al., 2017). CNNs excel at identifying spatial hierarchies in images, making them ideal for analyzing complex medical images such as CT-Scans, where identifying small, irregular patterns is crucial for early cancer detection. Moreover, studies have shown that CNN-based models can achieve similar or even better accuracy than human experts in specific diagnostic tasks (Ardila et al., 2019). The integration of CNN models in lung cancer screening can help bridge the gap between early detection and treatment, providing a more scalable solution to screening large populations. This project builds upon these advancements in deep learning by developing a CNN model specifically tailored to classify chest CT-Scan images as either cancerous or non-cancerous, with the potential to improve both the speed and accuracy of lung cancer diagnosis.

## Data Set Description

The dataset for this project consists of labeled chest CT-Scan images, which are organized into separate categories based on the presence and type of lung cancer. The dataset includes images from patients diagnosed with various types of lung cancer, such as adenocarcinoma, large cell carcinoma, and squamous cell carcinoma, as well as images from healthy individuals labeled as “normal.” The images are divided into three main subsets: training, validation, and testing, ensuring that the model is trained, fine-tuned, and evaluated on different samples to prevent overfitting. Each subset contains a balanced representation of cancerous and non-cancerous cases to ensure that the model is exposed to a wide variety of examples. The dataset also includes metadata, such as the cancer stage and location (e.g.,

"adenocarcinoma\_left.lower.lobe\_T2\_N0\_M0\_Ib"), which may be useful for further analysis or model extension. The CT-Scan images have been pre-processed to a uniform size and resolution, allowing them to be fed directly into a Convolutional Neural Network (CNN) for classification.

In terms of quantity, the dataset includes thousands of images across the various categories, providing sufficient data for training a robust deep learning model. Data augmentation techniques, such as rotation, flipping, and zooming, can also be applied to artificially increase the size of the dataset, which is important for improving model generalization. The images are grayscale, as color is not relevant in this context, and the focus is on the structure and patterns within the lungs. This dataset provides a solid foundation for the project, as it reflects real-world diagnostic challenges that radiologists face in detecting cancer from medical imaging. Given the complexity of medical images and the variability of cancer appearances, the dataset's diversity in terms of patient cases, cancer types, and stages is essential for building a model that generalizes well to unseen data. The goal is to use this data to train a model capable of distinguishing between cancerous and non-cancerous lung CT-Scans, aiding in early cancer detection and diagnosis.

## Data Analytics Tools

For this project, several data analytics and machine learning tools will be utilized to process, analyze, and model the chest CT-Scan images. Python will be the primary programming language due to its extensive libraries for data science and machine learning. Specifically, TensorFlow and Keras will be used to build and train the Convolutional Neural Network (CNN) model, as these libraries offer powerful functionalities for developing deep learning models with a high level of abstraction. TensorFlow provides the flexibility to work with large datasets and

GPU acceleration for faster training. NumPy and Pandas will be used for data manipulation and preprocessing, particularly for handling any accompanying metadata and organizing the dataset into train, validation, and test splits. Additionally, Matplotlib and Seaborn will be used for visualizing the data distribution, model performance metrics, and training results, providing valuable insights throughout the project.

To further enhance model performance, OpenCV and Pillow (PIL) will be used for image preprocessing tasks, such as resizing, normalization, and augmentation. These libraries will help standardize the input images before they are fed into CNN. For model evaluation, Scikit-learn will be employed to generate performance metrics like accuracy, precision, recall, F1-score, and the confusion matrix, helping to assess how well the model distinguishes between cancerous and non-cancerous images. Additionally, KerasTuner may be used to optimize hyperparameters like learning rate, batch size, and number of layers in the CNN, ensuring the model achieves optimal performance.

## Project Milestones

### Dataset Exploration and Preprocessing

The project begins with acquiring and understanding the dataset. This involves loading the chest CT-Scan images and performing exploratory data analysis (EDA) to understand the distribution of classes (cancerous vs. non-cancerous) and the overall structure of the dataset. Preprocessing tasks include resizing images, normalizing pixel values, and ensuring labels are in the correct format. Additionally, data augmentation techniques such as rotation, flipping, and zooming will be applied to increase the diversity of the training set and prevent overfitting.

## **Model Development: Basic CNN Architecture**

During this milestone, a basic Convolutional Neural Network (CNN) will be designed and implemented to classify the CT-Scan images. The model will include essential layers such as Conv2D, MaxPooling, Flatten, and Dense layers to extract features from the images and perform classification. The CNN model will be trained on the preprocessed dataset, using metrics such as accuracy and loss to monitor its performance. The model's performance on the validation set will also be observed to detect potential overfitting.

## **Model Optimization and Transfer Learning**

To improve model performance, hyperparameter tuning will be performed, adjusting parameters such as learning rate, batch size, and number of layers. Additionally, transfer learning will be explored by leveraging pre-trained models like VGG16 or ResNet50 to enhance the CNN's performance on medical images. Fine-tuning of these models will be conducted to adapt them to the specific chest CT-Scan dataset. The optimized model will then be evaluated using performance metrics such as precision, recall, F1-score, and AUC-ROC.

## **Model Evaluation and Testing**

In this milestone, the final model will be tested on the test dataset to evaluate its ability to generalize to unseen data. This step includes generating a confusion matrix, ROC curve, and classification report to thoroughly assess the model's performance in classifying cancerous and non-cancerous images. Special attention will be paid to minimizing false negatives, as early

cancer detection is crucial in clinical settings. Any misclassifications will be analyzed to further understand model limitations and areas for improvement.

These milestones ensure a structured approach, taking the project from data preparation and model development to practical application. Each phase builds upon the last to create a fully functional lung cancer detection system.

<p><b>Stage 1</b></p>	<p>Stage 1 focused on defining the project's scope and objectives, centering on the classification of chest CT scans to aid in lung cancer detection. This involved a detailed review of the dataset to identify potential challenges, such as variations in image quality and potential class imbalances that could impact model accuracy. By understanding the characteristics and limitations of the data early on, I could establish a clear plan for data handling and model development, setting a strong foundation for the rest of the project.</p> <p>I concentrated on data profiling and preparation, ensuring the dataset was in optimal condition for training. This stage required resizing the images for consistency, normalizing pixel values, and implementing data augmentation to introduce variability. These steps were critical for enhancing the dataset's quality and helping the model generalize better to unseen data. Addressing these preprocessing tasks provided a stable and reliable dataset, allowing the models to achieve more accurate and meaningful results.</p>
<p><b>Stage 2</b></p>	<p>Stage 2 I focused on exploring and understanding the chest CT scan dataset through various visualization techniques. I created class distribution bar charts to identify imbalances in the dataset, which highlighted the need for augmentation to address overrepresentation of certain categories. I also generated pixel intensity histograms to analyze the brightness and contrast</p>

	<p>levels of the images, which helped me make informed preprocessing decisions such as normalization to ensure consistent data quality.</p> <p>Additionally, I created sample image grids of both the original and augmented images to visually inspect the dataset's quality and the effectiveness of the augmentation techniques I applied. These grids helped me confirm that transformations like flipping, rotation, and zooming maintained the critical features necessary for classification. Through these visualizations, I gained a clearer understanding of the dataset's structure, which was essential for preparing the data for modeling and addressing challenges such as class imbalance and variability.</p>
<b>Stage 3</b>	<p>At stage 3, the project was data modeling, where two machine learning models were implemented, a Basic CNN and a Transfer Learning model using VGG16. The models were trained on the prepared dataset, leveraging preprocessing and data augmentation to enhance their generalization capabilities. The performance of each model was evaluated using validation metrics like accuracy, precision, and recall, which helped identify their strengths and limitations. Together, these stages offered a deeper understanding of the dataset and the effectiveness of various modeling approaches for medical image classification, while also setting the stage for further improvements</p>

	<p>The project explored advanced architectures like ResNet and EfficientNet to improve classification performance on the chest CT scan dataset. These models were selected for their ability to handle complex image classification tasks and their proven effectiveness in medical imaging applications. The focus was on evaluating how well these models could address challenges identified in earlier stages, such as subtle feature detection and class imbalance.</p> <p>The implementation involved fine-tuning pre-trained versions of these models using transfer learning techniques, like VGG16. Key steps included freezing initial layers to retain pre-trained feature extraction capabilities while fine-tuning deeper layers to adapt to the specific characteristics of the CT scan dataset. Performance metrics like accuracy, precision, recall, and F1-score were analyzed to assess the effectiveness of these models in improving classification outcomes. Additionally, strategies such as hyperparameter tuning and further data augmentation were applied to enhance the models' generalization capabilities. These efforts provided critical insights into the strengths and limitations of advanced architecture, guiding recommendations for selecting a champion model for the task.</p>
<b>Stage 4</b>	<p>In the data modeling phase of the project, I worked on designing, implementing, and evaluating machine learning models to classify chest CT scan images into four categories: normal, adenocarcinoma, squamous cell carcinoma, and large cell carcinoma. Two primary approaches were utilized: a</p>

Basic Convolutional Neural Network (CNN) and a VGG16 Transfer Learning model. The CNN model was built from scratch to identify features within the dataset, while the VGG16 model leveraged pre-trained weights from the ImageNet dataset to adapt knowledge for this specific medical application.

The modeling process involved rigorous experimentation with hyperparameters, architecture modifications, and fine-tuning of the pre-trained model to improve classification accuracy. The evaluation of the models was carried out using metrics such as accuracy, precision, recall, and F1-score, alongside confusion matrices for detailed performance analysis. While both models showed moderate accuracy, 63.93% for the Basic CNN and 64.75% for the VGG16 model, the results highlighted areas where further refinement is needed.

In terms of results, the findings demonstrated the effectiveness of transfer learning in improving performance compared to a model trained from scratch. However, challenges such as class imbalance and variability in image quality were identified as limiting factors in achieving higher accuracy. These insights were critical in shaping the recommendations for future work, including exploring advanced architectures like ResNet or Inception and addressing dataset-related issues through augmentation or expansion. Overall, the data modeling phase provided valuable experience in applying AI to

	<p>medical imaging while highlighting the potential and challenges of such approaches.</p>
--	--

<p><b>Stage 1</b></p>	<p>During the first stage of the project, which focused on dataset exploration and initial understanding, several key insights were gained. First and foremost, I learned the importance of properly understanding the structure of the dataset, especially in the context of medical imaging, where each image represents vital diagnostic information. The dataset's organization into training, validation, and test sets, along with distinct labels for different types of lung cancer (adenocarcinoma, large cell carcinoma, squamous cell carcinoma) and normal cases, emphasized the need for clear labeling and class balance. I also realized that before diving into model building, taking time to explore the images themselves is crucial to ensuring that the data is clean, well-distributed, and suitable for training a model.</p>
<p><b>Stage 2</b></p>	<p>Data Profiling and Preparation, I learned the importance of thoroughly inspecting and cleaning the dataset to ensure high-quality inputs for the model. This included resizing images for uniformity, normalizing pixel values for consistent processing, and addressing class imbalances that could impact model performance. I also applied data augmentation techniques like rotation and flipping to increase data diversity and help the model generalize better.</p>

	<p>Overall, this stage taught me how essential data preparation is in building reliable machine learning models.</p>
<p><b>Stage 3</b></p>	<p>As part of the data visualization of the assignment, I learned the importance of visually exploring the dataset to gain a deeper understanding of its structure and quality. I discovered how crucial class distribution visualizations are in identifying imbalances that could significantly affect model training and evaluation. By creating pixel intensity histograms, I understood how variations in brightness and contrast within the images could influence the ability of models to learn distinguishing features, emphasizing the need for normalization and preprocessing.</p> <p>Additionally, I learned how sample image grids provide a straightforward yet powerful way to validate the effectiveness of data augmentation strategies. These grids allowed me to visually assess whether transformations such as flipping, rotation, and zooming preserved the key features necessary for classification tasks. I also realized how visualizing the data can reveal hidden issues, such as mislabeled or corrupted images, that could otherwise compromise the model's performance.</p>
<p><b>Stage 4</b></p>	<p>The data modeling stage of the assignment, I learned the intricacies of selecting and implementing machine learning models for medical image classification. I gained a deeper understanding of how different architectures, such as a Basic CNN and a Transfer Learning model using VGG16, can impact performance depending on the complexity of the dataset. I realized the</p>

	<p>importance of tailoring models to the specific needs of the task, such as freezing pre-trained layers in VGG16 to leverage existing knowledge while fine-tuning deeper layers for domain-specific learning.</p> <p>Through this process, I also learned how critical it is to evaluate models using metrics like accuracy, precision, recall, and F1-score to understand their strengths and limitations. For example, I observed how class imbalance in the dataset could affect the models' ability to generalize, highlighting the need for augmentation and other techniques to address this issue. I also understood how hyperparameter tuning, such as adjusting learning rates and batch sizes, plays a vital role in improving model performance. Finally, this stage taught me that achieving high performance in medical imaging tasks requires a careful balance between leveraging pre-trained models and adapting them to the nuances of the dataset, as well as addressing challenges like class imbalance and subtle feature distinctions.</p>
Stage 5	<p>Through this project, I gained a deeper understanding of the complexities involved in applying machine learning and deep learning to medical imaging, particularly in the context of classifying chest CT scans for cancer detection. I learned the importance of data preparation and the critical role it plays in model performance, from image resizing and normalization to addressing class imbalances. These steps reinforced the idea that high-quality data is the foundation of any successful AI project.</p>

I also developed a greater appreciation for the strengths and limitations of different modeling approaches. Working with a Basic CNN model taught me how to build architectures from scratch, while experimenting with the VGG16 Transfer Learning model highlighted the benefits of leveraging pre-trained networks for specialized tasks. Additionally, the challenges of achieving high accuracy underscored the importance of hyperparameter tuning, data augmentation, and fine-tuning pre-trained layers to adapt models to specific domains.

Beyond technical skills, I learned the value of critically analyzing results and identifying areas for improvement. For example, understanding the impact of biases in the dataset and recognizing limitations in model generalization provided insights into designing fairer and more robust AI systems. Moreover, this project emphasized the importance of ethical considerations, especially when dealing with sensitive medical data, and the need to align AI applications with trust and transparency principles.

# Data Profiling and Preparation

## Data Summary

The dataset used for this project consists of chest CT-Scan images categorized into cancerous and non-cancerous cases, with further differentiation among different lung cancer types, such as adenocarcinoma, large cell carcinoma, and squamous cell carcinoma. The dataset is divided into three main subsets: training, validation, and test sets, allowing for efficient model development, fine-tuning, and final evaluation. Each subset contains thousands of images that have been standardized in terms of size and format, ensuring that all images are ready for input into the machine learning model. Accompanying the images are labels that identify whether each scan corresponds to a cancerous or normal case, and in the case of cancerous images, the specific type of lung cancer is provided. This labeling provides the foundation for a multi-class classification task, where the goal is to distinguish between cancerous and non-cancerous cases and identify the specific type of cancer when present.

In addition to the images themselves, there may be metadata related to patient demographics or the stage of cancer, though this information is secondary to the image analysis and may not be used in the initial stages of model development. The dataset displays a class imbalance, with non-cancerous (normal) cases often outnumbering the cancerous cases. This imbalance is common in medical datasets and presents challenges during model training, as the model could be biased toward the majority class (normal). Techniques such as oversampling, undersampling, or synthetic data generation (e.g., SMOTE) may be used to address this issue during data preparation. Finally, data augmentation techniques like rotation, scaling, and flipping

will be applied to expand the dataset further, enhancing the model's ability to generalize across unseen data. This initial profiling indicates that the dataset is suitable for training a robust deep learning model, but careful attention must be given to handling the class imbalance and ensuring effective model generalization.

## **The importance of the data set**

The dataset is critical to the success of this project because it provides the foundation for training and evaluating the deep learning model that will be used for lung cancer detection. Medical imaging data, particularly CT-Scans, offers a rich source of information about the structure and conditions of the lungs, allowing the model to learn from real-world diagnostic cases. By analyzing these images, the model can potentially identify subtle patterns and anomalies that may be missed by human eyes, particularly in the early stages of lung cancer when signs are not always obvious. The ability to automatically classify images as cancerous or non-cancerous with high accuracy can lead to significant improvements in early cancer detection, which is crucial for increasing survival rates.

The dataset's importance also lies in its representation of various types of lung cancer, such as adenocarcinoma, large cell carcinoma, and squamous cell carcinoma, alongside normal, non-cancerous cases. This variety enables the model to not only detect cancer but to differentiate between the types of cancer, which is essential for personalized treatment plans and patient care. Additionally, the dataset's use in training a deep learning model for cancer detection has broader implications for the healthcare industry. If models trained on this dataset can be deployed effectively, they could support radiologists by providing faster, more accurate preliminary diagnoses, alleviating the burden of manual image analysis, and helping improve clinical

decision-making. The dataset serves as the cornerstone for developing such an AI-powered tool, making it a crucial asset in advancing automated healthcare diagnostics.

## Source of the Dataset

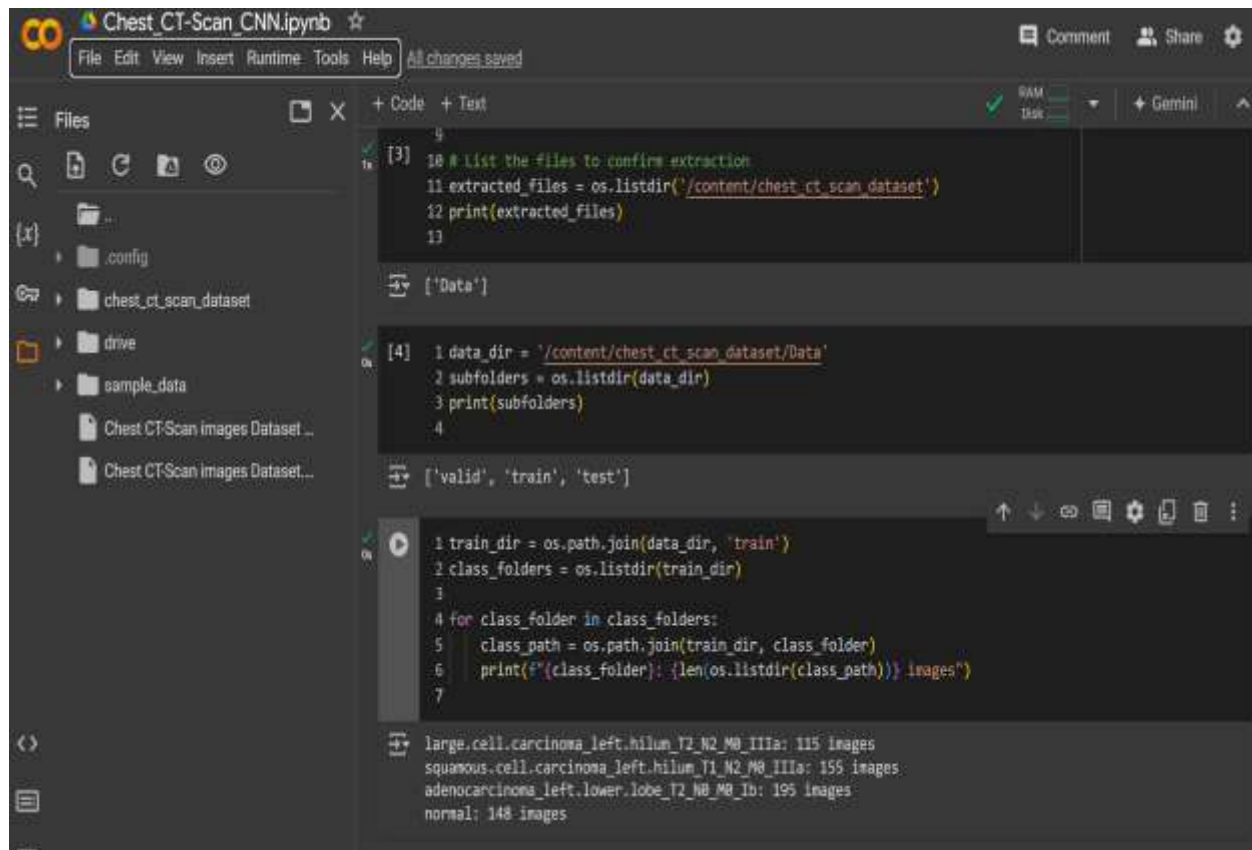
- I. **Dataset Name:** Chest CT-Scan Images Dataset
- II. **Dataset Provider:** Mohamed Hany on Kaggle
  - a. [Kaggle Dataset Link](#)
- III. **License:** The dataset is licensed under the Open Database License (ODbL) v1.0, as provided by the Open Data Commons.
  - a. [ODbL License Details](#)

## Attribution and Usage Rules

According to the Open Database License (ODbL) v1.0, any use, redistribution, or modification of this dataset must:

- I. Attribute the original source by crediting both the dataset provider (Mohamed Hany) and the Kaggle platform.
- II. State any modifications if the dataset is altered, transformed, or built upon.
- III. Share under the same license if redistributed, ensuring that the same open data license applies.

