Explainable 3D Convolutional Neural Networks for Enhancing Early Lung Cancer Detection

A Thesis Presented

By

Fikadu Ashebir Haile

To

The Faculty of Informatics

Of

St. Mary's University

In Partial Fulfillment of the Requirements
For the Degree of Master of Science

In

Computer Science

January 2025

ACCEPTANCE

Explainable 3D Convolutional Neural Networks (CNNs) for Enhancing Early Lung Cancer Detection

By

Fikadu Ashebir Haile

Submitted to St. Mary's University, the Faculty of Informatics, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

_____

Internal Examiner

_____

External Examiner

_____

Dean, Faculty of Informatics

January 2025

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other university, and all sources of materials used for the thesis work have been duly acknowledged.

Fikadu Ashebir Haile
Full Name of Student

_____
Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

Million Meshesha (Phd)
Full Name of Advisor

_____million_____
Signature

Addis Ababa

Ethiopia

{January, 2025}

ACKNOWLEDGEMENT

First and foremost, I extend my deepest gratitude to Jesus Christ and His Mother, Saint Mary, for their boundless grace and blessings that have guided me through this journey.

I am profoundly indebted to my advisor, Million Meshesha (Ph.D.), for the invaluable guidance, mentorship, and unwavering support during the research process. Your insightful feedback and constructive advice have been pivotal in shaping the direction and quality of my work. You have not only provided academic guidance but have also inspired me to strive for excellence, and for that, I am deeply grateful.

I would also like to extend my sincere gratitude to the data providers (LUNA22) whose contributions have been instrumental in the success of this research. Your support in granting access to vital resources and information has greatly enriched the scope of this study.

To Saint Mary's University, my alma mater, I am profoundly thankful for providing me with the opportunity to pursue my master's degree. My heartfelt appreciation also goes to my instructors, Especially Alembante (Ph.D.), whose expertise, encouragement, and dedication have been critical in shaping this work. Your guidance has been a source of great motivation and inspiration throughout my academic journey.

To my beloved family, I dedicate this work to the loving memory of my dear father, Ashebir Haile, the very foundation upon which I have built my dreams. Abaye, every step I take and every milestone I achieve is because of the values, strength, and love you instilled in me. This is all for you, and I know you are looking down on me with pride. You are my guiding star, and I carry your legacy with honor and love.

To my mother, Fana Fita (Fani), my unwavering pillar of strength, thank you for your endless sacrifices, unwavering belief in me, and unconditional support. Your encouragement has been my anchor, and I am forever indebted to you for your love and guidance.

I also want to express my deepest gratitude to my beloved wife, Bethlehem Zeleke, my partner, my rock, and my greatest cheerleader. Thank you for your endless love, patience, and unwavering support. You have stood by my side in every situation, celebrating my triumphs and uplifting me in my struggles. Your encouragement has been my greatest motivation, and your faith in me has been my guiding light. This journey would not have been possible without you, and I am forever grateful for your love and presence in my life.

My heartfelt appreciation goes to my darling sister, Segen Ashebir, and her wonderful family. Segen, your love, kindness, and constant support have been a blessing beyond words. In every challenge, you have stood by me, lifting me with your unwavering faith and encouragement, and I am profoundly grateful for everything you have done for me. I also extend my deepest gratitude to my dear brother, Mulugeta Ashebir, and his family. Your encouragement and steadfast support have been a pillar of strength throughout this challenging yet rewarding endeavor.

Last but certainly not least, to my friends Daniel Bekele, Michael Endalu, Dawit Desalegn, and Besufikad Nigussie, your companionship and unwavering support have been a true blessing. Your words of encouragement and camaraderie have brought light to this journey and have been a source of great strength.

To all who have touched my life in this journey, thank you from the depths of my heart your contributions have been a gift, and this achievement is as much yours as it is mine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AUC: Area under the Curve

CAD: Computer-Aided Diagnosis

CAT: Computed Axial Tomography

CBAM: Convolutional Block Attention Module

CIP: Cancer Imaging Program

CNN: Convolutional Neural Networks

CT: Computed Tomography

DICOM: Digital Imaging and Communication in Medicine

DL: Deep Learning

FN: False Negative

FP: False Positive

FNLCR: Frederick National Laboratory for Cancer Research

Grad-CAM: Gradient-weighted Class Activation Mapping

HU: Hounsfield Unit

IML: Interpretable Machine Learning

LIDC-IDRI: Lung Imaging Database Consortium-Image Database Resource Initiative

LIME: Local Interpretable Model-agnostic Explanations

LDCT: Low-Dose Computed Tomography

LIME: Local Interpretable Model-Agnostic Explanations

LRP: Layer-Wise Relevance Propagation

LUNA16: Lung Nodule Analysis 2016

LUNA22-ISMI: Lung Nodule Analysis 2022 - Intelligent Systems in Medical

NIfTI: Neuroimaging Informatics Technology Initiative

NSCLC: Non-Small Cell Lung Cancer

RLU: Rectified Linear Unit

RN-PDMP: Residual Neural and Partial Derivative Multilayer Perceptron

ROC-AUC: Receiver Operating Characteristic Curve's - Area under the Curve

XAI: Explainable Artificial Intelligence

ABSTRACT

Lung cancer is a major cause of cancer-related deaths, often due to late diagnosis. Early detection is vital for better outcomes and lower mortality rates. Traditional methods like chest X-rays are not sensitive enough to spot small, early-stage nodules, highlighting the need for advanced imaging and classification techniques to identify malignant nodules early on. This study investigates the application of interpretable deep learning models for classifying lung nodules in CT scans as benign or malignant, aiding in the early detection of lung cancer. We leverage 3D Convolutional Neural Networks (3D CNNs) trained on the LUNA22 ISMI dataset, comprising 1,176 lung nodules with a collection of lung nodule annotations from anonymized CT scans. In the preprocessing stage, we standardized pixel intensity values and applied augmentation techniques such as rotation, scaling, and flipping to enhance the diversity of the training data. These steps were implemented using Python packages, including SciPy for augmentation and NumPy for array operations. We compared four 3D CNN models with different architectures, including a baseline model, a 3D AlexNet-based model, a proposed 3D CNN model, and a 3D CNN model integrated with the Convolutional Block Attention Module (CBAM). The CBAM module enhances feature extraction by applying attention mechanisms to the most informative features. Our proposed 3D CNN with CBAM achieved the highest performance, with an accuracy of 94.06%, AUC of 98.84%, and F1-Score of 95.56%. To ensure model transparency and facilitate clinical adoption, we employed 3D Grad-CAM to generate visual explanations of the model's predictions. This technique provides insights into the regions of the lung nodule that most influence the classification decision. Despite these promising results, the study has several limitations. The dataset used, although robust, is limited in size and diversity, potentially impacting generalizability to broader populations. Additionally, our models were trained and evaluated under controlled conditions, which may not fully replicate real-world clinical environments. Future work should focus on validating these models in larger, more diverse datasets and deploying them in prospective clinical studies to assess their practical impact. Integration with existing diagnostic workflows and evaluating cost-effectiveness are also critical steps toward clinical adoption.

***Keywords**: 3D Convolutional Neural Network (3D CNN), Explainable AI (XAI), 3D Gradient Class Activation Maps (3D Grad-CAM), Convolutional Block Attention Module (CBAM), LUNA22 ISMI*

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Cancer is a general term that refers to many diseases that arise in any body organ. Malignant neoplasm and neoplasm are two other terms. One distinguishing feature of cancer is the rapid growth of tumor cells that exceed their normal size and can spread to nearby parts of the body and metastasize to secondary organs; metastasis is this latter phenomenon. Cancer cells tend to travel to the rest of the body, where they begin growing and replacing normal tissue. This is called metastasis, which occurs when the cancer cells travel into our body's blood or lymphatic system. But when cells from a cancer like breast cancer move to a different organ like the liver, the cancer is still breast cancer and not liver cancer. The primary cause of cancer mortality is widespread metastasis. (National Cancer Institute, 2021).

The top cause of mortality globally is cancer, which resulted in nearly 10 million deaths in 2020, nearly one in every six. Breast, lung, colon, rectum, and prostate cancers are the most common. (WHO, 2022). Based on the World Health Organization (WHO), in May 2020, (new cancer cases) breast (2.26 million cases), lung (2.21 million cases), colon and rectum (1.93 million cases), prostate (1.41 million cases), skin (non-melanoma) (1.20 million cases), and stomach (1.09 million cases). And in 2020, the top five causes of cancer death were lung (1.80 million), colon and rectum (916 000), liver (830 000), stomach (769 000), and breast (685 000 deaths) (Ferlay et al., 2022).

In the following two decades, the cancer burden will be raised by nearly 60% again, and it will challenge healthcare systems, people, and communities even further. In the year 2040, about 30 million new cancer diagnoses are expected all over the world, with increases expected to occur most in low- and middle-income nations (American Cancer Society, 2007).

Most cancers start as tumors. Some cancers, like leukemia, do not form tumors. These cancer cells invade the blood and blood-forming organs and move into other tissues where they grow. Remember that not all tumors are cancer. A benign (non-cancer) tumor does not spread to other parts of the body and, with very rare exceptions, is not life-threatening. (National Cancer Institute, 2021).

Cancers are unique diseases, for example, lung cancer and breast cancer are very dissimilar. They grow at a different rate and are handled differently. For this reason, cancer patients need treatment that is aimed at their particular type of cancer. (National Cancer Institute, 2021)

The International Agency for Research on Cancer (IARC), a WHO department, has a classification of carcinogenic agents. IARC estimates lung cancer was the world's most frequent cancer until 2020 when breast cancers slightly exceeded lung cancers. Lung cancer has been and still remains the global leader in cancer-related deaths. As is evident above, in the year 2020, almost 1.8 million lost their lives due to lung cancer, almost twice the number of deaths due to cancer caused by colorectal cancer, which is the second most frequent cause of cancer death. The major causative factor for developing lung cancer is stated by IARC to be tobacco smoking. Other frequent causes are air pollution outdoors and indoors, diesel engine exhaust, welding fume, and asbestos.

1.1.1 Lung Cancer

Lung cancer primarily develops in the lungs and is generally divided into two main types: Small Cell Lung Cancer (SCLC) and Non-Small Cell Lung Cancer (NSCLC). In some rare cases, lung cancer may display features of both SCLC and NSCLC, which is referred to as combined small cell/large cell carcinoma (American Cancer Society, 2007). Figure 1.1 illustrates the various parts of the lung.

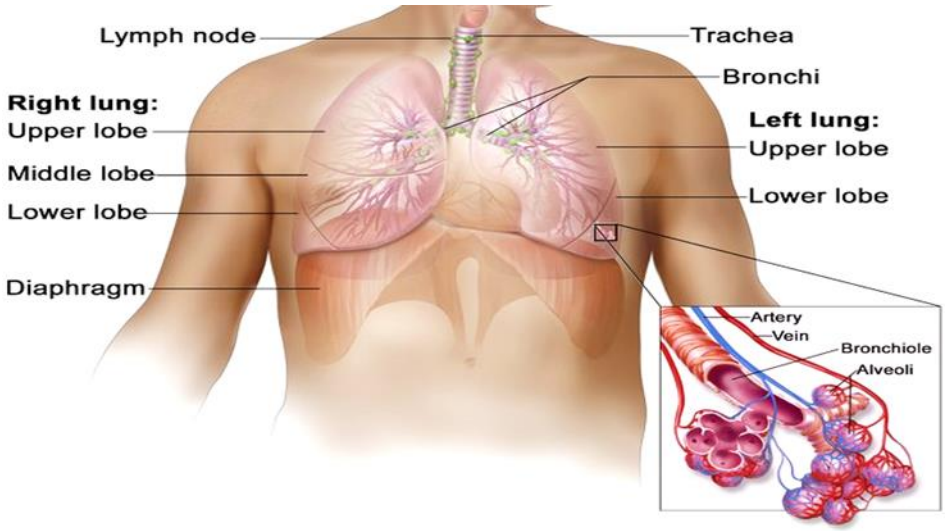


Figure 1.1 is an illustration of the respiratory system’s anatomy (https://www.cancer.gov/types/lung/patient/lung-prevention-pdq).

Lung cancer is a type of cancer that starts in the lungs. There are two major types of lung cancer: Small Cell Lung Cancer (SCLC) and Non-Small Cell Lung Cancer (NSCLC). A third less common type of lung cancer is called Carcinoid.

There are two forms of SCLC: Small Cell Carcinoma and Mixed Small Cell/Large Cell Cancer or Combined Small Cell Lung Cancer. Small cell lung cancer is classified by the types of cells in the cancer and how the cells appear when viewed with a microscope. Small-cell lung cancer is almost always associated with cigarette smoking. Small-cell lung cancer is usually treated with chemotherapy. (National Cancer Institute, 2021).

NSCLC is more common. It occurs in about 80 percent of lung cancer. This type of cancer develops and spreads to the rest of the body more slowly than small-cell lung cancer. There are three different types of NSCLC. The first is Adenocarcinoma: - which is a form of non-small cell lung cancer that most often occurs in the outer part of the lung. It occurs in epithelial tissue cells, which form the lining of the body surfaces and cavities and glands. The second one is Squamous cell carcinoma: A form of non-small cell lung cancer most typically found in the middle part of the lung along an air tube (bronchus). In addition, there exists large cell carcinoma: Non-small cell lung cancer which might occur in any part of the lung and tends to develop and spread faster compared to Squamous cell carcinoma or Adenocarcinoma. (National Cancer Institute, 2021). Carcinoid refers to a type of slow-growing tumor that originates from neuroendocrine cells, which are specialized cells found throughout the body and involved in hormone production. These tumors are part of a broader category called neuroendocrine tumors (NETs). While carcinoids can develop in various locations, they are most commonly found in the gastrointestinal (GI) tract, lungs, or, less frequently, in other organs like the pancreas or ovaries.

Lung Nodules: A lung nodule (or mass) is an abnormally small area that can sometimes be found on a CT chest scan. They are done for many reasons, such as part of lung cancer screening or as a screening of the lungs when you have a symptom. The majority of lung nodules found on a CT scan are not cancerous. They are more often caused by past infections, scar tissue, or other causes. Tests often must verify that a nodule is not cancer, though. In addition, LDCT Scan can also show the size, shape, and location of any lung cancers and can find enlarged lymph nodes that might have cancer that has spread.

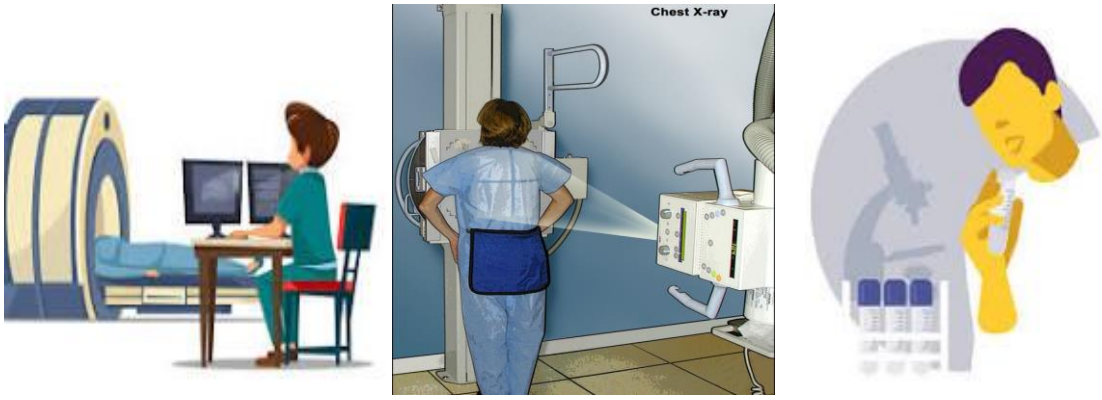
1.1.2 Lung Cancer Screening

The prognosis for most cancers, including cancer of the lungs, is much better if the illness can be detected and treated early. Lung cancer is detectable at an early stage by the use of screening methods, which involve the application of diagnostic tools or tests to determine the illness in individuals who have not yet started developing any symptoms. However, it must be stated that most lung cancers are typically diagnosed after symptoms arise. Therefore, a referral for a screening test is not always an indication that a doctor suspects cancer. A lung cancer diagnosis is established by examining a specimen of lung tissue in the lab (National Cancer Institute, 2021).