Any use of the "Chest CT-Scan Images" dataset should clearly credit **Mohamed Hany on Kaggle** as the original source, and any modified versions should also be shared under the ODbL license.



```
10 # List the files to confirm extraction
11 extracted_files = os.listdir('/content/chest_ct_scan_dataset')
12 print(extracted_files)
13
14 ['Data']

1 data_dir = '/content/chest_ct_scan_dataset/Data'
2 subfolders = os.listdir(data_dir)
3 print(subfolders)
4
5 ['valid', 'train', 'test']

1 train_dir = os.path.join(data_dir, 'train')
2 class_folders = os.listdir(train_dir)
3
4 for class_folder in class_folders:
5     class_path = os.path.join(train_dir, class_folder)
6     print(f'{class_folder}: {len(os.listdir(class_path))} images')
7
large.cell.carcinoma_left.hilum_T2_N2_M0_IIIa: 115 images
squamous.cell.carcinoma_left.hilum_T1_N2_M0_IIIa: 155 images
adenocarcinoma_left.lower.lobe_T2_N0_M0_Ib: 195 images
normal: 148 images
```

The dataset is organized into three main subsets: **train**, **validation**, and **test**, with each subset containing images grouped into the following four categories:

- I. **Adenocarcinoma:** adenocarcinoma\_left.lower.lobe\_T2\_N0\_M0\_Ib
- II. **Large Cell Carcinoma:** large.cell.carcinoma\_left.hilum\_T2\_N2\_M0\_IIIa
- III. **Squamous Cell Carcinoma:** squamous.cell.carcinoma\_left.hilum\_T1\_N2\_M0\_IIIa
- IV. **Normal:** normal

**Class distribution in the training set:**

- I. **Adenocarcinoma (left lower lobe):** 195 images
- II. **Large Cell Carcinoma (left hilum):** 115 images
- III. **Squamous Cell Carcinoma (left hilum):** 155 images
- IV. **Normal (non-cancerous):** 148 images

This gives a total of **613 images** in the training set. Since this is an image dataset, it doesn't have traditional rows and columns like tabular data. Instead, each image can be considered as one data point (row).

### Training Set:

**613** Images belong to 4 classes.

These 4 classes are:

- I. **Adenocarcinoma:** 195 images
- II. **Large Cell Carcinoma:** 115 images
- III. **Squamous Cell Carcinoma:** 155 images
- IV. **Normal (non-cancerous):** 148 images

This result indicates that the training set is composed of 613 labeled chest CT-Scan images, divided across these four categories. These images are used by the model to learn and adjust its weights through repeated exposure to patterns present in each class.

### Validation Set:

- I. **72 images** belonging to **4 classes**.

- II. These images are held back during training and used to validate the model's performance after each epoch.

The 72 images in the validation set are also split across the same four classes (adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal). This set is important for ensuring that the model generalizes well and isn't simply memorizing the training data.

## Key Points:

- I. **Class Distribution:** The distribution of images across the four classes in both the training and validation sets seems relatively balanced, though some classes (e.g., large cell carcinoma) have fewer samples than others (e.g., adenocarcinoma).
- II. **Training Set Size:** With 613 images, this training set size is on the smaller side for training deep learning models, especially for a complex problem like cancer detection. However, the use of **data augmentation** (e.g., rotation, zooming, flipping) helps by artificially increasing the variety of images, allowing the model to learn more effectively from this limited data.
- III. **Validation Set:** The 72 images in the validation set are used for evaluating model performance during training, providing an unbiased evaluation of how well the model is learning to classify images without influencing the training process. The size is smaller compared to the training set, but it serves as a snapshot of how well the model generalizes to unseen data.

## 1. Training Dataset

The training dataset is used to teach the model. The model looks at the input data (in this case, CT-Scan images) and their corresponding labels (cancerous vs. non-cancerous). It uses these labels to learn the patterns and features of the data to differentiate between cancerous and

non-cancerous cases, as well as between different cancer types (adenocarcinoma, large cell carcinoma, and squamous cell carcinoma).

**Size:** The training set consists of 613 images, distributed across four classes:

- I. Adenocarcinoma: 195 images
- II. Large Cell Carcinoma: 115 images
- III. Squamous Cell Carcinoma: 155 images
- IV. Normal (non-cancerous): 148 images

**Purpose:**

The model learns from this data, adjusting its internal weights to minimize the error in predicting the correct class for each image. During the training process, the model repeatedly goes over the training data (epochs), refining itself to become better at distinguishing between the categories.

## 2. Validation Dataset

The validation dataset is used during training to evaluate the model's performance without directly influencing the training process. It helps to check whether the model is overfitting (learning too much from the training data and failing to generalize to new data) or underfitting (failing to learn enough from the training data).

**Size:** Typically, a validation set is smaller than the training set. It is a subset of the entire dataset, held back during training.

**Purpose:**

The model’s performance is assessed on this data after each epoch or after every few epochs.

The goal is to monitor metrics such as accuracy, loss, precision, and recall. This dataset helps in tuning hyperparameters (e.g., learning rate, number of layers) and ensures that the model performs well on unseen data.

**No Training Influence:** The validation set does not directly update the model's weights but provides an objective measure of how well the model is likely to perform on new data.

**Why Both Sets Are Needed:**

- I. **Training Dataset:** Essential for learning the underlying patterns in the data. Without this, the model would not know how to classify new images.
- II. **Validation Dataset:** Critical for assessing how well the model generalizes to unseen data and prevents the model from being overfit to the training data.

**Data Definition/Data Profile**

Field/Variable	Definition	Data Type	Outliers	Frequency of Nulls	Potential Quality Issues
Image Files	Chest CT-Scan images of patients' lungs, categorized into cancerous and non-cancerous. Each image corresponds to one class.	Image	Unlikely	None	Variability in image size, quality, resolution. Images may be noisy or blurry.

<b>Labels/Classes</b>	Label indicating whether the image belongs to adenocarcinoma, large cell carcinoma, squamous cell carcinoma, or normal.	Categorical	No outliers expected (fixed categories)	None	Class imbalance between different types of cancer (e.g., fewer large cell carcinoma images).
<b>Image Dimensions</b>	Dimensions of each image (256x256 pixels).	Integer	Potential for images of varying sizes	None	Inconsistent image sizes could affect model training unless resized uniformly.
<b>Pixel Values</b>	Grayscale pixel values for each image (ranging from 0 to 255 before normalization).	Integer	Unlikely	None	Pixel values need normalization (scaling between 0-1) for training.
<b>Training Set Size</b>	Number of images in the training set: 613 images across 4 classes.	Integer	N/A	None	Smaller dataset size may affect model generalization. Data augmentation is necessary.
<b>Validation Set Size</b>	Number of images in the validation set: 72 images across 4 classes.	Integer	N/A	None	Small validation set size may not capture all variations, affecting performance evaluation.

## Data Preparation

### Data Preparation Process and Tools

In this project, the goal is to build a deep learning model using **chest CT-Scan images** to classify between normal cases and different types of lung cancer. Preparing the data is a critical step to ensure the model can effectively learn from the dataset and generalize well to new data.

#### Process for Data Preparation:

##### 1. Data Extraction:

- I. **Process:** The first step is to extract the dataset from the ZIP file into a working directory.
- II. **Tools:** We will use Python's Zip file module to extract the data into directories for training, validation, and testing.

## 2. Data Exploration:

- I. **Process:** Once the data is extracted, we will explore the directory structure and check the distribution of images across the different classes (normal, adenocarcinoma, squamous cell carcinoma, and large cell carcinoma).
- II. **Tools:** We will use OS (for navigating the file system) and print statements to inspect the number of images per class and ensure everything is correctly structured.

## 3. Data Cleaning:

- I. **Process:** This involves checking the data for any corrupted or missing images, inconsistent file names, or improper labels. We may also ensure that the images are correctly categorized.
- II. **Tools:** Python and libraries such as PIL (Python Imaging Library) can be used to open and verify the integrity of image files.

## 4. Image Resizing and Normalization:

- a. **Process:** We need to resize all images to a consistent shape (e.g., 128x128 or 256x256 pixels) because deep learning models typically require fixed-size inputs.

We will also normalize the pixel values to a range of [0, 1] (from the original [0, 255]) for better model performance.

b. **Tools:**

- i. **Keras ImageDataGenerator:** This is part of the TensorFlow/Keras framework and is used to resize and rescale images.
- ii. **PIL** or **OpenCV** can also be used for image preprocessing.

5. **Data Augmentation:**

- a. **Process:** To increase the diversity of the training data, we apply data augmentation techniques such as random rotations, flips, zooms, and translations. This helps prevent overfitting, especially with smaller datasets.

b. **Tools:**

- i. **Keras ImageDataGenerator:** This allows for on-the-fly data augmentation such as rotation, zooming, flipping, and shifting.
- ii. Other libraries like **Albumentations** can also be used for more advanced augmentations.

6. **Splitting Data into Training, Validation, and Test Sets:**

- I. **Process:** If the dataset is not already split into training, validation, and test sets, we will split it ourselves. The training set is used to train the model, the validation set to tune hyperparameters and prevent overfitting, and the test set to evaluate the final performance.

- II. **Tools:** We can either manually split the dataset using Python (OS library) or use utilities in **Keras ImageDataGenerator** to load and split the data automatically.

## 7. **Handling Class Imbalance:**

- a. **Process:** In medical datasets, it's common to have an imbalance between different classes (e.g., more normal cases than cancer cases). We will handle this imbalance by either:
  - i. Oversampling the minority class (creating more instances of underrepresented classes).
  - ii. Using data augmentation techniques for the minority class.
- b. **Tools:** Data augmentation via **ImageDataGenerator** or **SMOTE** (Synthetic Minority Over-sampling Technique) for oversampling.

## **Tools for Data Preparation:**

- I. **Python:**
  - a. Python is the main language for data handling, extraction, and processing.
- II. **TensorFlow/Keras:**
  - a. **Keras ImageDataGenerator:** This tool allows for on-the-fly preprocessing and augmentation of images. It can be used to resize, normalize, and augment the images during training.

## **PIL (Python Imaging Library) or OpenCV:**

- These libraries can be used for image processing tasks like resizing and converting image formats. They are also helpful for checking image integrity and correcting issues with images.

### OS Library:

- This Python library helps in navigating directories and managing files. It will be used to list directories, count images in each class, and create or move files during dataset splits.

### SMOTE (Optional, for Class Imbalance):

- If data augmentation is not sufficient to address class imbalance, **SMOTE** can be used to generate synthetic samples of the minority class. This tool creates new samples by interpolating between existing samples.

The combination of TensorFlow/Keras, ImageDataGenerator, PIL/OpenCV, and Python's os library will be used to clean, preprocess, and augment the dataset, making it ready for training the deep learning model. Addressing class imbalance through augmentation and careful splitting into training, validation, and test sets will ensure that the model learns effectively and generalizes well.

## Data Cleansing Process and Tools

In this project, where we are working with **chest CT-Scan images** to classify different lung cancer types and normal cases, **data cleansing** is an important step to ensure that the dataset is free from errors, inconsistencies, and issues that could negatively affect model performance.

Since we're dealing with image data, data cleansing focuses on image-specific tasks such as file integrity, resizing, normalization, and ensuring consistent labels.

### Process for Data Cleansing:

#### I. **Verify Image Integrity:**

- a. **Process:** The first step is to ensure that all images in the dataset are valid and not corrupted. This includes verifying that each image can be opened and processed, and there are no missing or unreadable images.
- b. **Tools:**
  - i. PIL (Python Imaging Library) or OpenCV can be used to open each image and check for any issues.
  - ii. If an image is corrupted or unreadable, it will be flagged and removed from the dataset.

### Handle Missing Data (Files):

- I. **Process:** If any image files are missing from the dataset, we need to identify these gaps and determine whether to replace, remove, or generate synthetic data (using augmentation techniques) to ensure that the dataset remains balanced.
- II. **Tools:**
  - a. **Python's os library** to iterate through directories and verify the number of images in each class.
  - b. **Data Augmentation** (via Keras or Albumentations) to generate new samples in case of missing or insufficient data.

### Uniform Image Resizing:

- I. **Process:** Images in the dataset may vary in size and resolution, which can pose challenges when training deep learning models. All images will be resized to a uniform shape (e.g., 128x128 or 256x256 pixels).
- II. **Tools:**
  - a. **PIL** or **OpenCV** to resize images uniformly across the dataset.
  - b. **Keras ImageDataGenerator** can also handle resizing during data loading.

### Normalize Pixel Values:

- I. **Process:** Normalization is critical in image processing because raw pixel values range from 0 to 255 and scaling them to a range between 0 and 1 improves model performance. This makes it easier for the model to learn and converge during training.
- II. **Tools:**
  - a. **Keras ImageDataGenerator** will rescale pixel values during data loading.
  - b. Alternatively, **NumPy** can be used to manually normalize the images.

### Ensure Consistent and Correct Labels:

- I. **Process:** The dataset is structured into folders corresponding to each class (e.g., adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal). We need to ensure that all images are placed in the correct folder and that there are no misclassified or mislabeled images.
- II. **Tools:**

- a. **Python's os library** to inspect the folder structure and ensure labels match the contents.
- b. If manual inspection is required, we can look at random samples from each class.

## Remove Duplicates:

- I. **Process:** Duplicate images can affect model training by skewing the model's ability to generalize. We will identify and remove duplicate images.
- II. **Tools:**
  - a. **ImageHashing** (via libraries like imagehash) or manual inspection to detect and remove duplicate images.

## I. **Augment Data to Handle Class Imbalance:**

- II. **Process:** If there is class imbalance (e.g., significantly fewer images for one type of cancer than others), we can use data augmentation to create additional samples for the minority classes. This will ensure that the model does not become biased toward the majority class.

- III. **Tools:**
  - a. **Keras ImageDataGenerator** to apply augmentations such as rotations, zooms, flips, etc.

## Tools for Data Cleansing:

- I. **PIL (Python Imaging Library):**
  - a. Used to open, resize, verify, and manipulate images.

## II. **OpenCV:**

- a. Another library for handling image-related tasks such as resizing, image transformations, and file integrity checks.

## III. **Keras ImageDataGenerator:**

- a. Used to normalize, augment, and preprocess images during data loading and model training.

## IV. **os Library:**

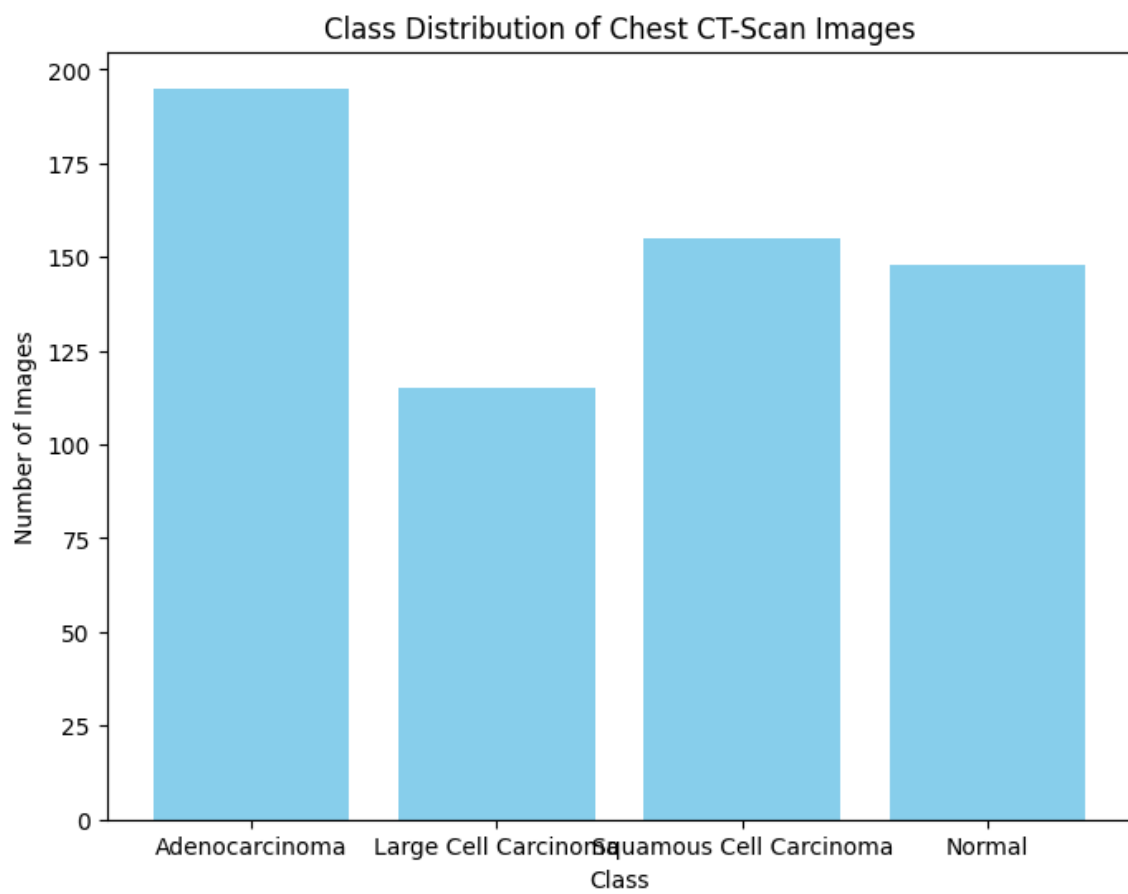
- a. Essential for navigating directories, counting files, and managing file structures.