The three most frequently used screening tests for the early identification of lung cancer are Low-dose computed tomography, chest X-ray, and Sputum cytology (National Cancer Institute et al., 2021). Low-dose computed tomography (LDCT) or low-dose CT scan is an imaging procedure that captures a series of highly accurate images of body tissues employing a very low-level radiating x-ray machine, which is connected to a computer. In this procedure, three-dimensional pictures of organs and tissues are created by capturing x-rays from many angles. LDCT is recommended to be applied as a screening test for adults at risk of lung cancer, particularly those with a history of smoking and increasing age, since it can identify early symptoms of the disease easily. (National Cancer Institute, 2021).

A chest radiograph, or a chest X-ray, is among the most prevalent imaging studies applied that employs an intense beam of radiation to form images of bones and organs contained within the thoracic cavity. These include such vital structures as the lungs, heart, vessels, air passages, and spinal and rib bones. The process is not too painful and takes only a few seconds, making chest X-rays a useful resource for initial assessment and regular monitoring of several chest-related disorders.

Sputum cytology is a laboratory test in which cells that have been extracted from sputum, 'the mucus secreted by the lungs,' are examined under a microscope. The purpose of this test is to find abnormal cells, specifically those which can be an indicator of lung cancer. Figure 1.2 depicts examples of lung cancer screening methods.



LDCT-Scans Chest X-rays Sputum cytology.

Figure 2.1 Examples of Lung cancer screening methods
https://www.google.com/ (different LDCT scans and chest X-rays and sputum cytology).

Also, there are Positron Emission Tomography (PET Scans) and Magnetic Resonance Imaging (MRI Scans) (American Cancer Society., 2007). Regular chest X-rays have also been researched as a lung cancer screening test for individuals at higher risk, but the results have not indicated that the majority of patients benefit from the test, and thus it is not suggested for lung cancer screening. Studies using LDCT scans in individuals at greater risk of lung cancer have established that, in contrast to chest X-rays, yearly LDCT scans to screen individuals at high risk of lung cancer can preserve lives. Getting yearly lung cancer screening prior to any symptoms develop reduces these individuals' risk of dying from lung cancer (Mayekar, Pattewar, Patil, & Dhruv, 2022).

The final goal of lung cancer screening and detection methods is the identification of lung nodules, or small abnormal locations that are detectable on CT scans. The majority of lung nodules are benign and may be due to conditions such as scar tissue or a history of infections, but further evaluation is usually required to exclude malignancy. Low-dose computed tomography (LDCT) scans play an important role here since they not only detect lung nodules but also provide vital information regarding enlarged lymph nodes, which can be indicative of cancer spread, and information on tumor characteristics, including size, shape, and location. All this information is very important in terms of proper diagnosis and treatment planning. By effectively identifying lung nodules and offering precise information on potential tumors, LDCT scans play a significant role in the early detection of lung cancer and result in improved patient outcomes.

In this study, an attempt is made to apply computer vision and deep learning to lung cancer detection.

1.1.3 Research Background

Artificial intelligence (AI) can enable computers to be able to see and understand their own visual environment in a discipline known as computer vision (CV). Computer vision, through specific application of deep learning, utilizes artificial neural networks as a method of "learning" from large data sets. These networks are modeled after the decision-making mechanisms of the brain, thus enabling them to analyze images and recognize specific details, making them extremely useful tools for identifying lung cancer at an early stage.

The transformative potential of artificial intelligence, in particular deep learning-based computer vision, is vast across many domains, including medicine, where it can possibly enhance diagnostic precision and patient results.

Though AI won't replace healthcare workers, it's transforming the field with personalized care, outcome prediction, and better diagnostics. Analysis of real-time data can assist in disease treatment, cancer risk analysis, and even drug recommendations (Panesar, 2019)

As AI systems increase in complexity, as represented by Deep Neural Networks, it becomes more difficult to understand the reasoning behind their decisions. While earlier generations of AI systems are easier to interpret, modern AI models are increasingly viewed as more opaque (Barredo Arrieta et al., 2020).

Deep Learning (DL) models, such as Deep Neural Networks (DNNs), have proven to be very effective due to their massive size and advanced learning techniques. However, this complexity, with millions of parameters and hundreds of layers, makes DNNs difficult to understand, often referred to as "black box" models (Barredo Arrieta et al., 2020).

To effectively implement machine learning (ML) and deep learning (DL) models, particularly in the healthcare industry, it is essential to address certain questions: “Are we aware of the factors that influence the model's outputs?” and “Do we understand the scenarios in which the model performs well and those in which it does not?” A new subfield of artificial intelligence, known as Explainable AI (XAI), focuses on addressing these inquiries and transforming black box models into more transparent ones. This field

encompasses a wide range of applications across all areas of AI. Figure 1.3 illustrates the breadth of XAI's scope.

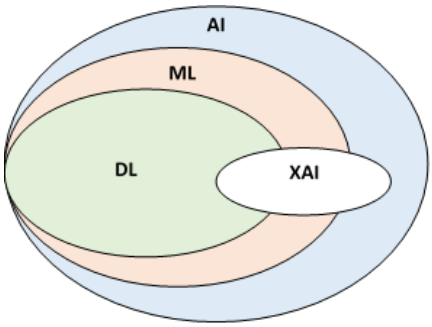


Figure 2.2 Scope of XAI (Conor O'Sullivan, n.d.).

The goal of Explainable AI (XAI), also referred to as interpretable machine learning (IML), is to develop machine learning models that are understandable to humans. This area of study encompasses both theoretical research and a variety of tools and methods designed for this purpose. It includes strategies for deciphering black box models as well as modeling techniques aimed at producing models that are easier to interpret. As noted by Conor O'Sullivan, "XAI can be viewed as both the process of interpreting models and enhancing their interpretability."

Computer vision (CV) empowers machines to analyze and interpret visual information from their environment. By leveraging deep learning techniques, particularly neural networks, computer vision systems can "learn" from extensive datasets to identify patterns and features in images. These systems emulate the decision-making processes of the human brain, enabling them to excel in tasks such as image analysis, object recognition, and feature extraction.

In healthcare, the integration of computer vision with deep learning holds transformative potential, especially for enhancing diagnostic accuracy and improving patient outcomes. For instance, the ability of deep learning models to process vast quantities of medical imaging data can significantly advance early disease detection. One prominent application is in the early detection of lung cancer, where computer vision enables the precise classification of lung nodules as benign or malignant. By analyzing CT scan data and identifying critical features, these systems bridge the gap left by traditional diagnostic tools like chest X-rays, which often fail to detect small, early-stage nodules.

When paired with explainable AI (XAI) techniques, computer vision can ensure transparency and trustworthiness in its applications. It can provide healthcare professionals with insights into model predictions and facilitate adoption in clinical settings.

1.2 Motivation of the Study

Lung cancer, which affects both men and women, is the foremost cause of cancer-related fatalities worldwide. According to recent data from the International Agency for Research on Cancer (IARC), approximately 19 million new cancer cases and 9 million deaths due to cancer were recorded globally in 2022 (WHO, 2022). Cancer continues to be one of the leading causes of death around the world, with lung cancer being the most common type. Figure 1.4 presents the incidence rates for both sexes in panels (a) and (b), while panels (c) and (d) illustrate the mortality rates (IARC, 2022).

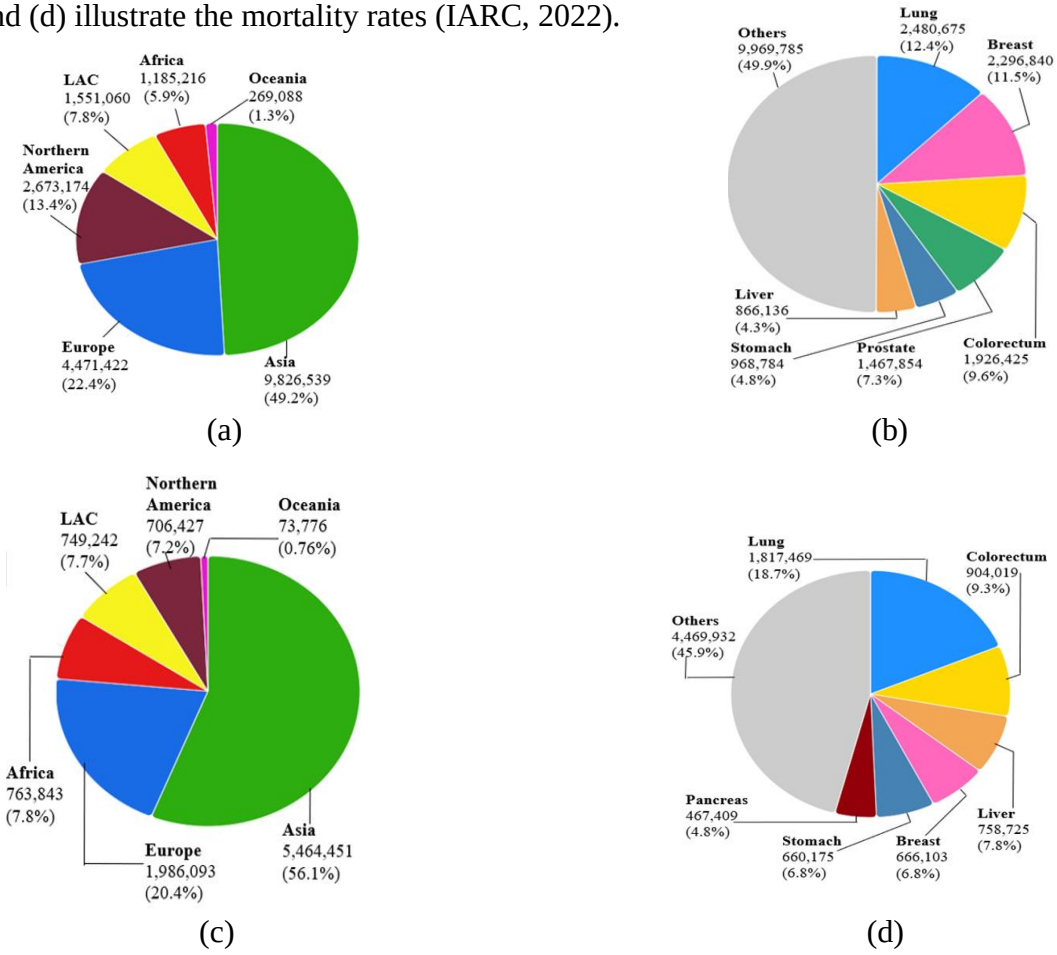


Figure 2.3 DataViz for Cancer Incidence and Mortality (IARC, 2022)

Many countries expect a rise in lung cancer cases and deaths by 2050, making it a major global health issue. As the nature of lung cancer changes, it's crucial to reallocate resources and improve prevention strategies to reduce the global burden of lung cancer in the future

(Luo et al., 2023). Figure 1.5 illustrates the projected number of new cases and deaths from 2022 to 2050, categorized by gender and age group, for cancers of the trachea, bronchus, and lung (Ferlay J et al., 2024) (IARC, 2024).

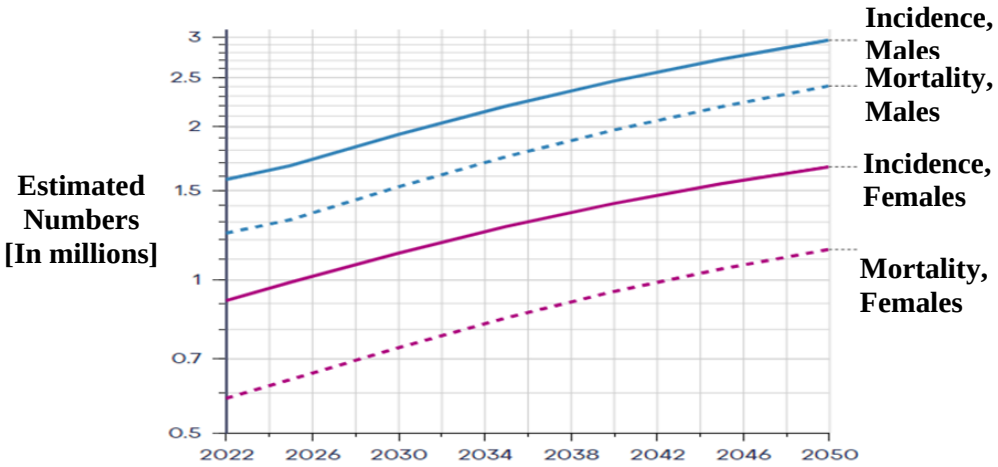


Figure 2.4 Projected number of new cases and mortality for trachea, bronchus, and lung cancers.

Even though treatments are available, many lung cancer cases are diagnosed at advanced stages. Early detection through screening programs for high-risk individuals is key to improving survival rates. Low-dose CT scans have been shown to significantly reduce lung cancer mortality in both current and former heavy smokers. Annual screening for three years is more effective than chest X-rays for early detection.

While LDCT scans are effective in detecting early-stage lung cancer, they also carry certain risks. One significant concern is the potential for false positives, where an individual is incorrectly identified as having cancer despite being cancer-free. Such errors can lead to unnecessary and invasive procedures. These issues often arise due to mistakes made by radiologists during the screening process. Over the past three decades, the workload for radiologists has increased substantially (Markotić et al., 2021), leading to burnout as they face overwhelming amounts of critical data with limited time to analyze it.

The data captured by medical imaging equipment has detailed information and has been used for AI models as AI is data-driven programming. If used wisely with advanced techniques in the ML and DL, it can assist the radiologists by minimizing the risks related to screening of lung cancer at an early stage.

As our models become more complex, they become opaque to the domain experts, in our case, to the medical doctors. Using XAI techniques, we can make our AI models more interpretable and transparent to clinicians.

1.3 Statement of the Problem

Lung cancer's high mortality rate is largely due to late-stage diagnosis when the disease is often beyond effective treatment. Early detection of small, localized nodules significantly improves survival rates, but traditional methods like chest X-rays lack the sensitivity to identify subtle abnormalities. Low-dose CT (LDCT) scans have proven more effective for early detection; however, they require highly skilled radiologists, creating challenges in managing the increased workload and data volume.

Deep learning models have shown promise in lung cancer diagnosis by identifying patterns in large datasets that are difficult for humans to detect. Despite their success, these models face challenges in clinical integration due to limited interpretability. Their "black box" nature makes it difficult for medical professionals to understand the reasoning behind predictions, raising concerns about transparency, patient safety, and trust in AI-driven tools. Studies such as (Shen et al., 2017), (Esteva et al., 2017), and (Arrieta et al., 2020) have highlighted the potential of deep learning in cancer detection but emphasize the need for explainable models.

While models like those in (Essaf et al., 2020) have demonstrated high accuracy in lung cancer detection, the lack of interpretability undermines their clinical utility. To ensure reliability and trustworthiness, it is crucial to enhance model transparency and Explainability, enabling healthcare providers to understand and validate AI predictions. This research aims to develop a deep learning-based early lung cancer detection model that integrates interpretability, bridging the gap between AI capabilities and clinical applicability.

The reliance on 2D imaging in early lung cancer detection using Explainable Convolutional Neural Networks (CNNs) presents significant challenges, creating a critical gap that limits diagnostic accuracy and model interpretability. 2D images fail to capture the full spatial context of lung anatomy, resulting in fragmented representations, loss of critical spatial relationships, and reduced effectiveness in detecting small or irregular nodules. These limitations also hinder Explainable AI (XAI) methods, such as Grad-CAM and Integrated Gradients, as they cannot generate comprehensive visualizations or meaningful insights with 2D inputs. In contrast, 3D imaging provides a holistic view of the lungs, enabling the model

to analyze nodule characteristics across multiple dimensions, improve spatial understanding, and enhance the precision of Explainability techniques through detailed 3D heatmaps and visual overlays. The gap between these approaches is evident in the inability of 2D-based models to meet the demands of early and reliable lung cancer detection, leading to higher false negative rates and reduced clinician trust in AI systems. Bridging this gap by adopting 3D imaging allows for more accurate predictions, robust interpretability, and better alignment with clinical practices, ultimately improving early detection and fostering greater trust in AI-driven healthcare solutions.

1.4 Research Questions

The research questions formulated to guide this work to come up with an optimal model with interpretability are the following:

RQ1: Which Deep learning techniques are performing best for early screening and detection of lung cancer using CT scan images?

RQ2: What is the performance of the proposed model for early-stage lung cancer detection?

RQ3: How do we integrate the explainable AI (XAI) method for interpretability of the detection results of the model?

1.5 Objective of the Study

1.5.1 General Objective

The general objective of this study is to build deep learning models for early lung cancer screening and detection from 3D images with interpretability of the prediction results using explainable AI (XAI).