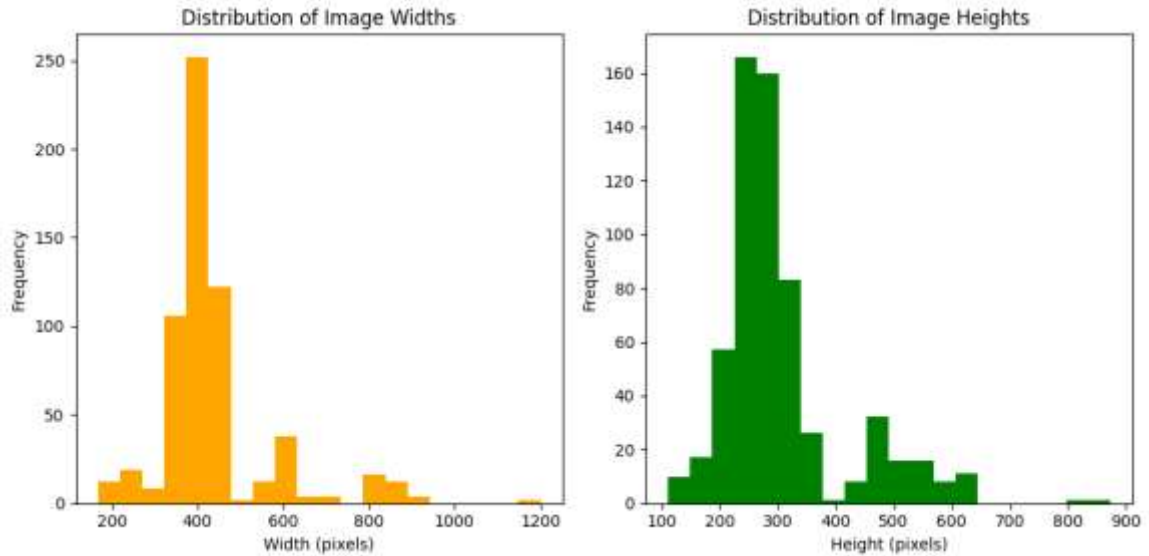
## V. **imagehash:**

- a. A library that can be used to detect and remove duplicate images by computing hash values for each image.

By following these steps and using the mentioned tools, we will cleanse the dataset to ensure it is of high quality for training the deep learning model. This includes verifying image integrity, ensuring correct labels, resizing images, normalizing pixel values, removing duplicates, and addressing class imbalance through data augmentation. Clean, consistent data will help ensure the model learns efficiently and generalizes well to new data.

# Data Visualizations





## What insights do they provide?

This visualization shows the number of images per class: adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal. The chart helps identify class imbalances in the dataset. For instance, there are fewer images for Large Cell Carcinoma than for other classes, which could lead to biased model predictions if not addressed.

What did you discover that alters the scope of expected results of the project? What additional ones could be useful?

## Impact on the Project:

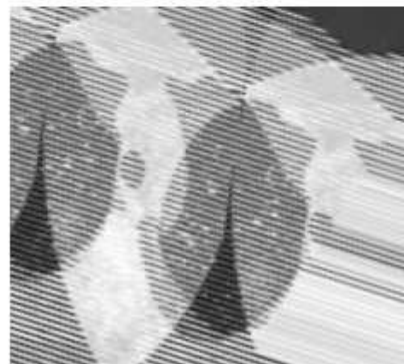
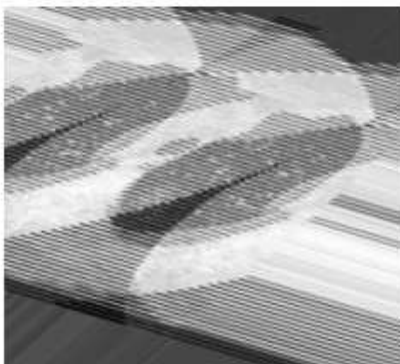
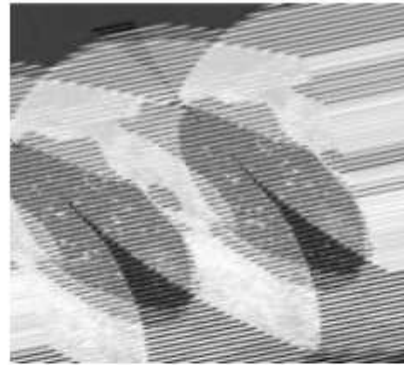
- I. **Class Imbalance:** The imbalance observed here suggests that the model might have difficulty learning equally from all classes. Therefore, techniques like data augmentation or oversampling of the minority classes (e.g., large cell carcinoma) may be necessary.

- II. **Scope of Expected Results:** This imbalance could alter the expected performance of the model, particularly for minority classes, and might lead to lower sensitivity for certain cancer types unless mitigation strategies are applied.

## Data Augmentation

This visualization shows the effect of data augmentation techniques (rotation, zoom, translation, and flipping) applied to a sample image from the adenocarcinoma class.

Augmentation artificially increases the diversity of the training set, helping prevent overfitting and improving the model's ability to generalize to unseen data.



## Impact on the Project:

- I. **Data Augmentation:** This step is crucial given the relatively small size of the dataset. By applying transformations, we effectively expand the dataset, improving the model's ability to learn features across different orientations, scales, and lighting conditions.
- II. **Scope of Expected Results:** The use of augmentation should improve the model's robustness, reducing the risk of overfitting. However, aggressive augmentation may introduce artifacts that could confuse the model, so it's important to balance augmentation intensity.

### Potential Additional Visualizations:

- I. **Comparison of Original and Augmented Data:** Showing side-by-side comparisons of original and augmented images to verify that the transformations retain important features relevant for classification.
- II. **Impact of Augmentation on Model Performance:** Track how augmentation improves or changes the model's performance on the validation set.

### Insights and Discoveries

- I. **Class Imbalance:** The visualization of class distribution clearly shows that there is an imbalance, especially with fewer samples in certain cancer types like Large Cell Carcinoma. This requires us to address it with techniques like oversampling, undersampling, or weighted loss functions during model training.
- II. **Data Augmentation:** The augmentation visualization demonstrates the benefit of augmenting the dataset. It will allow the model to generalize better, especially given the relatively small size of the training set. Augmentation will play a key role in improving

model robustness and reducing overfitting, but care must be taken not to apply augmentations that distort critical diagnostic features.

## Scope Alterations Based on Discoveries

- I. **Handling Imbalanced Classes:** Due to the observed class imbalance, the original expectation of training a balanced model must be adjusted. The focus will shift toward mitigating class imbalance and ensuring the model performs well on minority classes.
- II. **Impact of Augmentation:** While augmentation helps improve generalization, the augmented data may alter the way the model perceives subtle features in the images. We need to carefully balance augmentation to avoid overfitting while preserving critical diagnostic features in the images.

## Additional Visualizations That Could Be Useful:

- I. **Confusion Matrix:** After training the model, a confusion matrix would help identify which classes the model struggles with, especially for imbalanced classes like Large Cell Carcinoma.
- II. **Model Learning Curves:** Plotting the training and validation accuracy/loss over time will help identify overfitting and inform decisions regarding early stopping or additional regularization.
- III. **AUC-ROC Curves for Each Class:** This will help in evaluating how well the model differentiates between each class and provide insights into the sensitivity and specificity for each cancer type.

## Descriptive Statistics

The Chest-CT dataset involves image data rather than traditional tabular data, the "fields" or "variables" (like columns in a table). In the case of image datasets, the primary variables we focus on pixel values, and we typically analyze statistics such as the mean, standard deviation, and pixel intensity distributions to better understand the data.

### 1. Pixel Values:

For image datasets, pixel values are the core variable. Images are composed of pixel grids, where each pixel has an intensity value that typically ranges from 0 to 255 for grayscale or 3 values for RGB images (one for each of the red, green, and blue channels).

### 2. Image Dimensions:

Images can vary in size (width and height), and understanding the distribution of image dimensions can be useful for resizing the images before training.

### 3. Class Distributions:

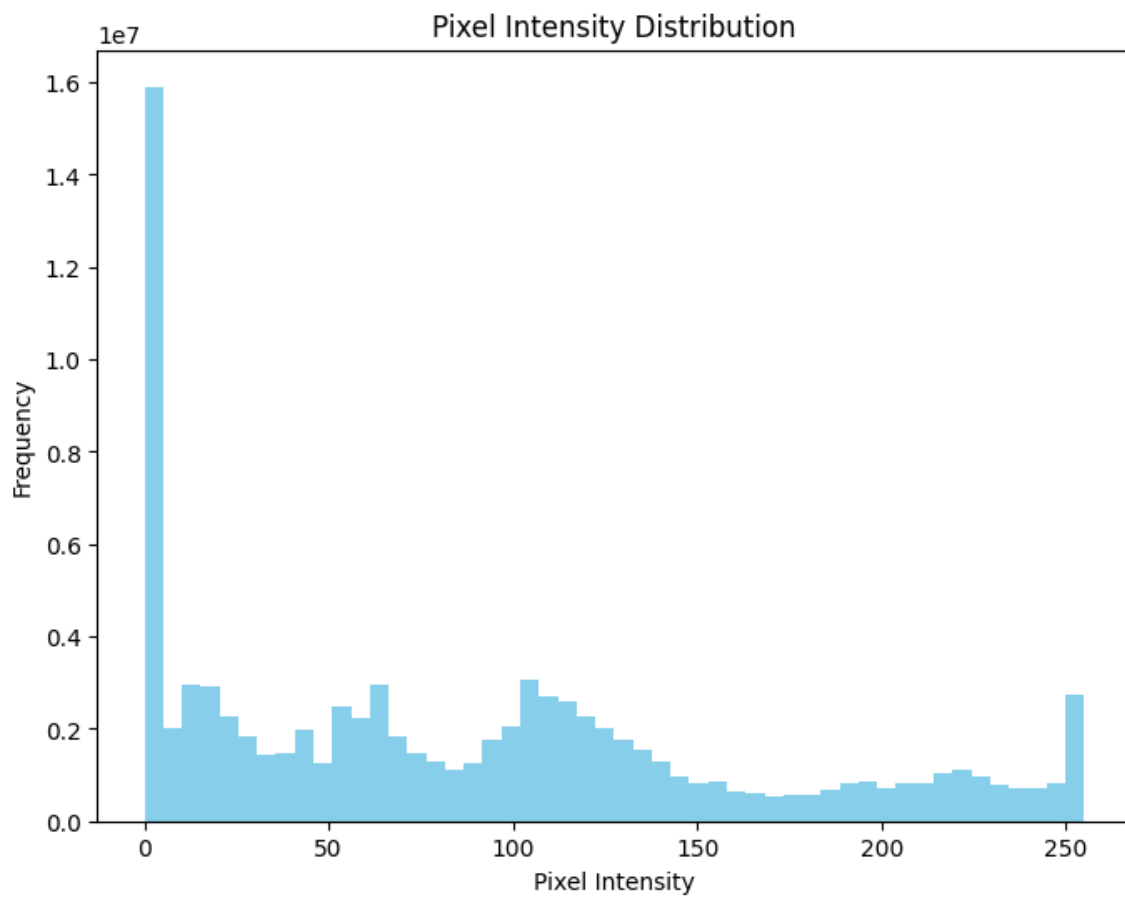
Understanding how the data is distributed across different classes (e.g., normal, adenocarcinoma, squamous cell carcinoma, large cell carcinoma) is important for calculate the following statistics for image dataset:

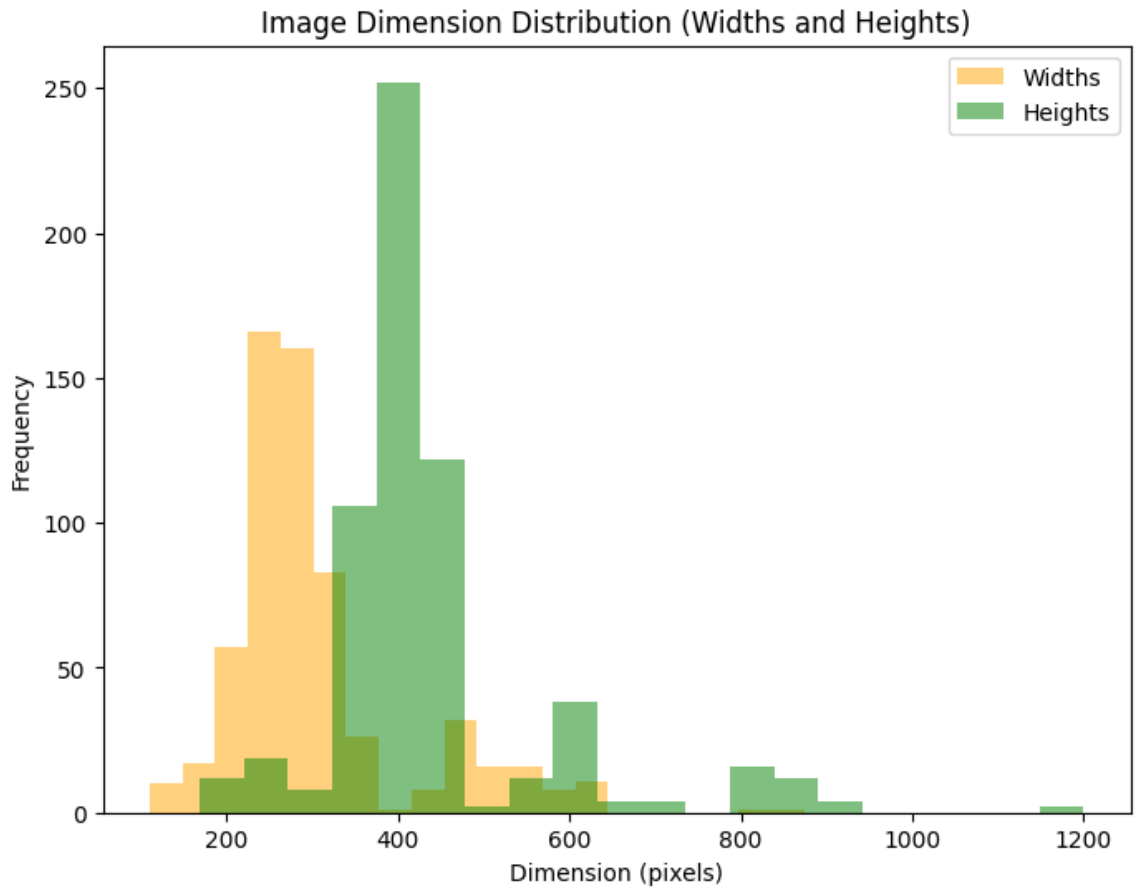
- I. Mean and standard deviation of pixel values.
- II. Quantiles of pixel intensities.
- III. Image dimension distributions.

#### IV. Class distribution.

### Assessing class balance and potential bias in the dataset.

- I. **Mean Pixel Value:** The average brightness of the images. If it is too low or high, it could indicate that images are mostly dark or overly bright, respectively.
- II. **Standard Deviation of Pixel Values:** This shows how spread out the pixel values are. A low standard deviation means the pixel intensities are concentrated around the mean, while a high value indicates diverse brightness levels.
- III. **Quantiles:** These give insights into how pixel values are distributed across the range, helping us understand the pixel intensity spread.
- IV. **Image Dimensions:** Analyzing the average width and height ensures that image preprocessing (such as resizing) is properly considered for the model.





## Data Visualization Definitions

### 1. Histogram

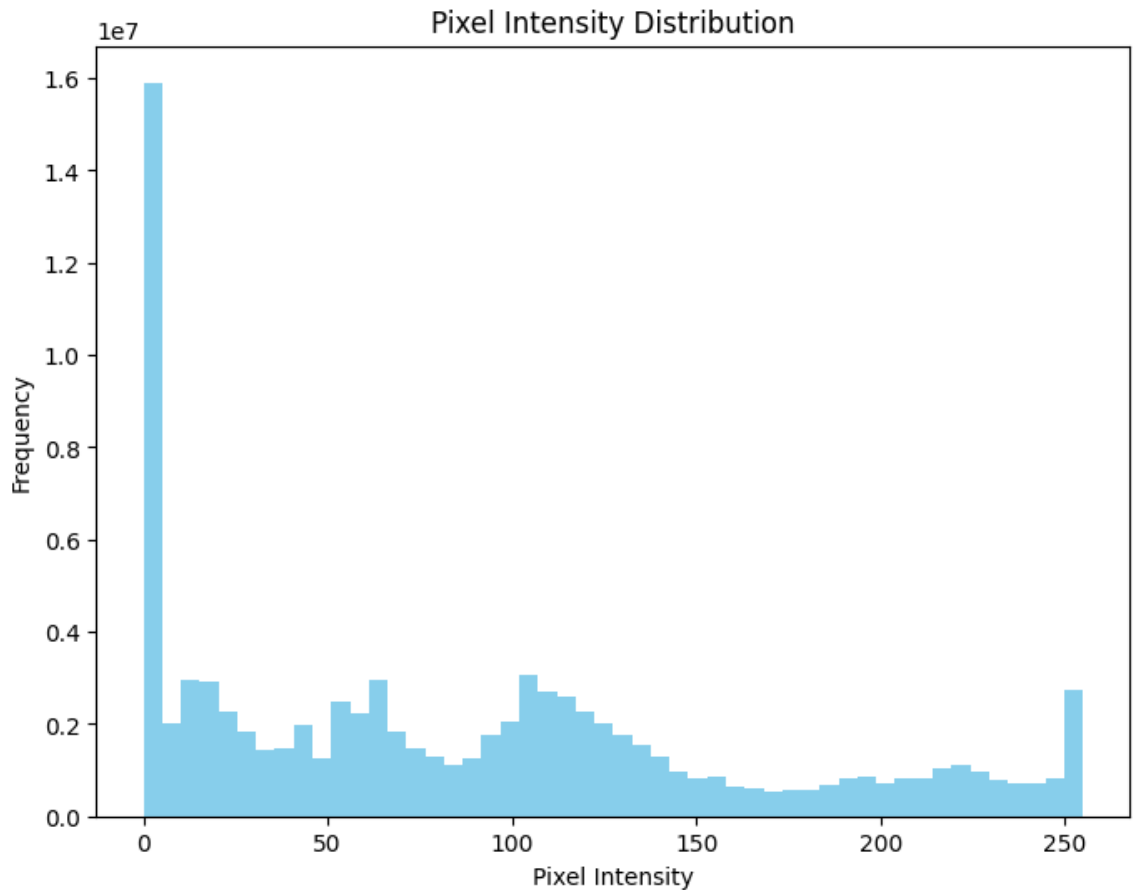
Histograms allow us to understand the distribution of pixel values, image dimensions, and class distributions in the dataset.

#### Use Cases:

- I. **Pixel Intensity Distribution:** A histogram of pixel intensities (ranging from 0 to 255 for grayscale images) shows how brightness levels are distributed across the

dataset. This can help identify overexposed or underexposed images, or other abnormalities in the data.