1.5.2 Specific Objectives

This study is conducted to achieve the following specific objectives.

- To prepare CT scan lung cancer medical image data for AI model building.
- To select suitable deep learning models for experimentation.
- To develop a 3D CNNs model using lung cancer CT scan raw medical image data.
- Optimize deep learning models that incorporate explainability and interpretability to support decision-making processes in lung cancer diagnosis.

- To evaluate the performance of the proposed deep learning model

1.6 Significance of the Study

Lung cancer is a significant public health issue, and early detection is crucial for better patient outcomes. Deep learning, especially Convolutional Neural Networks (CNNs), has become a powerful tool for analyzing medical images like CT scans and identifying lung nodules. Here is how this research can aid radiologists in early lung cancer detection:

3D convolutional neural networks (CNNs) can be trained to automatically analyze CT scans and identify potential lung nodules, offering substantial support to radiologists during the screening process. This enables radiologists to concentrate on evaluating suspicious nodules and making accurate diagnoses. Deep learning (DL) models have demonstrated exceptional performance in detecting and classifying lung nodules, often surpassing human accuracy in certain scenarios. Such advancements enhance early lung cancer detection while reducing false positives, where patients are incorrectly diagnosed with the disease. By decreasing unnecessary treatments, 3D CNNs help optimize the allocation of healthcare resources and improve the patient experience. Additionally, these models process large volumes of data rapidly and efficiently, allowing radiologists to assess more scans in less time. This leads to improved patient throughput and shorter waiting times for true cancer-positive diagnoses.

Our research aims to address the "black box" issue by developing explainable deep-learning models for lung cancer diagnosis. By using explainable AI techniques, we intend to make these models more transparent and trustworthy, facilitating their safe and effective integration into clinical practice. Overall, integrating deep learning into lung cancer screening workflows can significantly assist radiologists in improving detection accuracy, ultimately leading to better patient outcomes.

This research can help both the AI developers and radiologists understand how these models work and make decisions and provide insights into how they can also be improved; this research will help the medical experts to understand how these models work and make decisions and provide insights into how they can be improved.

Significance of the Study (from Researchers' Perspective): This study holds significant value from a research perspective, as it addresses critical challenges and contributes to the advancement of knowledge and technology in early-stage lung cancer detection. The key points of significance are as follows:

Advancement in AI and Computer Vision: This study pushes the boundaries of AI and computer vision by developing interpretable deep learning models for complex medical tasks. The integration of 3D Convolutional Neural Networks (3D CNNs) with the Convolutional Block Attention Module (CBAM) showcases the potential of advanced AI techniques in handling high-dimensional medical imaging data (He et al., 2018).

Contribution to Explainable AI (XAI): A major contribution of this research is its focus on explainability. By employing techniques such as 3D Grad-CAM, the study addresses the critical need for transparency in AI models, making it easier for medical professionals to trust and understand AI-driven decisions. This work aligns with the growing emphasis on Explainable AI (XAI), which aims to make AI systems more interpretable and trustworthy (Barredo Arrieta et al., 2020).

Pioneering Research in Early Detection Models: The proposed model leverages state-of-the-art methodologies, such as artificial intelligence and machine learning, to improve the precision of early-stage lung cancer detection. This aligns with prior studies highlighting the critical role of AI in oncology diagnostics (Smith et al., 2021).

1.7 Scope and Limitations of the Study

The scope of this research is to develop 3D CNN models for lung cancer diagnosis while ensuring that these models are transparent and comprehensible for medical professionals, thereby enhancing their adoption in the medical industry.

Data Used in the Study: Type of Data: the data used in this study consists of volumetric CT scan images of lung nodules. These images provide detailed three-dimensional representations of the lungs, allowing for accurate detection and classification of nodules as benign or malignant. **Source Coverage:** the primary dataset used is the LUNA22 ISMI dataset. This dataset is part of the larger Lung Nodule Analysis (LUNA) Grand Challenge, which is a publicly available collection of lung nodule annotations from anonymized CT scans. **LUNA22 ISMI Dataset: Source:** Lung Nodule Analysis (LUNA) Grand Challenge **Composition:** The dataset includes 1,176 lung nodules with detailed annotations. These annotations are created by expert radiologists, providing reliable ground truth labels for training and evaluating deep learning models. **Time Coverage:** The time coverage of the dataset spans multiple years as the data is collected from various sources over time. The exact collection period may vary, but the LUNA Grand Challenge data has been compiled

and updated regularly to include diverse and comprehensive cases. **LUNA22 ISMI Dataset:** The dataset includes CT scans and annotations collected over the last five years, ensuring a varied range of cases and scenarios for robust model training. **Importance of the Dataset:** The use of the LUNA22 ISMI dataset justifies the use of high-quality, annotated data for training the model, which helps to achieve reliable and accurate performance in lung nodule classification. The dataset's diversity in terms of nodule size, shape, and location also helps in creating a model that generalizes well to real-world clinical scenarios. This comprehensive dataset forms the backbone of the study, enabling the development of interpretable deep-learning models for early lung cancer detection to improve patient outcomes.

As noted earlier, numerous deep-learning models have been successfully implemented for lung cancer detection. However, a significant research gap remains in the interpretability and optimization of these models. Providing insights into the reasoning behind predictions is crucial for enabling medical professionals to trust and effectively use these AI-driven solutions. The scope of explainable AI (XAI) is defined by its boundaries and limitations. This includes the specific areas where XAI is relevant and the characteristics that define what "explainable AI" means. Key aspects defining XAI's scope include its relevance to decision-making in fields like healthcare, finance, and criminal justice.

1.8 Methodology

The methodology of this study encompasses a systematic approach to design, develop, and validate the proposed model for early-stage lung cancer detection. It integrates data collection, preprocessing, model development, evaluation, and interpretation to ensure accuracy, reliability, and generalizability. Also, this study employed a comprehensive methodology to investigate the application of interpretable 3D Convolutional Neural Networks (CNNs) for accurate lung nodule classification.

1.8.1 Research Design

The study follows an experimental research design, which is a scientific approach used to investigate cause-and-effect relationships between variables. In this type of design, researchers actively manipulate one or more independent variables (IV) to observe their effects on dependent variables (DV) while controlling for other potential confounding factors. In the context of our study on early-stage lung cancer detection, the independent variable could be the type of model or algorithm used for detection (e.g., the proposed AI

model vs. traditional diagnostic methods), the dependent variable could be the performance metrics, such as accuracy, sensitivity, or specificity and control variables could include patient demographics, imaging quality, or clinical data parameters to ensure fairness in the evaluation. Using an experimental research design allows the study to systematically test hypotheses, validate findings, and provide actionable insights. By focusing on measurable outcomes like sensitivity, specificity, and AUC-ROC, the study ensures robust and scientifically sound results, establishing the model's viability for clinical adoption. And also it aimed at developing a predictive model and validating its performance on clinical and imaging datasets. The steps are organized into three main phases: data acquisition, model development, and performance evaluation.

1.8.2 Data collection and preparation

Dataset: We chose to leverage publicly available datasets curated for research purposes. Among the datasets we explored were LIDC-IDRI, LUNA16, Kaggle Data Science Bowl, and LUNA22-ISMI. After conducting thorough exploratory data analysis on these and other additional datasets, we selected the most recent and computationally inexpensive dataset for our study: the LUNA22-ISMI dataset. This selection was based on its suitability for our research needs and its ability to provide relevant and high-quality data for our deep learning model. **Data loading and Exploration:** CT scan images were loaded and visualized using ITK-SNAP software for initial exploratory data analysis. **Data Characteristics:** The distribution of Hounsfield Units (HUs) within the lung nodules was analyzed, revealing a range from -3024.0 to +6054.0. Data **Preprocessing: Windowing:** To focus on the lung parenchyma and reduce noise, HU values were clipped to the range of -1000 to 400 HU. **Normalization:** Min-Max scaling was applied to normalize pixel intensity values within the range of -1 to 1, ensuring consistent data representation for the model. **Data Augmentation:** To address the class imbalance and improve model robustness, various augmentation techniques were applied to the training data, including flipping, scaling, translation, noise addition, contrast adjustment, and elastic deformations. **Resizing:** 3D nodule patches were resized to a consistent dimension using spline interpolation to ensure compatibility with the input requirements of the 3D CNN models.