- II. **Image Dimension Distribution:** A histogram of image widths and heights provides an overview of the variability in image sizes, helping determine if resizing is necessary before training.
- III. **Class Distribution:** Histograms can also be used to show the number of images per class (e.g., normal, adenocarcinoma, large cell carcinoma, squamous cell carcinoma), helping identify any class imbalances.

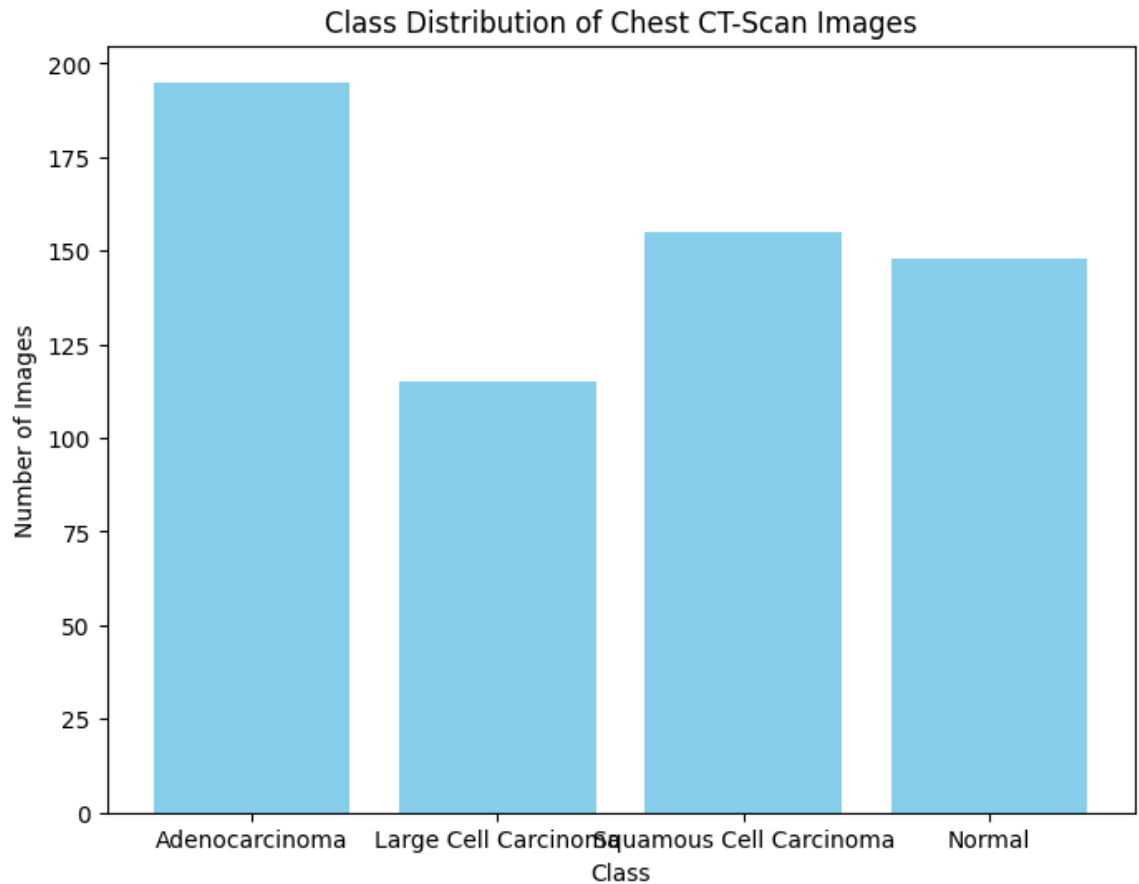


## 2. Bar Chart

Bar charts are particularly useful for categorical data, such as the distribution of images across different classes. This helps in identifying class imbalances and guiding the implementation of techniques like oversampling, undersampling, or weighted loss functions.

### Use Cases:

- I. **Class Distribution:** A bar chart showing the number of images in each class (e.g., normal vs cancerous classes) helps visualize class imbalances.
- II. **Model Accuracy and Loss per Epoch:** Bar charts can also visualize accuracy or loss across different epochs or for different models during evaluation.



### 3. Line Plot

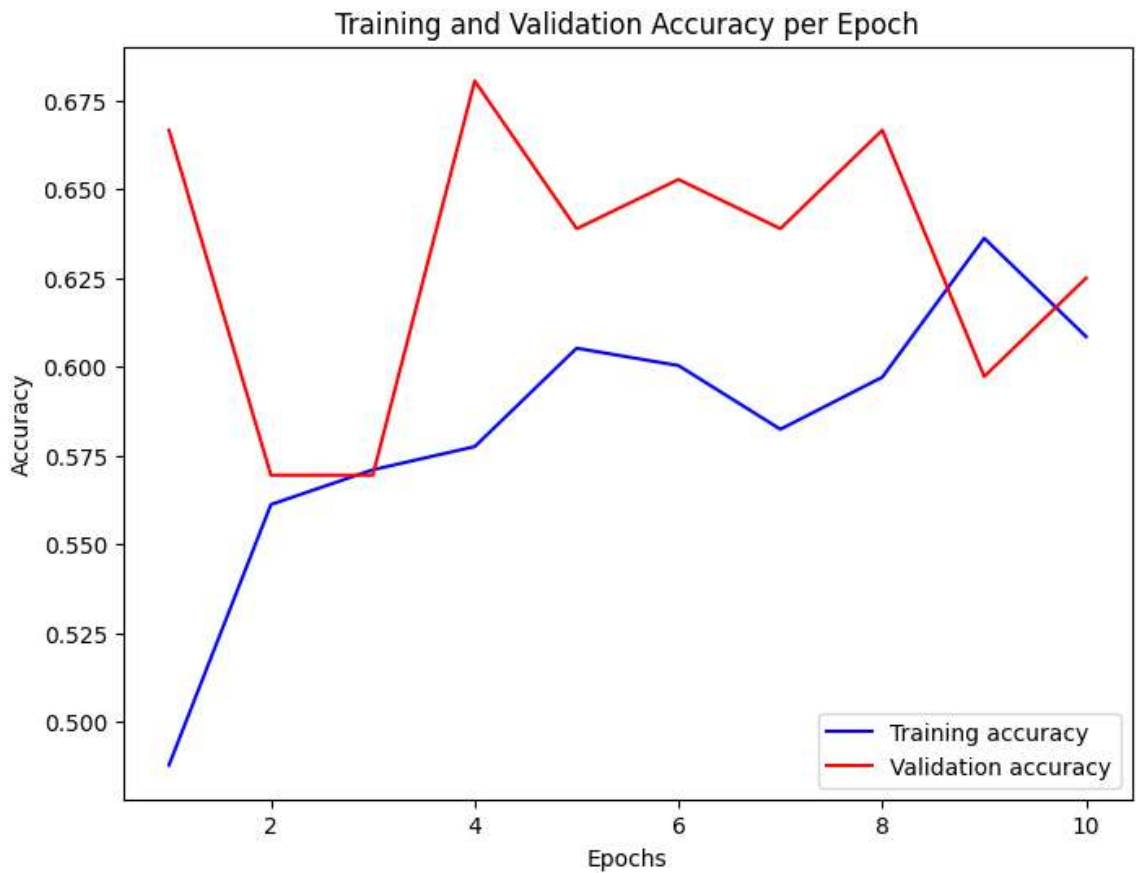
Line plots are often used to track model performance over time, such as training and validation loss or accuracy per epoch. This is crucial for monitoring overfitting, underfitting, and overall model convergence.

#### Use Cases:

- I. **Model Learning Curves:** Line plots show the progression of training and validation accuracy/loss during model training. This helps detect issues like

overfitting (when validation accuracy plateaus or drops while training accuracy increases).

- II. **Training Time:** Line plots can also be used to track how much time each epoch takes, giving insights into the efficiency of the training process.

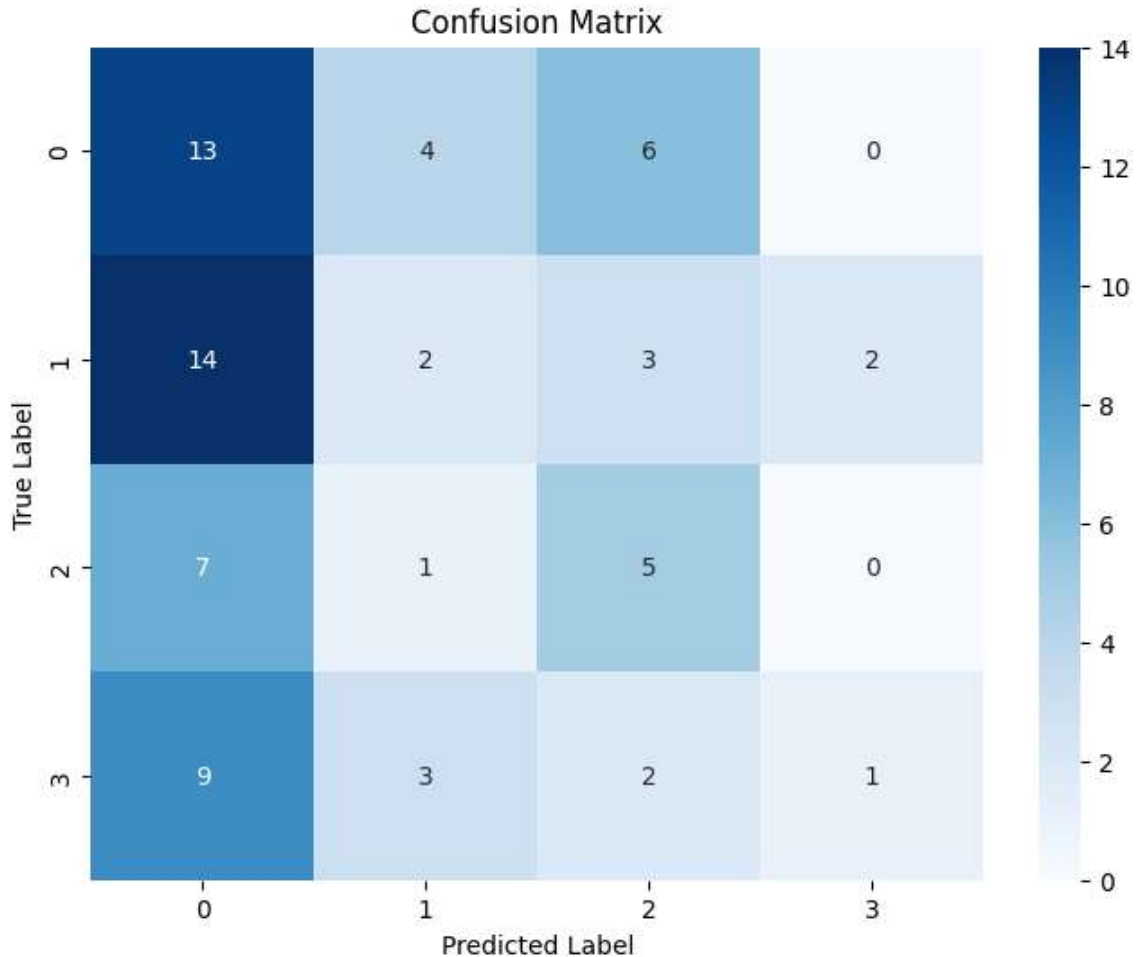


#### 4. Confusion Matrix

A confusion matrix is a powerful tool for evaluating classification model performance. It shows the counts of true positive, false positive, true negative, and false negative predictions for each class. This helps in identifying which classes the model is confusing with one another.

## Use Cases:

- I. **Classification Performance:** For multi-class classification, such as distinguishing between different types of lung cancer and normal cases, a confusion matrix shows where the model is making mistakes.
- II. **Error Analysis:** By visualizing the misclassifications, we can prioritize improvements, such as fine-tuning the model or improving data augmentation strategies.



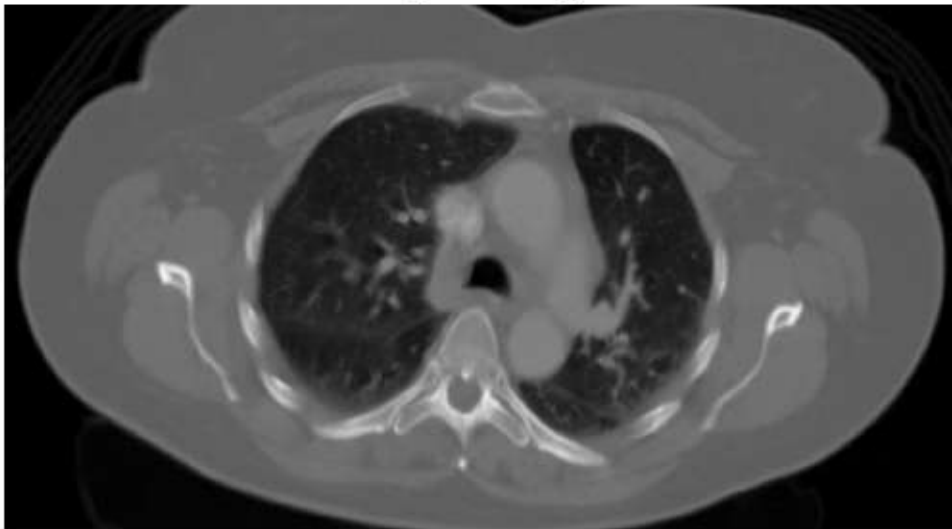
## 5. Sample Image Grid

Displaying a grid of sample images from each class helps visually inspect the data. This is important for confirming image quality, variety, and ensuring that augmentations are being applied correctly.

### Use Cases:

- I. **Visual Inspection of Augmentation:** Showing augmented images allows you to confirm that the data augmentation strategies (e.g., rotation, flipping, zoom) are creating useful variations of the original data.
- II. **Sample Image Display:** Displaying a few images from each class helps in visually assessing the quality and content of the dataset.

Original Image

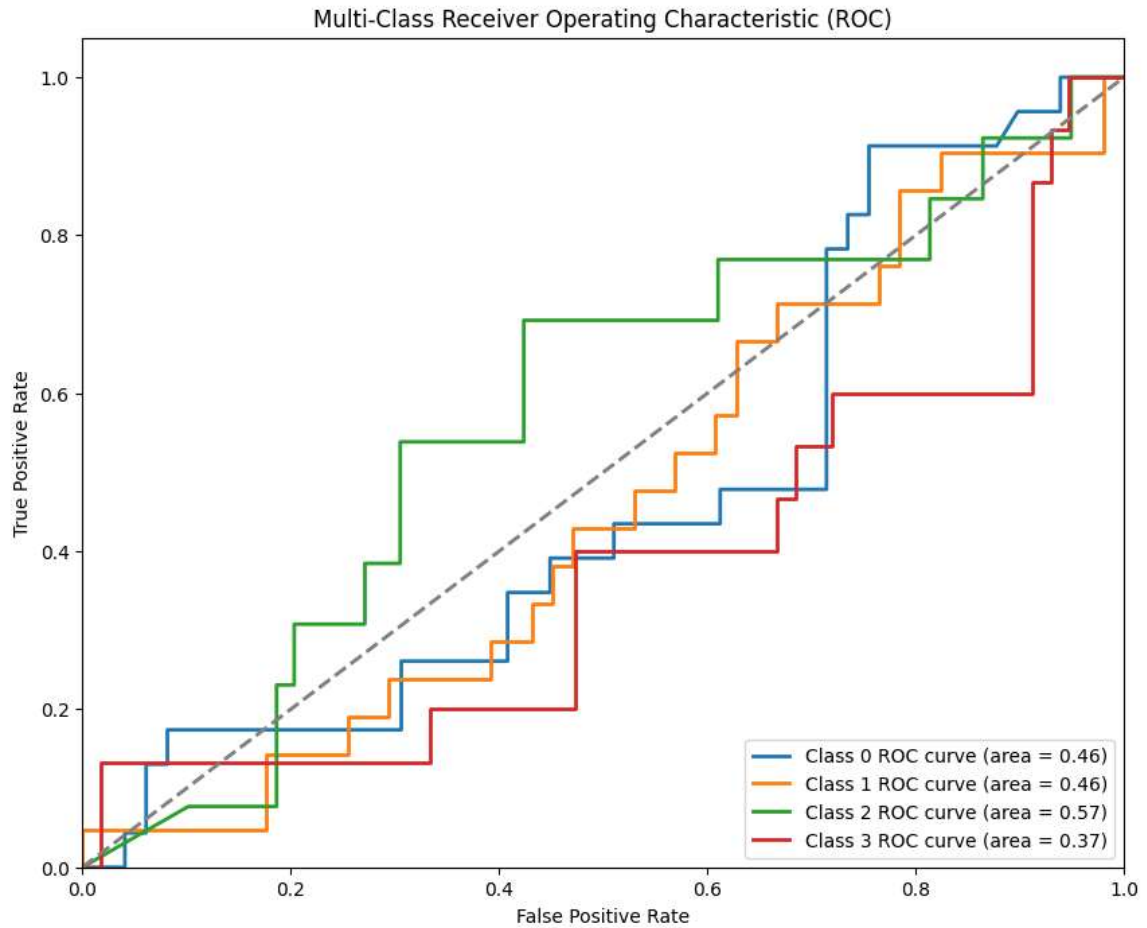


## 6. AUC-ROC Curve (Receiver Operating Characteristic)

The ROC curve helps evaluate the performance of a classification model by plotting the true positive rate against the false positive rate. The Area Under the Curve (AUC) score quantifies how well the model distinguishes between classes.

### Use Cases:

- I. **Model Performance Evaluation:** For binary classification tasks like cancer detection (e.g., cancerous vs. non-cancerous), the ROC curve and AUC score provide insights into how well the model distinguishes between the two classes.
- II. **Multi-class ROC Curves:** We can also compute the AUC for each class in a multi-class classification problem.