1.8.3 Implementation Tools

For implementing the proposed model for "Explainable 3D CNNs for Accurate Lung Nodule Classification," various tools and programming resources were utilized to streamline data

processing, model development, and evaluation. Here's an overview of the tools and their roles:

Implementation Tools Hardware: GPU (Graphics Processing Unit): High-performance GPUs, such as NVIDIA Tesla or RTX series, were used for training the deep learning models. GPUs accelerate the computation of large neural networks, significantly reducing training time.

CPU (Central Processing Unit): Standard CPUs were used for data preprocessing and other tasks that do not require extensive parallel computation.

Software: Python Programming Language: Python was the primary programming language used for implementing the research. Its extensive libraries and ease of use make it an ideal choice for developing machine learning models.

Python Packages Used for Modeling: TensorFlow: An open-source deep learning framework developed by Google. TensorFlow provides the underlying infrastructure for training and constructing deep learning models.

Keras: A high-level neural networks API that runs on top of TensorFlow. Keras simplifies the process of training neural networks, making it easier to experiment with different architectures.

NumPy: A necessary library for scientific computing with Python. NumPy supports large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.

SciPy: An open-source library used for scientific and technical computing. SciPy is used for tasks such as data augmentation, statistical analysis, and numerical integration.

Scikit-learn: A machine learning package that provides simple and efficient algorithms and tools for data analysis. Scikit-learn has been critical for preprocessing tasks, such as scaling and splitting data, as well as for evaluating model performance using metrics like accuracy, recall, precision, and F1-score.

Matplotlib and Seaborn: A plotting library used for creating static, interactive, and animated visualizations in Python. It is used to visualize the training process and performance metrics.

Seaborn: This is a data visualization library based on Matplotlib, which offers a high-level interface for sketching attractive statistical graphics.

OpenCV: An open-source computer vision package that is used for image and video processing tasks, such as reading, writing, and manipulation.

Grad-CAM (Gradient-weighted Class Activation Mapping): An implementation of the Grad-CAM technique used for generating visual explanations of the model's predictions. This helps in understanding regions of the input image that are mostly influencing the model's decision.

H5py: A Pythonic interface to the HDF5 binary data format. It was used to store and manage large amounts of numerical data generated during the training process.

Use of Python Programming: Python programming plays a crucial role in conducting research and enabling the development and implementation of complex deep-learning algorithms. Its extensive libraries and frameworks provide the necessary tools to preprocess data, design and train neural networks, and evaluate model performance. Python's simplicity and readability also facilitated rapid prototyping and experimentation, allowing us to fine-tune the models for optimal performance.

1.8.4 Evaluation Method

After splitting the data set using the hold-out method, the model performance is evaluated using a set of metrics, such as accuracy, recall, precision, F1-score, and AUC-ROC.

1.9 Organization of the Thesis

There are six chapters in the thesis. The second chapter is dedicated to understanding the fundamentals of medical imaging, specifically scans, Convolutional Neural Networks of DL and related works done for lung cancer screening, and finally, XAI and related works done in the healthcare sector. The third chapter discusses the materials and methods used for the research work. Chapter four is devoted to describing the implementation of the proposed model. Chapter five is all about the discussion of results found in every step of the implementation phase, including the performance metrics used for evaluation of the DL models. Last but not least, the last chapter is everything regarding conclusions and recommendations for future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview

This chapter presents an extensive review of literature on topics directly related to this research. As previously mentioned, the primary objective of the current study is to develop an interpretable deep-learning model for detecting and classifying lung cancer from lung CT scan data. The following sections explore fundamental concepts in medical imaging and the data associated with medical images, particularly CT scans. We then discuss topics related to Convolutional Neural Networks (CNNs), their applications in computer vision, and relevant works on lung cancer detection and classification. Finally, we delve into Explainable AI (XAI), its taxonomies, and related XAI applications in healthcare.

2.2 Medical Imaging

Medical Imaging refers to a collection of methods and technologies applied to create visual representations of the internal structure of the human body. It allows healthcare professionals to:

Diagnose diseases: Identify abnormalities, such as tumors, fractures, or infections. **Guide treatment planning:** Plan surgical procedures, radiation therapy, and other interventions. **Monitor treatment progress:** Track the effectiveness of treatment plans and assess disease progression. **Conduct research:** Advance medical knowledge and develop new diagnostic and therapeutic approaches.

Visual representation, with the help of Medical Imaging, is crucial for identifying and diagnosing a variety of illnesses, guiding the development of treatment plans, and monitoring their effectiveness. Medical imaging is extensively employed to visualize internal structures of body parts, such as muscles, organs, bones, blood vessels, and others (Islam et al., 2023).

2.2.1 General Workflow of Radiologists

A radiologist's workflow begins with a doctor referring a patient for imaging and the radiology department scheduling the appointment. Depending on the type of scan, patients may need to fast or take contrast material. The radiologist then positions the patient, adjusts the machine settings, and ensures the patient's comfort and safety. The technician captures

the images and adds relevant metadata. Finally, the radiologist analyzes the images, compares them to previous studies, and writes a report with observations, conclusions, and recommendations (Giard, 2023).

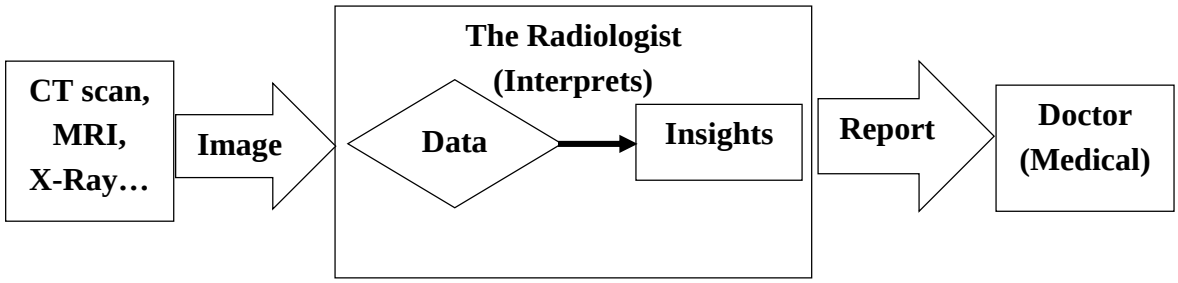


Figure 2.1 Basic workflow of a radiologist (Giard, 2023).

2.2.2 CT-scan

Medical imaging modalities encompass various techniques designed to generate visual representations of the human body, aiding in the diagnosis and treatment of diseases (Islam et al., 2023). Common modalities include Radiography, Computed Tomography, Magnetic Resonance Imaging, Ultrasound Imaging, and Positron Emission Tomography. **Radiography (X-ray Imaging):** Produces 2D images through projection radiography, such as chest X-rays. **Computed Tomography (CT):** Generates detailed cross-sectional images of the body and produces 3D visualizations using X-rays. **Magnetic Resonance Imaging (MRI):** Provides high-resolution 3D images using magnetic fields and radio waves. **Ultrasound Imaging:** Captures 2D or 3D images using high-frequency sound waves. **Positron Emission Tomography (PET):** Produces 3D images to visualize metabolic processes in the body.

A CT-Scan, or a CAT-Scan, stands for Computed Axial Tomography, Where the term 'tomography' comes by combining two Greek words, 'tomos' (meaning 'slice') and 'graphein' (meaning 'to write'). The CT scan uses radiation to create detailed images of internal structures from various angles by measuring the intensity of X-rays passing through the body part this is -achieved with the help of detectors. More details on CT scan imaging can be found in Appendix C.

2.2.3 Hounsfield Units

The physical density of a tissue or organ is directly linked to photon attenuation or absorption. The CT detectors measure the extent to which tissues attenuate photons (i.e.,

their density), and the image processor converts this data into byte values, which are then used to assign appropriate pixel brightness in the image. Each value of the image on this scale is known as a Hounsfield Unit (HU), named after Hounsfield, and the full range of these values constitutes the Hounsfield scale. (Shady Hermena & Michael Young, 2023).

In CT scans, the physical density of a tissue or organ directly correlates with its ability to attenuate (absorb) X-ray photons. The CT scanner measures the degree to which different tissues attenuate X-rays and translates these measurements into a standardized scale called Hounsfield Units (HU).

HU Scale: is a relative measure of radiodensity, where- Water is assigned a value of 0 HU, Air is assigned a value of -1000 HU, and Dense bone can have values exceeding +1000 HU. *HU Values and Tissue Density:* Higher HU values indicate denser tissues that absorb more X-rays. Lower HU values indicate less dense tissues that allow more X-rays to pass through. *Image Formation:* The CT scanner measures the attenuation of X-rays passing through the body at multiple angles. This data is then processed by a computer to reconstruct 3D images of the body. Each pixel in the CT image is assigned an HU value, representing the relative density of the corresponding tissue. *Clinical Significance:* HU values provide valuable information for radiologists to differentiate between different types of tissues and identify abnormalities. For example, lung tissue typically has low HU values, while bone has high HU values. Abnormalities such as tumors or calcifications may have distinct HU values that help in their diagnosis. (Shady Hermena & Michael Young, 2023). Figure 2.2 shows the HU scales of different substances or materials.

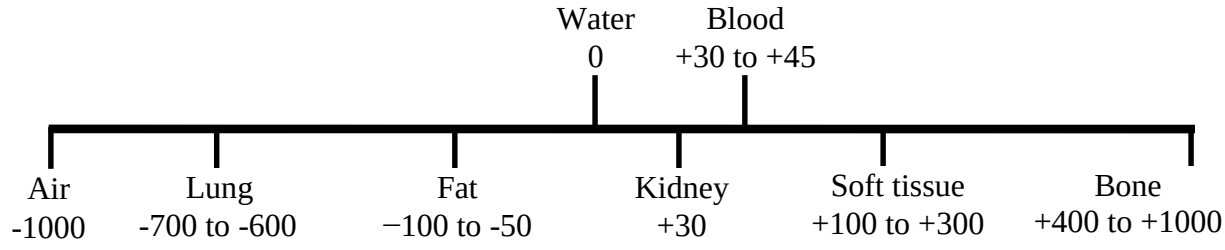


Figure 2.2 Hounsfield Unit Scale (https://en.wikipedia.org/wiki/Hounsfield_scale).

In CT scans, the Hounsfield Unit (HU) values for various tissues are calculated using an Equation. (2.1) (Shady Hermena & Michael Young, 2023).

$$HU = 1000 * \frac{\mu_{tissue} - \mu_{water}}{\mu_{water} - \mu_{air}} \qquad 2.1$$

Where: μ_{tissue} is the linear attenuation coefficient of the tissue or organ and μ_{water} and μ_{air} are the linear attenuation coefficients of water and air, respectively (Shady Hermena & Michael Young, 2023).

The Hounsfield unit (HU) is a measure of tissue density based on X-ray attenuation, represented pixel by pixel. A higher HU value indicates denser tissues, such as bone, while a lower HU value indicates less dense tissues, such as air. The Linear Attenuation Coefficient (μ) quantifies how X-rays interact with tissues; a higher μ means more X-ray absorption (denser tissue), while a lower μ means less X-ray absorption (less dense tissue).

HU is derived from the linear attenuation coefficient of a tissue compared to water (reference point). In a CT scan, that is proportionate to the degree of X-ray attenuation (Al-Zahrani, 2017). The conversion from pixel data to HU is a linear transformation using equation (2.2) (Shady Hermena & Michael Young, 2023).

$$HU = pixel\ value * \ slope + intercept \qquad 2.2$$

The formula in equation (2.2) requires two key values stored within the CT scan image files. Rescale slope (slope) scales the pixel intensity to physical units, and Rescale intercept (intercept) adjusts the baseline for the HU scale.

2.2.4 Data Formats in Medical Imaging

The DICOM and NIFTI formats are among the most widely used data and image formats in medical imaging due to their efficiency and compatibility. These formats are not exclusive to CT scans but are extensively utilized across a broad range of imaging modalities, such as MRI, X-ray, Ultrasound, and PET scans. DICOM (Digital Imaging and Communications in Medicine) is particularly notable for its ability to store not only image data but also metadata, including patient information, imaging parameters, and equipment details, making it indispensable in clinical workflows. On the other hand, NIFTI (Neuroimaging Informatics Technology Initiative) is primarily used in neuroimaging research due to its streamlined format for storing volumetric data. Beyond these, other formats like JPEG and PNG are occasionally used for simplified visualization, though they lack the comprehensive metadata capabilities of DICOM and NIFTI. Each format serves a specific purpose, ensuring versatility in handling diverse medical imaging needs. (Heindl, 2022; Li et al., 2016).

DICOM: “Digital Imaging and Communications in Medicine is the international standard for medical images and related information. It defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use.” (NEMA PS3 / ISO 12052 et al., 2024). It is used for storing and sharing medical images in hospitals. DICOM files have a file extension of “.dcm” or “.dicm.” It has two parts: Header: Contains patient information, study details, and image metadata. And Body: Stores the actual image data (2D, 3D, or 4D) (Heindl, 2022; Li et al., 2016).

Primarily used in clinical settings, raw scanner data often comes in DICOM format. Tools exist to handle DICOM files, like the “PYDICOM” library, their complexity often leads to conversion to other formats for research and machine learning (Li et al., 2016). They have Key-value pairs for easy access to data using unique tags. An example is shown in Figure 2.3

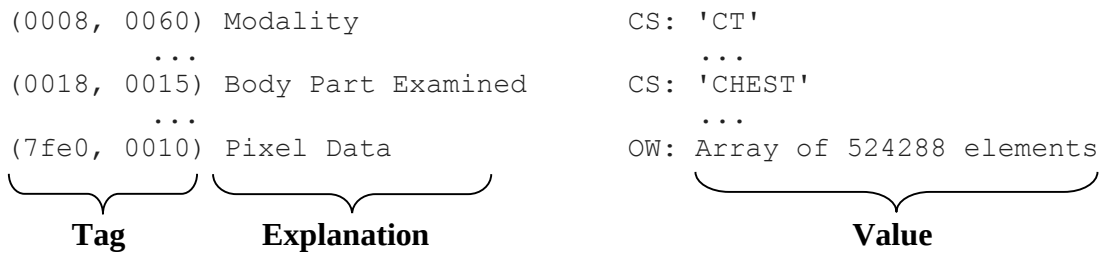


Figure 2.3 Sample DICOM file key-value pairs to access the data (from our visualization).

NIfTI: Neuroimaging Informatics Technology Initiative Open file format for medical images, often used in neuroimaging research. Simpler and more interoperable than DICOM. Smaller file size, easier for storage and sharing. It has two parts: The Header contains essential information about image geometry, not patient or scanner details. The body stores the actual image data (Heindl, 2022; Li et al., 2016). Similar to the DICOM files, the data can be accessed using libraries with header index.

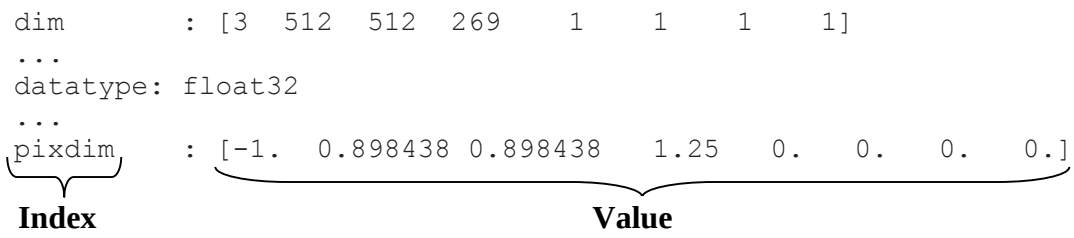


Figure 2.4 Sample metadata of NIFTI formats (from our visualization)

NIfTI and DICOM differ from each other in the following ways: while DICOM works with 2D layers, NIfTI can display 3D detail (with NIfTI files, images, and other data saved in a

3D format). NIfTI can take longer to load, while DICOM allows users to display one layer at a time. Less metadata is included in NIfTI files. More information can be contained in DICOM files.

2.3 Deep Learning

Deep Learning (DL) is a subfield of Machine Learning (ML) that utilizes models based on Artificial Neural Networks (ANN). DL algorithms have extensive applications in healthcare, particularly in informed clinical decision support systems. Compared to DL, classical ML has not been as effective in addressing problems such as Natural Language Processing (NLP), image detection, image recognition, and other complex tasks. Therefore, for this research study, DL and its various architectures and tools, especially for feature extraction, are the appropriate choices. DL's ability to automatically learn and extract intricate features from data makes it highly suitable for advancing healthcare technologies and improving diagnostic accuracy.