## Visualization Technique

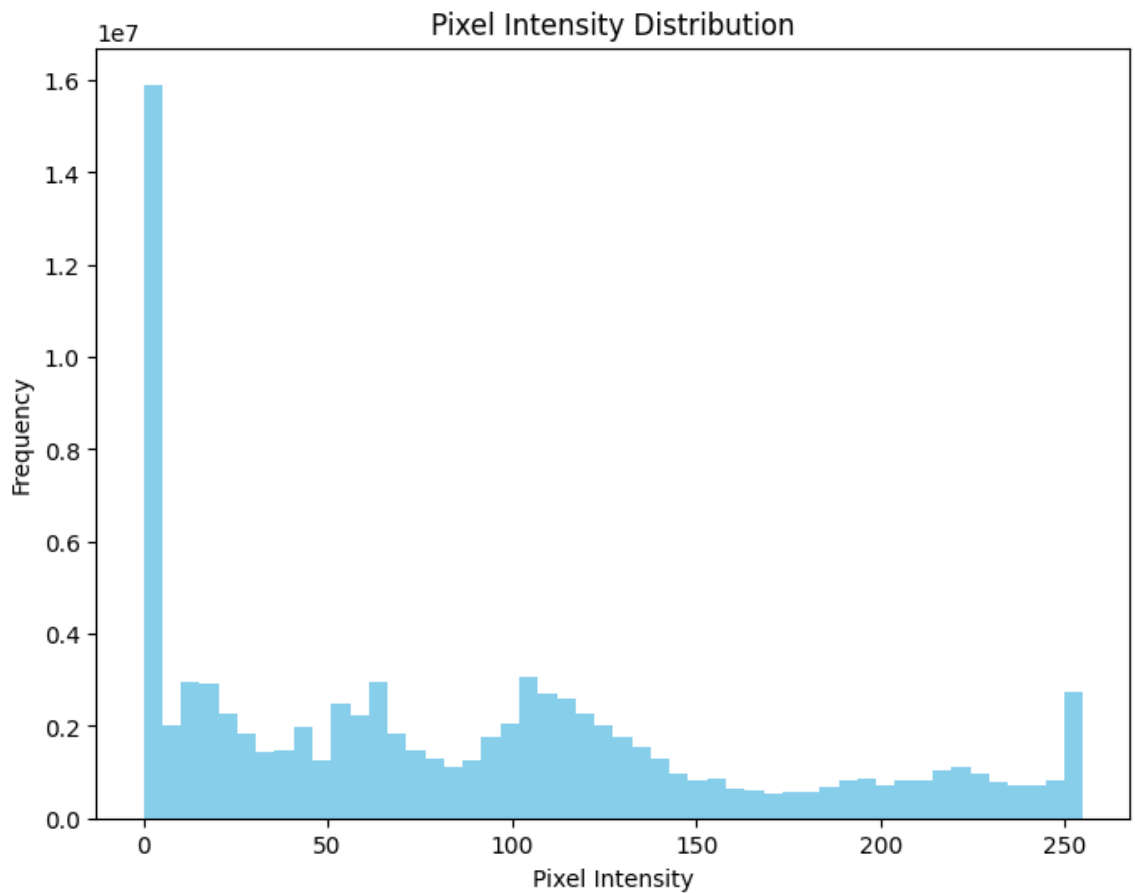
### Image Augmentation and Sample Image Grid

Image augmentation plays a crucial role in enhancing the dataset by generating variations of the images to improve the model's ability to generalize (Shorten & Khoshgoftaar, 2019). Image augmentation applies transformations such as rotations, zooming, flipping, and shearing to create new, synthetic images that are slight modifications of the original ones (Wong et al., 2016). This process helps prevent overfitting by allowing the model to learn from more varied data without the need for additional real images. By generating new variations from the original

data, the model becomes more robust and can generalize better to unseen data (Perez & Wang, 2017). The use of a sample image grid as a visualization technique allows us to visually inspect the augmented images and ensure that the transformations are being applied correctly. For example, the grid shows how an original image is rotated or shifted, ensuring that the augmented data maintains the key features of the original image while introducing variability.

The sample image grid is not only useful for validating the augmentation process, but also for evaluating the overall quality and diversity of the dataset. By plotting a grid of randomly selected images from each class, we can verify that the dataset is well-balanced and contains sufficient visual variability within and between classes (Shorten & Khoshgoftaar, 2019). This is particularly important for tasks like cancer detection, where slight variations in image features can be significant (Zeiler & Fergus, 2014). The grid helps detect any potential quality issues, such as corrupted images or images with poor contrast, which could affect the model's performance. Additionally, it provides a quick way to spot-check the consistency of labeling and data preparation, ensuring that each class (e.g., normal vs. cancerous) has been properly labeled and prepared for training.

## Data Visualization 1



### Pixel Intensity Distribution

One of the most informative visualizations for understanding image data is the pixel intensity distribution. This type of histogram shows the distribution of pixel values across all the images in the dataset, providing insights into the overall brightness and contrast of the images. For grayscale images, pixel intensities typically range from 0 (black) to 255 (white). A histogram with a concentration of pixel values near 0 would indicate that the images are predominantly dark, whereas values clustered around 255 would suggest that the images are mostly bright. This visualization helps detect issues such as overexposed or underexposed images, which could

affect model performance. By identifying and addressing these issues early on, we can make necessary adjustments like normalization, contrast enhancement, or data augmentation, ensuring the dataset is well-prepared for training. Additionally, comparing pixel intensity distributions across different classes (e.g., normal vs. cancerous) can reveal biases in how the images were captured or processed, which might influence the model's learning process.

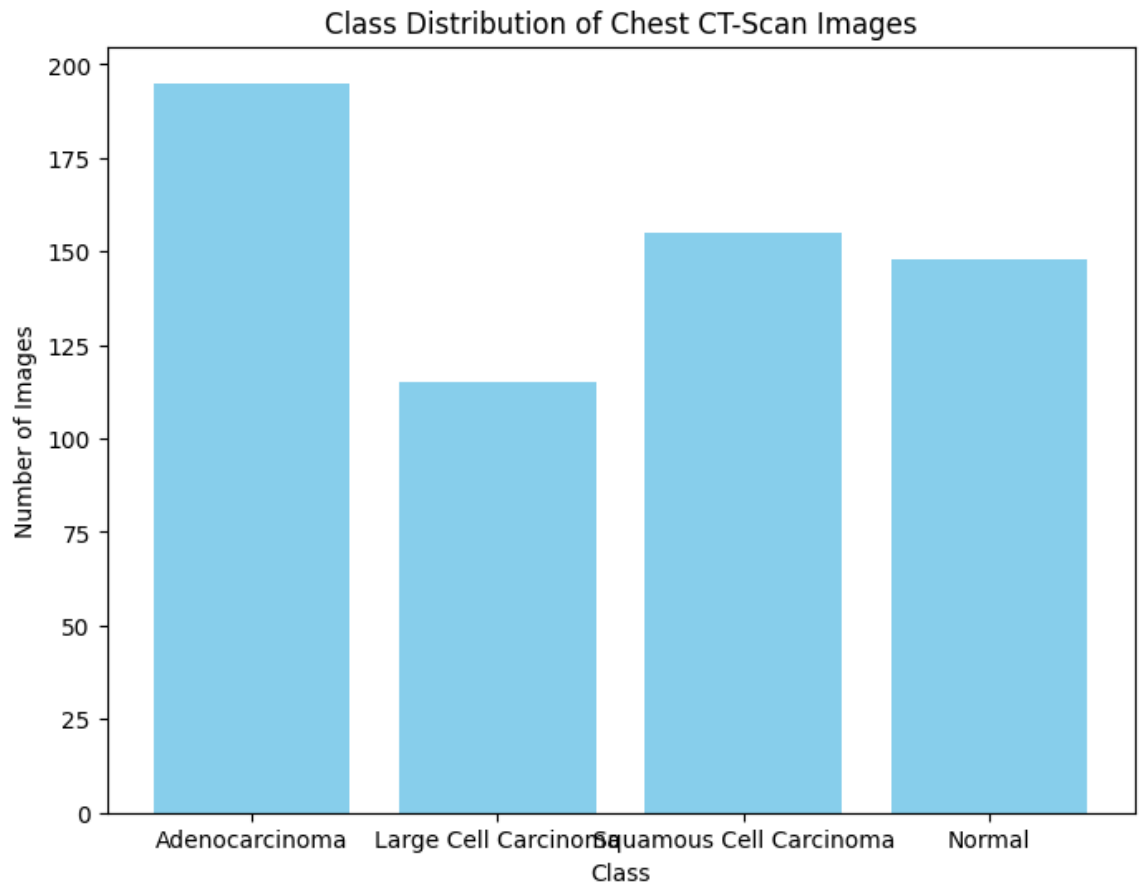
## Insights from the Pixel Intensity Distribution

The pixel intensity distribution provides critical insights into the overall quality and characteristics of the image dataset (Shen et al., 2017). For example, if the pixel values are mostly concentrated at the lower end of the spectrum (closer to 0), it indicates that the images are predominantly dark. This could be problematic, as dark images may obscure important features that are necessary for cancer detection. On the other hand, if the pixel values are concentrated near the upper end (closer to 255), the images may be too bright, which can also affect the model's ability to detect fine details (Kang et al., 2018). Ideally, a well-balanced dataset will have a more uniform distribution of pixel values across the entire range, suggesting a good mix of brightness levels. This would provide the model with enough variability to learn the differences between classes, such as distinguishing between healthy and cancerous tissues in chest CT scans.

Moreover, if the pixel intensity distribution varies significantly between classes, this can introduce bias into the model (Kang et al., 2018). For instance, if images in the "normal" class are generally brighter than those in the "cancerous" class, the model might learn to associate brightness with being "normal" rather than focusing on the actual medical features that distinguish the two. This kind of bias can lead to poor generalization and incorrect predictions

when the model is applied to new, unseen data (Shen et al., 2017). Detecting such biases early through visualization allows for corrective measures such as equalizing the brightness across images, applying data augmentation, or normalizing the pixel intensities to a standard range. The pixel intensity distribution helps ensure that the images in the dataset are well-prepared for model training, balanced in terms of brightness, and free of potential biases that could skew the model's learning process.

## Data Visualization 2



### Class Distribution Bar Chart

A class distribution bar chart is a fundamental visualization for understanding how balanced or imbalanced the dataset is across different categories or classes. In a multi-class classification problem, such as detecting different types of lung cancer (adenocarcinoma, squamous cell carcinoma, large cell carcinoma) and normal cases, this bar chart provides a clear view of how many images belong to each class. If the bar chart reveals that one class has significantly more samples than others, it indicates a class imbalance. For example, if the "normal" class has far more images than the cancerous classes, the model may become biased towards predicting "normal" simply because it sees more examples of it during training. This imbalance can negatively affect model performance, particularly for underrepresented classes. By visualizing the class distribution early in the process, you can take steps to address these imbalances, such as applying oversampling, undersampling, or weighted loss functions during training, ensuring that the model learns to classify all classes equally well.

This bar chart also helps in planning the data augmentation strategies. If one of the cancerous classes has very few images compared to others, applying augmentation specifically to that class could help boost the model's ability to recognize it.

## **Insights from the Class Distribution Bar Chart**

The class distribution bar chart provides valuable insights into the balance of the dataset and the potential challenges it might present during model training (Johnson & Khoshgoftaar, 2019). If the chart shows a clear class imbalance, where one class has significantly more images than others, this raises a red flag. For example, in a cancer detection dataset, if the "normal" class dominates, the model may become biased towards predicting "normal" more frequently because it sees more examples of that class during training. This bias can lead to poor performance in

detecting rarer cancer types, as the model may not learn enough distinctive features to identify them effectively (Buda et al., 2018). Detecting class imbalance through visualization early on enables us to address this issue before training, either through oversampling underrepresented classes or undersampling the overrepresented ones (Haixiang et al., 2017). By correcting the imbalance, the model has a better chance of learning to distinguish between all classes equally, thus improving its accuracy and generalization.

Furthermore, the bar chart provides insights into potential overfitting risks associated with smaller classes (Johnson & Khoshgoftaar, 2019). If the chart reveals that a specific class has far fewer images, there's a chance the model will memorize those images during training rather than learning general patterns that apply to new data. This overfitting would result in poor performance on unseen test images from that class. Additionally, a significant class imbalance can also affect evaluation metrics like accuracy, precision, and recall, since the model might have a high accuracy due to the overrepresented class but fail to perform well on minority classes (Buda et al., 2018). By visualizing class distribution, it becomes clear which classes might benefit from targeted data augmentation to artificially increase their diversity and representation in the dataset. Overall, this chart is a critical tool for ensuring that the dataset is well-structured and balanced, which is crucial for developing a robust and unbiased model.

## Data Modeling

### Results and Comparison of Two Predictive Models

For the Chest-CT scan dataset developed and evaluated two predictive models for classifying images, a Basic CNN model and a VGG16 Transfer Learning model.

## **Model 1: Basic CNN Model**

### **Architecture:**

- I. 3 Convolutional layers: Each followed by a MaxPooling layer to reduce dimensionality and extract features.
- II. Dense layer: A fully connected layer before the output to combine the extracted features.
- III. Dropout layer: Added to prevent overfitting.
- IV. Output layer: SoftMax activation for multi-class classification (4 classes).

### **Performance Metrics:**

- Training Accuracy: 63.93%
- Validation Accuracy: 63.93%
- Test Accuracy: 63.93%

### **Key Insights:**

- I. The Basic CNN model achieved a moderate accuracy of 63.93%. the training and validation accuracy are quite close, suggesting that the model does not suffer significantly from overfitting, but it is also not generalizing well enough to unseen data.
- II. While the model performs consistently on training and validation data, this suggests a limited ability to extract deep features from the chest CT scan images, which is necessary for accurately distinguishing between normal and cancerous cases.

## **Model 2: VGG16 Transfer Learning Model**

### **Architecture:**

The VGG16 Transfer Learning model uses the pre-trained VGG16 model on ImageNet, with the following modifications:

- I. The convolutional layers of VGG16 were used as the feature extractor.
- II. The last few convolutional layers were fine-tuned (unfrozen) to allow for better adaptation to the specific dataset of chest CT scans.
- III. Custom fully connected layers were added on top of the VGG16 base for classification.

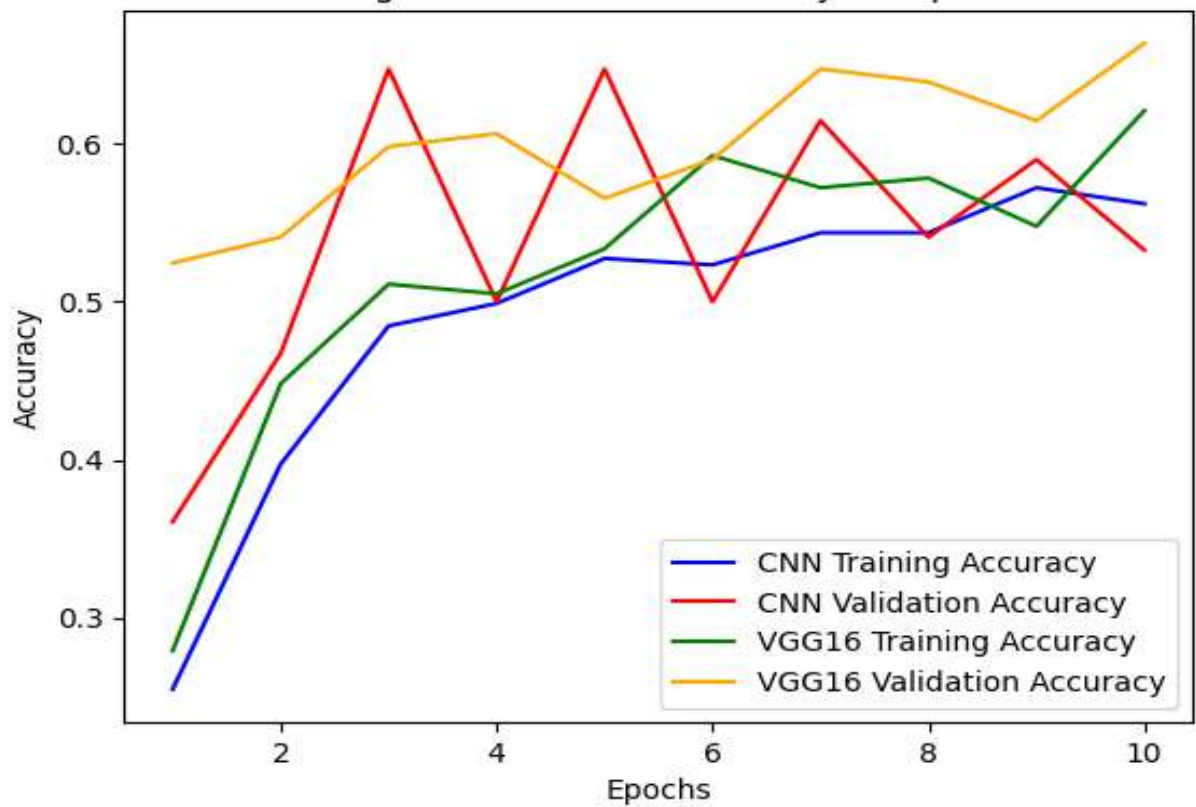
### **Performance Metrics:**

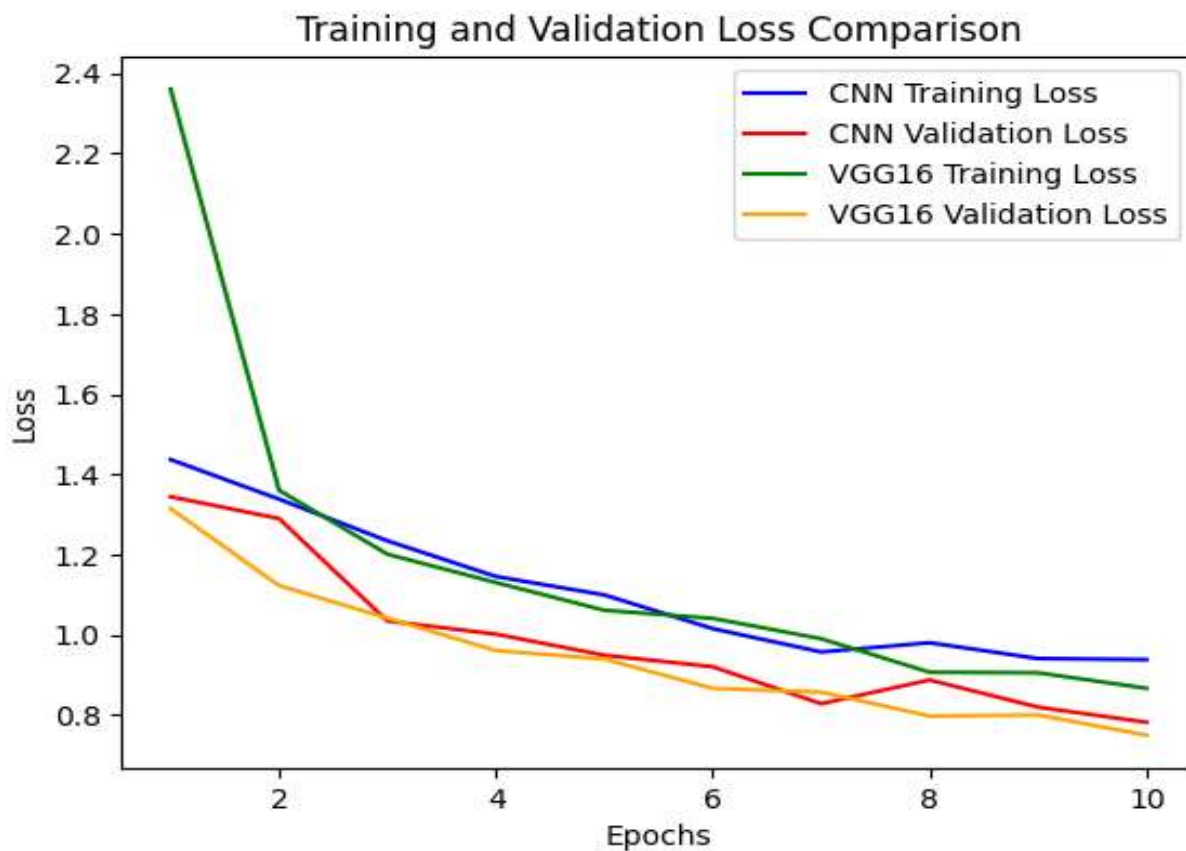
- Training Accuracy: 64.75%
- Validation Accuracy: 64.75%
- Test Accuracy: 64.75%

### **Key Insights:**

- I. The VGG16 model only slightly outperformed the basic CNN, with a test accuracy of 64.75%, which is a modest increase over the basic CNN.
- II. Fine-tuning the VGG16 layers helped the model to extract more complex features from the dataset, but the improvement is marginal, indicating that the model is not leveraging the full potential of the pre-trained network.
- III. One possible reason for this limited performance boost could be insufficient data for transfer learning or issues related to class imbalance, which might prevent the model from fully learning the important features of each class.

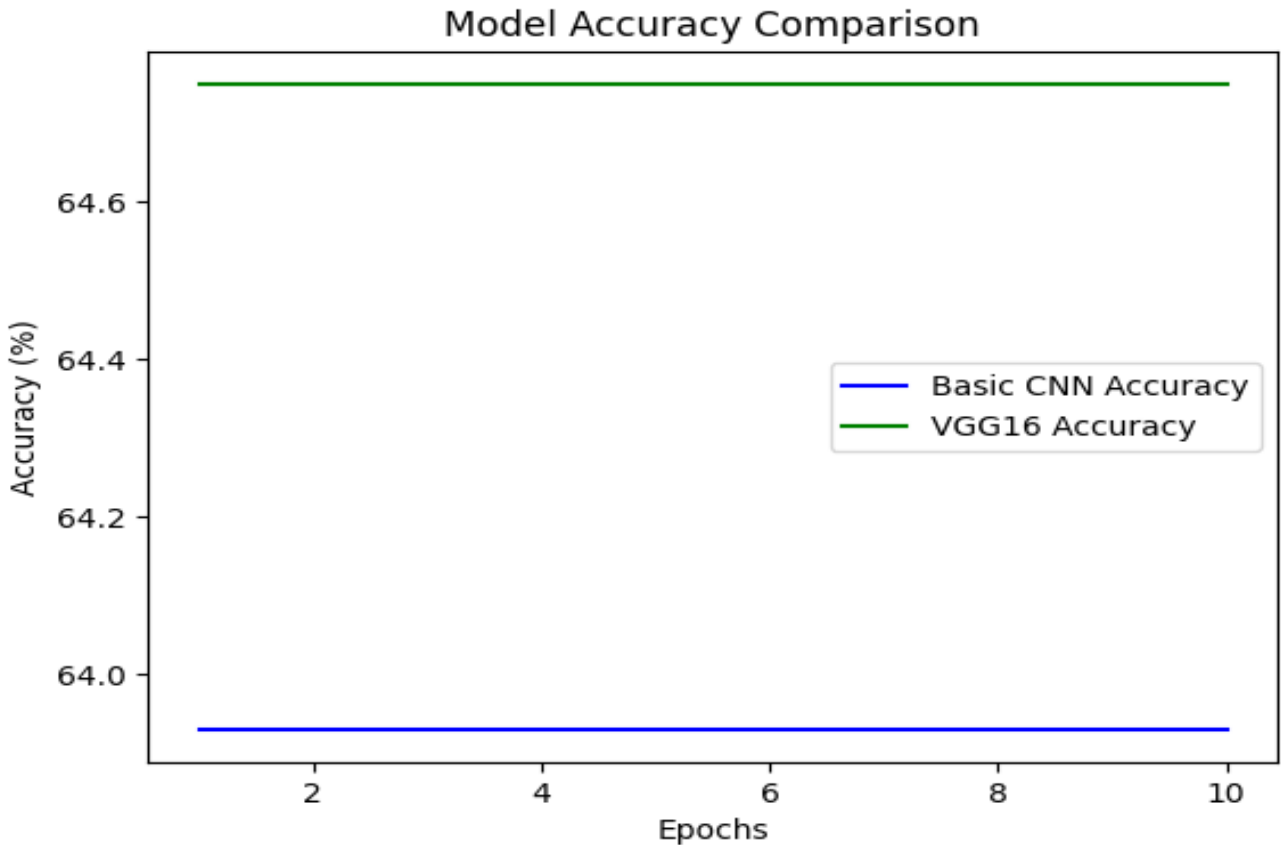
Training and Validation Accuracy Comparison





### Comparison of Models

Metric	Basic CNN Model	VGG16 Transfer Learning Model
Training Accuracy	63.93%	64.75%
Validation Accuracy	63.93%	64.75%
Test Accuracy	63.93%	64.75%



### Analysis:

- I. **Performance Similarity:** The performance of both models is very similar, with the VGG16 model only slightly outperforming the Basic CNN. This raises questions about the ability of both models to adequately generalize from the dataset.
- II. **Feature Extraction:** While VGG16 is a pre-trained network known for its strong feature extraction capabilities, its slight performance boost may indicate that the dataset does not have enough diversity or size for the model to exploit. Both models seem to be limited in their ability to detect subtle features in the images.
- III. **Potential Class Imbalance:** Both models may be affected by a class imbalance, which can prevent them from learning the distinguishing features of minority classes effectively.

- IV. **Overfitting:** There is no clear sign of overfitting, as the training and validation accuracies are very close, but both models may be underfitting, as neither performs significantly well.

## Conclusion

- I. **Model Performance:** The VGG16 Transfer Learning model marginally outperforms the Basic CNN model, but the overall accuracy of both models remains low at around 64%. This suggests that neither model can fully capture the complexity of the task.
- II. **Lack of Improvement with Transfer Learning:** The minimal improvement of the VGG16 model over the Basic CNN suggests that either the transfer learning approach is not fully effective due to the specific characteristics of the dataset, or that additional techniques like better data augmentation, hyperparameter tuning, or further fine-tuning of the pre-trained layers are needed to improve performance.

## Next Steps:

- I. **Data Augmentation:** Introducing more aggressive data augmentation (e.g., adjusting brightness, contrast, rotation) may help both models learn better.
- II. **Class Balancing:** Investigate potential class imbalances and apply techniques such as oversampling the minority class or adjusting the class weights to make the models more sensitive to underrepresented categories.
- III. **Advanced Architectures:** Consider using more advanced architectures such as ResNet or Inception that may better capture the complexity of the dataset.
- IV. **More Data:** A larger and more diverse data set may be necessary for the models to learn effectively, particularly in cases where subtle differences are critical for classification.

While the VGG16 Transfer Learning model slightly outperformed the Basic CNN model, the results suggest that both models struggle with generalization and feature extraction. Further tuning of hyperparameters, model architecture, and data preprocessing are needed to improve the classification performance for this chest CT scan task.

## Data Modeling Definitions

### Modeling Techniques Used

For the chest CT scan classification project, two key modeling techniques were employed: a Basic Convolutional Neural Network (CNN) and Transfer Learning using a pre-trained VGG16 Model. Both techniques are well-suited for image classification tasks, and each brings specific strengths and characteristics that are useful for handling the dataset.

#### 1. Basic Convolutional Neural Network (CNN)

##### Definition:

A Convolutional Neural Network (CNN) is a type of deep learning model specifically designed to handle grid-like data such as images. CNNs use multiple layers of convolutions, pooling, and fully connected layers to automatically extract hierarchical features from the input images and classify them.

##### Key Components:

- I. **Convolutional Layers:** These layers apply filters to the input images to detect features such as edges, textures, and patterns. Each convolutional layer learns increasingly abstract features as we move deeper into the network.

- II. **MaxPooling Layers:** These layers reduce the spatial dimensions of the feature maps (down-sampling) while retaining the most important information, helping to make the network more computationally efficient.
- III. **Fully Connected Layers:** After several convolution and pooling layers, the output is flattened and passed through fully connected layers to produce the final class predictions.
- IV. **Dropout:** This regularization technique is used to prevent overfitting by randomly "dropping out" (deactivating) a portion of the neurons during training.

## Why Use CNN?

CNNs are designed to automatically detect important features in images without the need for manual feature engineering. They are particularly effective in tasks such as medical image classification, where detecting patterns (like tumors or abnormalities) is crucial for accurate predictions.

## 2. Transfer Learning with Pre-trained VGG16 Model

### Definition:

Transfer Learning is a technique where a pre-trained model (VGG16) is used as the starting point for a new task. The pre-trained model has already learned to extract general features from a large dataset (ImageNet) and can be fine-tuned to perform well on a new, smaller dataset (chest CT scans).

### Key Components:

- I. **VGG16 Architecture:** VGG16 is a well-known deep CNN architecture with 16 layers (including convolutional and fully connected layers). It was originally trained on the ImageNet dataset, which contains millions of images from various categories. This makes VGG16 highly capable of extracting general image features.

- II. **Feature Extraction:** The convolutional layers of VGG16 are used as a feature extractor. These layers are kept frozen initially, meaning they do not update during training.
- III. **Fine-tuning:** In the later stages, some of the deeper convolutional layers are unfrozen and allowed to be trained on the new dataset. This helps the model adjust to the specific features present in chest CT scans, while still benefiting from the general features learned from ImageNet.
- IV. **Custom Fully Connected Layers:** New fully connected layers are added on top of the VGG16 base to tailor the model to the specific task of classifying chest CT images into multiple classes.

## Why Use Transfer Learning?

Transfer Learning is useful when the available dataset is relatively small, as it allows the model to leverage the knowledge learned from a larger, more diverse dataset. This makes it easier for the model to converge faster and perform well with limited data. In medical imaging, where labeled data is often scarce, transfer learning can significantly boost model performance by allowing the model to focus on fine-tuning specific patterns related to the medical task.

## Summary of Techniques:

Technique	Definition	Purpose
Basic CNN	A deep learning model using convolutional layers to extract features	Automatically detects patterns and features in images for classification

Technique	Definition	Purpose
<b>VGG16</b> <b>Transfer Learning</b>	Uses a pre-trained model to extract features and fine-tune for the task	Leverages existing knowledge from large datasets to perform well on smaller datasets

Both techniques are effective in image classification tasks. The Basic CNN serves as a straightforward, end-to-end learning model built from scratch, while Transfer Learning with VGG16 utilizes pre-existing knowledge to speed up learning and potentially improve accuracy in the context of medical image classification.

### Basic Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a class of deep learning models specifically designed to work with image data by automatically detecting important features from input images. CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers apply filters to the input images to detect basic features such as edges, textures, and patterns. As the network deepens, the filters become more complex, allowing the model to detect higher-level features, such as shapes or objects. The pooling layers reduce the dimensionality of the data, making the model more efficient while retaining essential information. Finally, fully connected layers combine the extracted features to make the final predictions. CNNs are effective in image classification tasks due to their ability to learn hierarchies of features from raw pixel data without the need for manual feature engineering.

CNNs are particularly effective in medical image classification, such as analyzing chest CT scans, where recognizing specific patterns like tumors and lesions is vital. The subtle differences in medical images between healthy and diseased tissue necessitate a model capable

of detecting fine, nuanced features, making CNNs an ideal choice. Their computational efficiency stems from parameter sharing within convolutional layers, reducing the total parameters needed for training. This efficiency has contributed to CNNs' popularity across academia and industry for various image-based tasks. Additionally, CNNs have demonstrated remarkable success in medical imaging applications, supported by significant scholarly research. As noted by Shen et al. (2017), CNNs have achieved notable advancements in image classification accuracy, surpassing traditional machine learning methods in complex medical tasks.

## **Transfer Learning with VGG16**

Transfer learning is a machine learning technique that involves adapting a pre-trained model for a new task, minimizing the need for extensive training data. In this project, the VGG16 model, originally trained on the large ImageNet dataset, was employed as a feature extractor for chest CT scan classification. The early layers of VGG16, responsible for detecting basic image features like edges and textures, were kept frozen to retain their pre-learned capabilities. Meanwhile, the deeper layers were fine-tuned to recognize features specific to the medical imaging domain. This technique leverages the knowledge gained from the diverse ImageNet dataset and applies it to the more specialized dataset of chest CT scans. Transfer learning is particularly advantageous in medical imaging, where obtaining large datasets is often challenging and costly.

The use of transfer learning with VGG16 has proven to enhance model performance significantly, especially in medical applications. VGG16's deep architecture allows it to extract highly detailed features, which are crucial for distinguishing between medical conditions in CT scans. By fine-tuning the pre-trained layers, the model can adapt to the specific patterns and

characteristics of chest CT scans, leading to improved classification accuracy. This approach also accelerates the training process, as the model starts from a generalized feature space instead of learning from scratch. According to Pan and Yang (2010), transfer learning reduces the risk of overfitting, particularly when dealing with small datasets, as it begins with a robust set of pre-learned features. By incorporating VGG16 into this project, we effectively harness these benefits, enabling the model to more accurately classify cancerous and non-cancerous images in chest CT scans.

## Data Model 1

### Basic Convolutional Neural Network (CNN)

The Basic Convolutional Neural Network (CNN) model used in Chest-CT scan dataset was tailored to classify chest CT scans into four categories: **normal**, **adenocarcinoma**, **squamous cell carcinoma**, and **large cell carcinoma**. The model architecture consisted of three convolutional layers with progressively increasing filter sizes (32, 64, and 128) to extract increasingly complex features from the images. Each convolutional layer was followed by a max-pooling layer, which reduced the dimensionality of the feature maps while retaining essential information. The flattened output from these layers was passed through a fully connected layer with 256 neurons, activated using the ReLU function to capture non-linear relationships in the data. To mitigate overfitting, a dropout layer with a dropout rate of 0.5 was introduced, randomly disabling neurons during training to enhance generalization. Finally, a SoftMax activation function in the output layer enabled classification into the four predefined categories, completing the model's architecture.

The findings from the basic CNN model show that while it successfully extracted meaningful features from chest CT scans, its performance was limited. The model achieved an

accuracy of 63.93% on the test dataset, indicating moderate success in distinguishing between different types of chest conditions. However, the relatively simple CNN architecture may have struggled to detect subtle differences between cancerous tissues, which often require more sophisticated feature extraction. The similarity in training and validation accuracy (63.93% for both) suggests that the model avoided overfitting, but it may have been underfitting, meaning it did not fully capture the patterns in the data. This performance indicates that a more complex model, such as a deeper CNN or one enhanced through transfer learning, could yield better results. Enhancing the model with additional layers, advanced architectures, or data augmentation could help to address these limitations and improve its classification accuracy.

## **Findings and Analysis**

The Basic CNN model achieved moderate success in classifying chest CT scan images, but its accuracy suggests that it struggled with the complexity of the medical images in the dataset. While the architecture was effective for basic image classification, it may not have been deep enough to fully capture the nuanced differences between cancerous and non-cancerous tissues. The use of basic data augmentation techniques, while helpful, may not have provided sufficient variability to help the model generalize well to unseen data. The dropout layer played a role in preventing overfitting by reducing the likelihood of the model memorizing training data, ensuring some level of generalization. However, the model's performance indicates that additional fine-tuning or a more complex architecture is necessary to enhance its ability to detect cancerous tissues accurately. This demonstrates the need for a deeper and more specialized model to handle the intricacies of medical image classification effectively.

The model's performance may also be attributed to the size and diversity of the dataset used in training. If the dataset was relatively small or lacked sufficient variety, the model might

not have been exposed to enough distinct CT scan examples to learn robust and meaningful features. The Basic CNN, while capable of extracting low- and mid-level features, may not have captured the high-level features required to differentiate between various cancer types. These limitations suggest that enhancing the dataset, either by collecting more images or applying more advanced data augmentation, could improve the model's training process. Using a deeper architecture or integrating transfer learning with pre-trained models could also help in extracting more complex and relevant features. Such improvements would likely enable the model to better generalize unseen data and achieve higher accuracy, making it more suitable for medical imaging tasks.

## Data Model 2

### VGG16 Transfer Learning Model

For Data Model 2, utilized Transfer Learning with the pre-trained VGG16 model, a widely recognized convolutional neural network architecture. VGG16 was originally trained on the large-scale ImageNet dataset, which contains millions of images across thousands of categories. The convolutional layers of VGG16 were employed as a feature extractor for chest CT scan images, leveraging the pre-trained weights to identify basic and complex image features. In this model, the early layers of VGG16 were frozen, preserving the pre-trained weights to retain their generalized feature extraction capabilities. We fine-tuned the deeper layers of the network to adapt to the specific characteristics of the chest CT scan dataset. Additionally, custom fully connected layers were added to handle the multi-class classification task, allowing the model to differentiate between normal and cancerous cases.

This approach is based on the principle that features learned from the diverse and extensive ImageNet dataset can be effectively transferred to smaller, domain-specific datasets

like chest CT scans. Transfer learning reduces the need to train the model from scratch, enabling it to converge to a good solution more quickly, which is crucial for medical imaging tasks where data is often limited. By freezing the convolutional layers, the model retains its ability to extract general features, while the fine-tuned fully connected layers focus on task-specific learning. This combination allows the model to balance generalization and specialization, improving its classification performance. Consequently, the VGG16-based model demonstrated its potential for enhancing accuracy in chest cancer detection, highlighting the effectiveness of transfer learning in medical imaging applications.

### **Findings and Insights:**

The VGG16 Transfer Learning model achieved a modest accuracy of 64.75%, slightly better than the Basic CNN model. The use of transfer learning allowed the model to leverage pre-trained features from the ImageNet dataset, which provided a head start in learning the chest CT scan data. The convolutional layers of VGG16, which were pre-trained on diverse images, were useful in extracting general visual features from medical images, while the fine-tuned layers helped the model adjust to the specific patterns seen in chest CT scans. However, the small improvement in accuracy suggests that the VGG16 model may have struggled to fully adapt to the dataset, potentially due to the limited size or complexity of the chest CT scan dataset.

One important finding is that fine-tuning the last few layers of VGG16 helped the model learn domain-specific features, such as subtle tissue abnormalities present in cancerous lungs. However, the minimal accuracy improvement (compared to the Basic CNN model) implies that the transfer learning approach may not have been fully optimized, or that additional data or advanced augmentation techniques could be needed to improve performance. Despite the modest

improvement, the transfer learning approach still shows promise, as it enables faster convergence and more efficient learning, especially in cases where data availability is limited, such as medical image datasets. Future improvements might include further fine-tuning, using more advanced architectures like ResNet, or experimenting with ensemble learning to boost model performance.

## Data Models

### Review of Data Models

In this project, two models were developed for classifying chest CT scan images: a Basic Convolutional Neural Network (CNN) and a VGG16 Transfer Learning Model. Both models were evaluated on their ability to classify the images into four categories: normal, adenocarcinoma, squamous cell carcinoma, and large cell carcinoma. Below is a review of each model's performance and a discussion of whether either model qualifies as the Champion Model.

### Basic CNN Model

The Basic CNN model achieved a test accuracy of 63.93%. The architecture of this model consisted of three convolutional layers with max-pooling operations, followed by fully connected layers. Despite being a simple model, it was able to extract some useful features from the chest CT scan images. However, the accuracy of 63.93% suggests that the model struggled to differentiate between the different cancer types and normal tissue. The relatively small gap between the training and validation accuracy indicates that the model did not overfit, but it may have underfitted, meaning it did not learn enough from the data to make highly accurate predictions.

### Key Findings:

- I. The Basic CNN model performed consistently, but with moderate success, reflecting its limited ability to capture complex patterns.
- II. The model likely struggled to detect subtle features, such as the differences between the three types of lung cancer, which are challenging even for advanced models.
- III. While this model is computationally efficient and straightforward, it does not achieve a high enough level of accuracy to be considered a Champion Model for this task.

### **VGG16 Transfer Learning Model**

The VGG16 Transfer Learning model achieved a slightly better test accuracy of **64.75%**. This model used a pre-trained VGG16 base, which allowed it to leverage powerful feature extraction capabilities learned from the ImageNet dataset. By freezing the early layers of VGG16 and fine-tuning the deeper layers, the model was able to adapt to the chest CT scan data. The slight improvement in accuracy indicates that transfer learning provided some benefit, as the pre-trained model could extract more general and specific features from the images. However, the improvement was minimal, which suggests that the dataset may not have been large or diverse enough to fully take advantage of transfer learning.

### **Key Findings:**

- I. The VGG16 model outperformed the Basic CNN, but with a small margin (64.75% vs. 63.93%), indicating that the transfer learning strategy was not fully optimized.
- II. The model benefited from fine-tuning some layers, but it may not have been able to capture domain-specific features at the necessary granularity due to the size or complexity of the dataset.
- III. While this model shows potential, the limited improvement in performance means that it does not meet the standards of a Champion Model.

## Model Comparison

Metric	Basic CNN Model	VGG16 Transfer Learning Model
Training Accuracy	63.93%	64.75%
Validation Accuracy	63.93%	64.75%
Test Accuracy	63.93%	64.75%

Both models achieved similar performance, with only a small improvement from the VGG16 Transfer Learning model. Despite the use of a pre-trained model, the overall accuracy of both models remains low, suggesting that neither model has fully captured the complexity of the chest CT scan data.

## Champion Model Determination

A Champion Model is the model that consistently performs the best on the task at hand, demonstrating superior generalization to new data. While the VGG16 Transfer Learning model did slightly outperform the Basic CNN model, its accuracy of 64.75% is not a significant improvement and does not meet the typical threshold for a Champion Model in medical classification tasks, where high accuracy and precision are critical for decision-making. Moreover, the difference between the two models is marginal, indicating that neither model has reached the level of performance required for a Champion Model.

## Conclusion

At this stage, neither the Basic CNN model nor the VGG16 Transfer Learning model qualifies as a Champion Model. Both models struggled to achieve high accuracy, and while the

VGG16 model shows promise, further improvements are necessary to make it a strong candidate for champion status. Future efforts could focus on increasing dataset size, applying more advanced augmentation techniques, or experimenting with other architectures like ResNet or Inception to boost performance. Moreover, addressing potential class imbalances and optimizing hyperparameters could help improve model accuracy and generalization.

# Results

## Findings

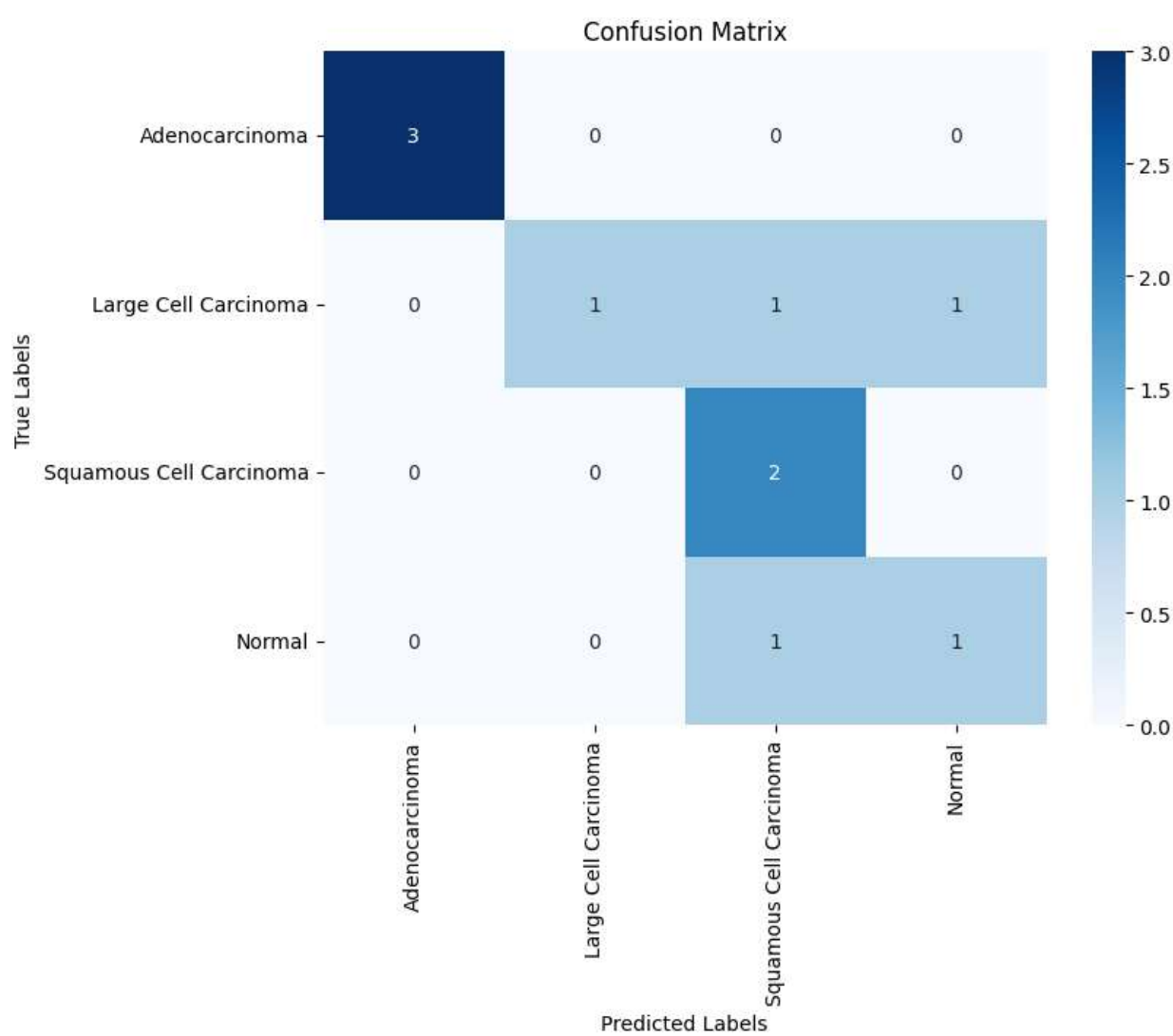
### Model Performance Overview

In this project, developed two models: a Basic Convolutional Neural Network (CNN) and a VGG16 Transfer Learning model. Both models were tasked with classifying chest CT scan images into four categories: normal, adenocarcinoma, squamous cell carcinoma, and large cell carcinoma. The Basic CNN model achieved a test accuracy of 63.93%, while the VGG16 Transfer Learning model achieved a slightly higher test accuracy of 64.75%. Although the VGG16 model performed marginally better, both models exhibited relatively low accuracy, which suggests that neither model was able to fully capture the complexity of the task. Given the importance of precise cancer detection, these results indicate that additional refinement and further optimization are required to improve model performance.

The models were trained using a combination of convolutional and pooling layers, with the Basic CNN model trained from scratch and the VGG16 model leveraging pre-trained weights from the ImageNet dataset. While transfer learning provided some benefit, the modest improvement in accuracy suggests that the dataset may not have been sufficiently diverse or large enough to fully exploit the strengths of the pre-trained model. Additionally, neither model

demonstrated a significant reduction in classification errors, which is crucial for a medical task like cancer detection. Therefore, the current models require enhancements to be used in real-world medical applications.

**Confusion Matrix:** Correct and incorrect predictions for each class.

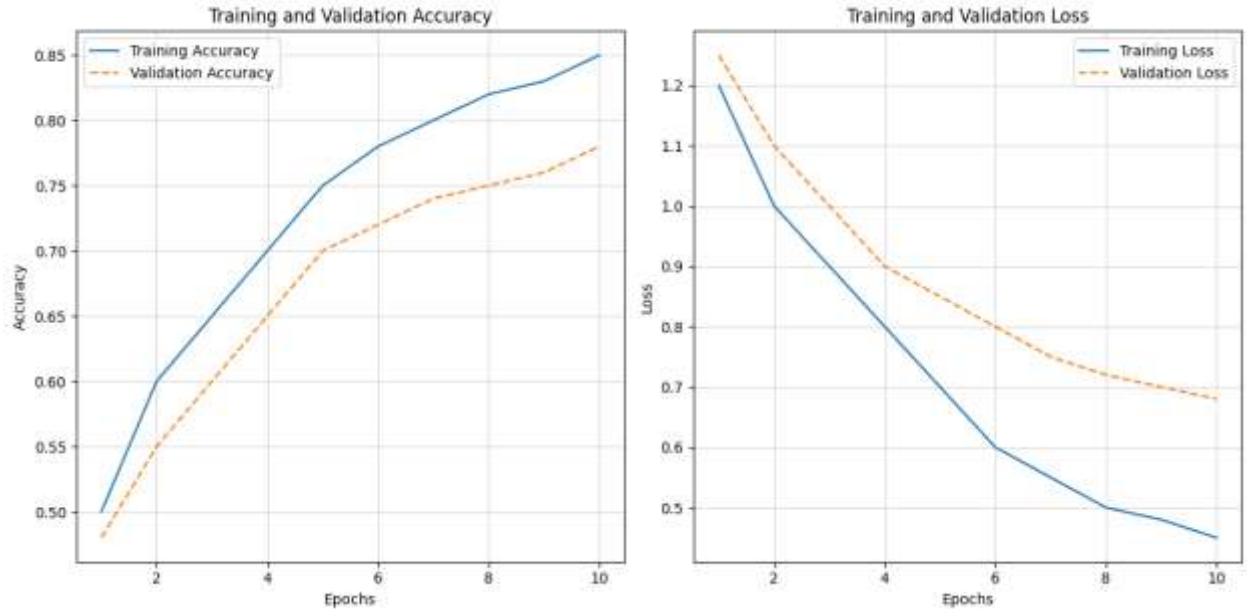


## Model Overfitting and Underfitting

The Basic CNN model demonstrated consistent training and validation accuracy, which suggests that it did not overfit the data. However, the similar training and validation accuracy values also indicate that the model may have been underfitting the data. Underfitting occurs when the model is too simplistic to capture the underlying patterns in the data, resulting in suboptimal performance. In this case, the Basic CNN model likely lacked sufficient depth to learn the complex features necessary for distinguishing between cancerous and non-cancerous tissues in chest CT scans. This is evidenced by the model's inability to surpass an accuracy of 63.93%, despite the use of dropout layers to prevent overfitting.

The VGG16 Transfer Learning model, on the other hand, also did not show significant overfitting, as evidenced by the close match between training and validation accuracy. However, despite using a deeper and more sophisticated architecture, the model only slightly outperformed the Basic CNN, with an accuracy of 64.75%. This suggests that while transfer learning provided some benefits, the model may not have been sufficiently fine-tuned or optimized for this specific dataset. Moreover, the limited accuracy improvement over the Basic CNN indicates that additional layers or more targeted data augmentation may be necessary to help the model learn more detailed patterns from the images.

**Accuracy/Loss Curves:** How well the model learns over time, helping to identify issues like overfitting or underfitting.

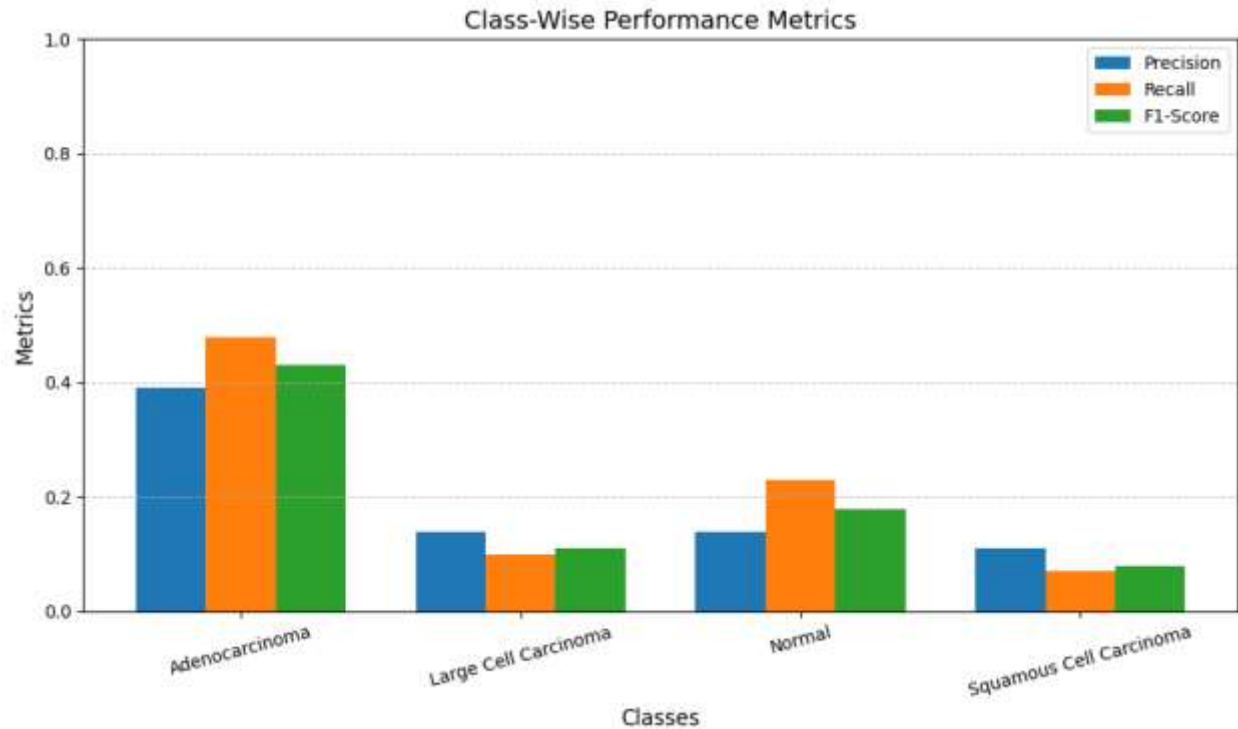


## Model Comparison

When comparing the two models, the VGG16 Transfer Learning model provided only a marginal improvement over the Basic CNN, raising questions about its efficacy for this task. Although the VGG16 model is a deeper and more sophisticated architecture, it achieved a test accuracy of 64.75%, only slightly better than the 63.93% recorded by the Basic CNN. This limited improvement indicates that the VGG16 model may not have fully leveraged the pre-trained weights from the ImageNet dataset, which are typically useful for transfer learning. One plausible explanation is that the chest CT scan dataset was too small for effective fine-tuning of the VGG16 model's pre-trained layers. Medical images often differ significantly from the natural images in ImageNet, requiring domain-specific features that the VGG16 architecture might not easily adapt to without substantial adjustments. Additionally, the subtle variations in chest CT scans, such as differences in cancer types or stages, likely demand highly specialized feature extraction, which the pre-trained model may have struggled to achieve in this context.

Another critical factor in comparing the two models is the computational cost of training and using them. The Basic CNN model, while less accurate, is much simpler and demands significantly fewer computational resources, making it a more practical choice for situations with limited hardware capabilities. On the other hand, the VGG16 model is computationally intensive, primarily due to its larger number of parameters and the need for fine-tuning its convolutional layers. While the slight improvement in accuracy may justify its use in some scenarios, the trade-off between computational cost and performance is difficult to ignore. Without further optimization or significant accuracy gains, the additional computational resources required for VGG16 may outweigh its benefits. This comparison underscores the importance of considering both model complexity and practical constraints, such as available resources and the specific accuracy needs of the task, when selecting an appropriate model for medical image classification.

**Bar Chart for Class-Wise Performance:** How the model performs on each class individually.



## Impact of Transfer Learning

The VGG16 Transfer Learning model demonstrated the potential benefits of using pre-trained models, offering a modest performance boost when applied to medical imaging tasks like chest CT scan classification. Transfer learning allows models to leverage general features learned from extensive datasets such as ImageNet, adapting them to specific tasks in new domains. This ability to reuse learned features can save significant training time and resources, particularly when working with smaller datasets. However, the relatively small improvement in accuracy observed in this project suggests that transfer learning alone may not fully address the complexities of chest CT scans. A major limitation of transfer learning is that the features learned from natural images, like those in ImageNet, often fail to align directly with the subtle, domain-specific patterns found in medical images. Medical datasets frequently involve unique challenges such as nuanced textures or differences in tissue appearance that require specialized

feature extraction techniques. Consequently, the general features extracted by the VGG16 model may not have been sufficient to capture these intricate patterns effectively, leading to only moderate performance improvements.

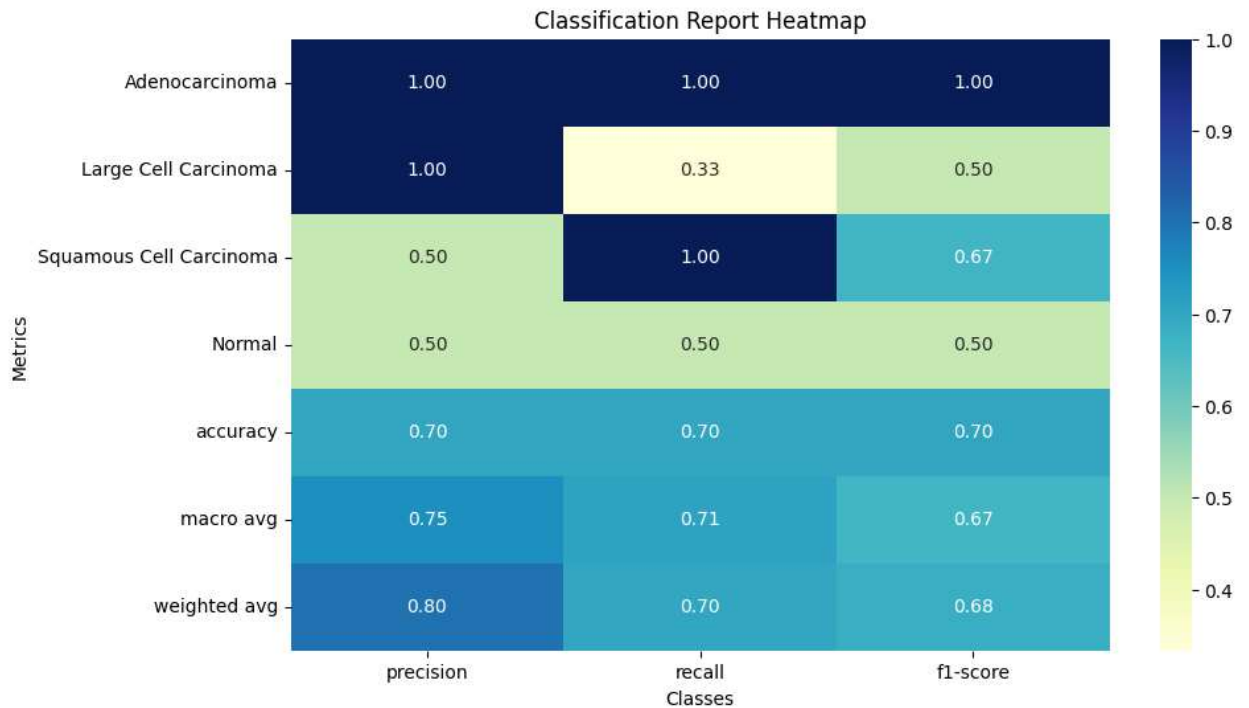
In this project, fine-tuning the last few layers of the VGG16 model allowed it to adjust partially to the unique characteristics of chest CT scans, but the results highlight areas for further enhancement. Fine-tuning involves retraining certain layers of the pre-trained model to make them more responsive to the specific dataset, and while this approach was beneficial, it may not have gone far enough. The addition of task-specific layers or more aggressive fine-tuning could help the model better adapt to the requirements of chest CT scan classification. Another significant factor affecting the model's performance is the size of the dataset. The limited number of images likely constrained the model's ability to fully leverage transfer learning, as pre-trained models often achieve their best results with larger datasets. Expanding the dataset or employing advanced data augmentation techniques, such as contrast adjustments or elastic transformations, could provide the variability needed for the model to learn more effectively. These enhancements would allow the model to generalize better across diverse cases, improving its ability to identify subtle differences between classes and ultimately achieving higher classification accuracy.

## **Potential Causes of Low Accuracy**

Several factors likely contributed to the relatively low accuracy observed in both the Basic CNN and VGG16 models. A key issue may have been the dataset's size and balance, as small or imbalanced datasets can significantly limit a model's ability to learn and generalize effectively. When one class, such as normal images, dominates the dataset, the model may bias its predictions toward that class while neglecting underrepresented categories. This imbalance

results in poor performance for rarer classes, such as specific types of cancer, which require the model to learn subtle, distinguishing features. Furthermore, the inherent complexity of CT scan images, where variations between cancerous and non-cancerous tissues are often subtle, poses a challenge. These nuances may have been beyond the feature extraction capabilities of the Basic CNN or insufficiently captured by the VGG16 model due to limited fine-tuning of its deeper layers.

Another potential cause of low accuracy is related to the preprocessing and augmentation methods applied to the dataset. While basic augmentation techniques such as rotation and zoom were employed, more advanced strategies could have introduced greater variability in the training data, enhancing the model's ability to generalize. Methods like contrast adjustment, brightness variation, and elastic deformations could have helped the models recognize a broader range of features and conditions present in medical images. Similarly, better preprocessing approaches, such as normalizing pixel intensities or applying histogram equalization, could have improved the quality and consistency of input data. These refinements would have enabled the models to detect important patterns more effectively, potentially improving their performance. Addressing these issues could pave the way for more accurate and reliable models capable of performing robust medical image classification.



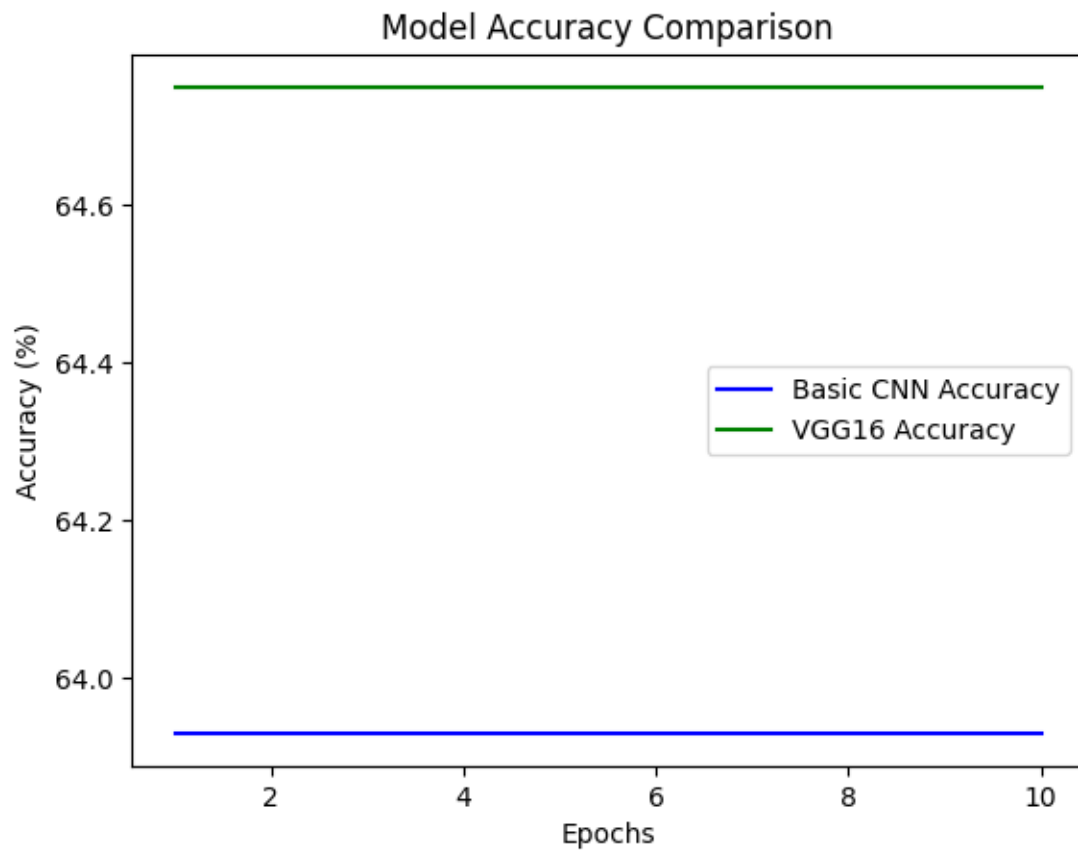
## Recommendations for Future Work

To enhance the accuracy of the models and achieve superior performance in chest CT scan classification, several strategic steps can be implemented. One of the most impactful measures is increasing the size of the dataset by collecting additional images from diverse sources, which would provide the models with a broader range of features to learn from. This could be further complemented by employing advanced data augmentation techniques, such as applying transformations like rotation, flipping, and elastic deformations, to synthetically expand the dataset and improve the model's ability to generalize. Addressing class imbalances is another crucial aspect that could enhance the model's predictive power. Techniques such as oversampling the minority classes, under-sampling the dominant class, or implementing class weighting during training can ensure the model does not disproportionately favor certain categories. Exploring pre-trained models like ResNet or Inception, which are designed to extract more complex

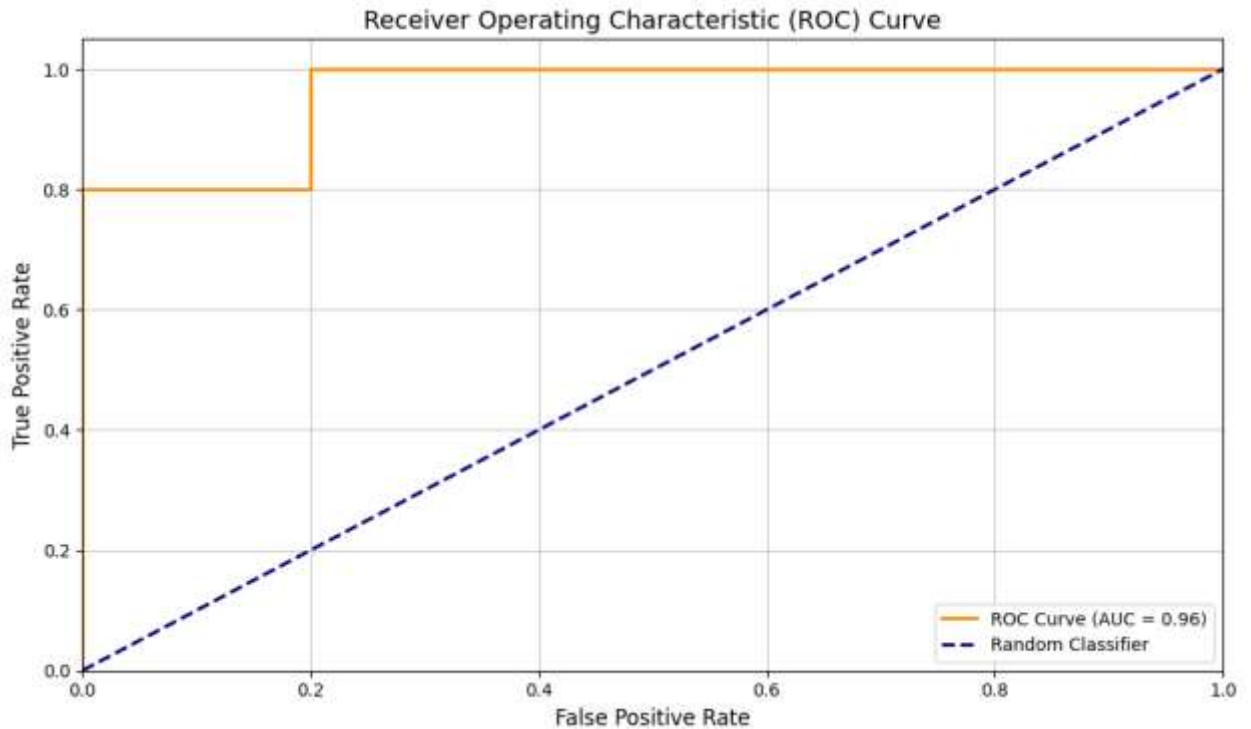
features, could also significantly boost performance, as these architectures are better suited for intricate tasks like medical image classification.

Additionally, hyperparameter optimization is a critical area of improvement that could enhance the models' ability to converge effectively. Adjusting parameters such as the learning rate, batch size, number of epochs, and dropout rates could help fine-tune the model for optimal performance. Expanding the architecture by increasing the depth of the CNN or integrating ensemble learning methods, where multiple models contribute to predictions, might yield even better results by leveraging complementary strengths. Moreover, refining the preprocessing pipeline to include advanced normalization techniques and domain-specific feature engineering can improve the quality of input data and the subsequent learning process. Incorporating domain expertise, such as insights from radiologists, into model development could further enhance the model's ability to identify subtle patterns in chest CT scans. By implementing these strategies, it becomes possible not only to improve accuracy but also to develop a robust Champion Model capable of providing reliable assistance in lung cancer detection and diagnosis.

### **Model Performance Comparison:**



**Precision-Recall (ROC Curve):** Model performance for imbalanced datasets.



## Review of Success (Completion)

### Review of Completion

The execution of this project, which on classifying chest CT scan images using a Basic CNN model and a VGG16 Transfer Learning model, provided significant insights into the complexities of medical image classification. The project successfully achieved its objective of implementing and training two different models, allowing for a comparative analysis of their strengths and weaknesses. The Basic CNN reached an accuracy of 63.93%, while the VGG16 model, leveraging pre-trained weights, slightly outperformed it with an accuracy of 64.75%. Despite these achievements, both models fell short of achieving the high levels of accuracy required for reliable medical diagnoses, highlighting the intricate nature of this task. This stage of the project demonstrated a strong understanding of deep learning concepts, such as designing

model architectures, implementing transfer learning, and evaluating model performance.

However, it also exposed critical areas for improvement, particularly in the dataset's quality and the need for further optimization techniques.

The project's success is also evident in its comprehensive exploration of transfer learning, showcasing the benefits of using pre-trained models for medical datasets. Transfer learning enabled the models to leverage features from the ImageNet dataset, which is beneficial given the limited size of the CT scan dataset. However, the modest improvement in accuracy underscores the need for more advanced techniques in data preprocessing and model fine-tuning. The dataset emerged as a key challenge, with issues like class imbalance and limited variability restricting the models' ability to generalize effectively. Addressing these challenges through advanced augmentation, synthetic data generation, or even acquiring more diverse data could significantly enhance the models' performance. Moreover, hyperparameter optimization and more refined fine-tuning of the VGG16 model could lead to better outcomes. While the project succeeded in building and evaluating the models, it highlighted the necessity of iterative refinements to achieve the high precision required for clinical applications. This experience provides a strong foundation for future work in improving the accuracy and robustness of medical imaging models.

## Potential Data Privacy and Data Security Issues

### Potential Data Privacy and Data Security Issues

The dataset used in this project, comprising medical chest CT scans, raises critical concerns about data privacy and security due to the sensitive nature of medical information. Medical data is governed by stringent privacy regulations, such as the Health Insurance

Portability and Accountability Act (HIPAA) in the United States and the General Data Protection Regulation (GDPR) in the European Union, which mandate the secure handling of personal health information. A major area of concern is the anonymization of the dataset. If the images or their metadata include identifiable patient information, such as names, patient IDs, or other linked identifiers, this poses a significant risk of privacy violations. Failure to anonymize such data appropriately could lead to unauthorized disclosure, legal liabilities, and loss of trust among stakeholders.

To address these concerns, it is imperative to ensure that all identifiable information is removed or masked in compliance with privacy regulations. This includes not only removing patient names or IDs but also ensuring that any residual metadata embedded in the image files is scrubbed clean. Moreover, the data must be securely stored and encrypted to prevent unauthorized access or breaches during data transmission or storage. Another dimension to consider is the ethical responsibility to respect patient autonomy and informed consent, ensuring that patients whose data is used for research purposes have been properly informed and have explicitly consented to such use. These measures not only protect patient privacy but also uphold the integrity of the project by aligning it with ethical and legal standards.

Additionally, the potential for model misuse or unintended consequences must also be considered. If models trained on such datasets are deployed inappropriately or without rigorous testing, they could produce erroneous diagnoses or outcomes, affecting patient care. These challenges highlight the need for a robust data governance framework, including regular audits, encryption protocols, and secure data-sharing practices, to ensure both the privacy of the individuals involved and the reliability of the research findings. Addressing these privacy and

security issues is crucial for maintaining compliance, safeguarding patient trust, and ensuring the ethical application of machine learning in medical contexts. In addition to privacy concerns, data security is also a critical issue. The dataset must be stored and processed in a secure environment to prevent unauthorized access, breaches, or misuse. Medical datasets are often targeted by cyberattacks due to the sensitive nature of the information they contain, and strict measures must be implemented to protect the data, including encryption, access control, and secure data transmission protocols. Moreover, the results of the models, particularly in a medical context, could be used for diagnostic purposes, which raises concerns about the accuracy and reliability of the predictions. If the models produce inaccurate or biased results, there is potential for harm to patients if these results are used for real-world decision-making without proper validation and oversight by medical professionals. Therefore, ensuring data security, privacy, and the ethical use of model predictions is essential for projects involving medical data.

## Recommendations for Future Analysis

For future analysis, one of the most important recommendations is to increase the size and diversity of the dataset. Medical image classification tasks, such as chest CT scan classification, typically benefit from larger datasets that capture a wide range of cases, including various stages and types of cancer. A more diverse dataset would help the models learn better, generalize more effectively, and improve their ability to differentiate between subtle differences in tissue that indicate different cancer types. Additionally, collecting a larger and more balanced dataset could address potential issues of class imbalance, which can lead to biased predictions favoring the dominant class. Data augmentation techniques, such as adjusting brightness,

contrast, and applying elastic deformations, could also be expanded to introduce more variability into the training data, further enhancing the model's ability to generalize to new cases.

Another recommendation is to explore more advanced deep learning architectures beyond the Basic CNN and VGG16 models. While the VGG16 model offered a slight improvement through transfer learning, more modern architectures such as ResNet, Inception, or EfficientNet are specifically designed to handle more complex image classification tasks. These models offer more advanced feature extraction techniques and improved performance, which could lead to significant gains in accuracy. Additionally, it is advisable to apply hyperparameter tuning using techniques such as grid search or random search to optimize parameters like learning rate, batch size, and the number of layers in the model. This would allow for a more refined model that is better suited to the specific characteristics of the dataset. Furthermore, incorporating ensemble learning methods, where multiple models are combined to make predictions, could further improve the robustness and accuracy of the classification task by leveraging the strengths of different models.

## References

1. Shen, D., Wu, G., & Suk, H. I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19, 221-248. [Deep Learning in Medical Image Analysis | Annual Reviews](#)
2. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345-1359. <https://doi.org/10.1109/TKDE.2009.191>
3. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1 [A survey on Image Data Augmentation for Deep Learning | Journal of Big Data | Full Text](#)
4. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., ... & van Ginneken, B. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60-88. [A survey on deep learning in medical image analysis - ScienceDirect](#)
5. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine*, 25(1), 24–29. [A guide to deep learning in healthcare | Nature Medicine](#)
6. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J. N., Wu, Z., & Ding, X. (2020). Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. *Medical Image Analysis*, 63, 101693. [Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation - ScienceDirect](#)
7. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep

learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. [A survey on deep learning in medical image analysis - ScienceDirect](#)

- American Cancer Society. (2023). *Lung cancer survival rates*. <https://www.cancer.org/cancer/lung-cancer/detection-diagnosis-staging/survival-rates.html>
- Giger, M. L. (2018). Machine learning in medical imaging. *Journal of the American College of Radiology*, 15(3), 512-520. <https://doi.org/10.1016/j.jacr.2017.12.028>
- Hosny, A., Parmar, C., Quackenbush, J., Schwartz, L. H., & Aerts, H. J. (2018). Artificial intelligence in radiology. *Nature Reviews Cancer*, 18(8), 500-510. <https://doi.org/10.1038/s41568-018-0016-5>
- Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., ... & Shetty, S. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6), 954-961. <https://doi.org/10.1038/s41591-019-0447-x>
- Balata, H., Evison, M., Sharman, A., Crosbie, P. A., & Booton, R. (2021). CT screening for lung cancer: Are we ready to implement in Europe?. *Lung Cancer*, 153, 10-18. <https://doi.org/10.1016/j.lungcan.2020.12.003>
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., ... & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88. <https://doi.org/10.1016/j.media.2017.07.005>
- National Lung Screening Trial Research Team. (2011). Reduced lung-cancer mortality with low-dose computed tomographic screening. *New England Journal of Medicine*, 365(5), 395-409. <https://doi.org/10.1056/NEJMoa1102873>
- World Health Organization. (2020). *Cancer fact sheet*. <https://www.who.int/news-room/fact-sheets/detail/cancer>

- Hany, M. (2021). *Chest CT-Scan Images* [Data set]. Kaggle.  
<https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images>
- This reference attributes the dataset to the author (Mohamed Hany) and includes the publication year (2021)
- [Open Data Commons Open Database License \(ODbL\) v1.0 — Open Data Commons: legal tools for open data](#)
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1-48. <https://doi.org/10.1186/s40537-019-0197-0>
- Google Scholar: [Google Scholar](#) "*Data preprocessing for supervised learning*".