Powerful computational models are capable of learning complex representations from data through multiple processing layers. These models have achieved state-of-the-art results in various domains, including speech recognition, image recognition, and object detection. They utilize backpropagation to adjust internal parameters and uncover intricate patterns in large datasets. There are different types of deep learning models (Lecun et al., 2015): **Convolutional Neural Networks (CNNs):** Excel in processing images, videos, and audio by capturing spatial hierarchies in data through convolutional layers. **Recurrent Neural Networks (RNNs):** Specialized in handling sequential data like text and speech, making them ideal for tasks involving time series and natural language processing. **Generative Adversarial Networks (GANs):** Known for generating realistic images, videos, and other data by pitting two neural networks against each other in a game-theoretic framework. **Autoencoders:** Used for tasks such as dimensionality reduction, denoising, and unsupervised learning by encoding input data into a lower-dimensional space and then reconstructing it.

These models continue to push the boundaries of what artificial intelligence can achieve, providing advanced solutions across various applications.(Lecun et al., 2015).

The fundamental components of deep learning systems are Neural Networks (NN) or Artificial Neural Networks (ANN). The term "network" refers to a structure similar to a

graph, while "neural" is derived from "neuron." Thus, an "Artificial Neural Network" is a computing system designed to mimic or, at the very least, draw inspiration from the neural connections found in the human nervous system (Rosebrock, 2017). These networks consist of interconnected nodes or "neurons" that process and transmit information, much like the neurons in the human brain. By adjusting the connections and weights between these nodes through training, ANNs can learn to recognize patterns and make predictions, enabling them to perform complex tasks such as image recognition, natural language processing, and more. This biologically inspired approach is at the core of many advancements in artificial intelligence and machine learning.

For a system to be classified as a Neural Network (NN), it must have a directed, labeled graph structure where each node in the graph performs simple computations. Each of these nodes, or individual neurons, is referred to as a "perceptron" or "artificial neuron" (Rosebrock, 2017). The perceptron is the fundamental building block of an NN, capable of processing input data and generating an output based on learned weights and biases. This structure allows NNs to model complex relationships in data, making them powerful tools for tasks such as pattern recognition, classification, and regression. By connecting multiple perceptrons in various layers, NNs can learn to perform intricate functions and make accurate predictions across different applications.

Each node in a Neural Network (NN) performs a basic calculation. The output, or signal, from this calculation is transmitted to other nodes through connections, each marked with a weight that indicates the degree of signal amplification or attenuation. Connections with significant positive weights amplify the signal, highlighting its importance in the categorization process. Conversely, connections with negative weights attenuate the signal, suggesting that the node's output is less critical for the final categorization. When a system has a graph structure with adjustable connection weights via a learning algorithm, it is referred to as an Artificial Neural Network (ANN) (Rosebrock, 2017). This configuration allows ANNs to learn from data by adjusting the weights during training, enabling them to

recognize patterns and make accurate predictions. Figure 2.5 illustrates a single-layer artificial neuron or perceptron, showcasing the fundamental building block of an ANN.

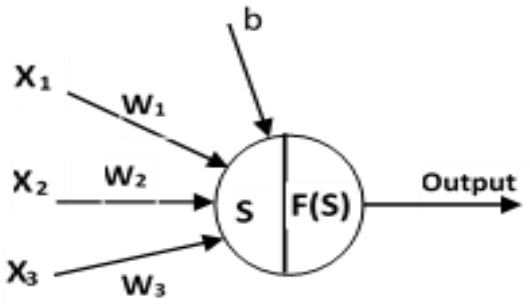


Figure 2.5 Single-layer neural network (https://www.researchgate.net).

It receives all the inputs (X_1, X_2, X_3) and the weights (W_1, W_2, W_3), and the summation function sums all the inputs multiplied by the weights and then adds the bias:

$$S = [\sum_{i=1}^{n} w_i * x_i] + b \qquad 2.3$$

F(S) is an activation (transformation) function; the output of the summation function can be the input to the activation function. After we apply the activation function on it, it will give us Output.

$$Output = F(S) \qquad 2.4$$

There are different types of activation functions some of them are the Step function, Signum function, linear function, ReLu function, Leaky-ReLu function, sigmoid function, Softmax function, and Hyperbolic Tangent function. Figure 2.6 shows a multi-layered feed-forward NN.

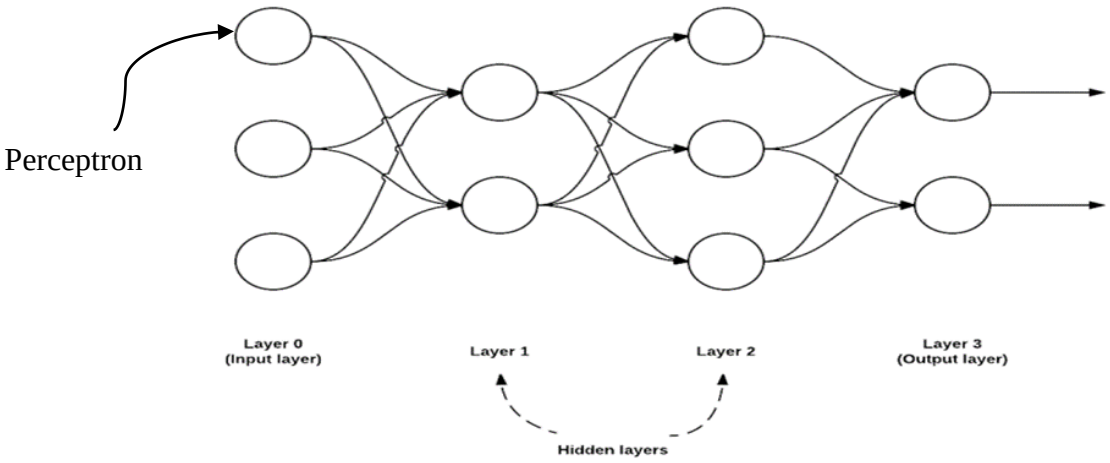


Figure 2.6 a multi-layer feed-forward ANN (https://www.researchgate.net).

Deep Learning (DL) models are a powerful subset of machine learning algorithms that learn intricate representations from data through multiple processing layers. They have achieved state-of-the-art results in various domains, including image recognition, natural language processing, and speech recognition. Deep Learning (DL) models are commonly categorized into three types: supervised, unsupervised, and semi-supervised.

Supervised DL models: These use labeled data or examples where the target variable or desired output is known. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are widely used supervised DL models. CNNs, in particular, excel at processing image, speech, or audio signal inputs and are highly effective in applications such as image classification, object detection, feature extraction, and image segmentation. *Unsupervised DL models:* These extract structure and patterns from unlabeled data. Autoencoders are prime examples of unsupervised neural networks, which learn to encode and reconstruct input data, helping to discover underlying data representations.

Semi-supervised DL models: These are used when labels are available for part of the data but not for the rest. They combine the strengths of both supervised and unsupervised learning to improve model performance when labeled data is limited. By leveraging these types of DL models, researchers can address a wide range of complex problems in various domains, advancing the capabilities of artificial intelligence (Rosebrock, 2017).

Convolutional Neural Networks (CNNs) have proven particularly effective in various image processing tasks; there are three main kinds of layers in CNN. The first one is the Convolutional Layer, which is a core component Responsible for most computations in CNNs. Key elements include Input data, which is the image to be analyzed. Filters are small matrices that detect specific features in the image. Feature maps that are outputs of filters, indicating where features are found. The convolution process involves a filter sliding across an image and computing dot products with input regions. The resulting output feature map highlights the locations where the filter detects the desired features. Key hyperparameters in this process include:

- **Number of filters:** Determines the number of feature maps produced.
- **Stride:** Controls how much the filter shifts, affecting the output size.
- **Padding:** Adds zeros to the image border, if needed, to maintain the output size.

After each convolution operation, an activation function is applied to introduce non-linearity into the CNN model, which helps the model learn more complex relationships in the data.

This step is crucial for enabling the model to capture intricate patterns. Figure 2.7 demonstrates how the convolution layer operates and provides the formula to calculate the size of feature maps. By fine-tuning these hyperparameters, CNNs can be optimized for various tasks, such as image classification, object detection, and feature extraction.

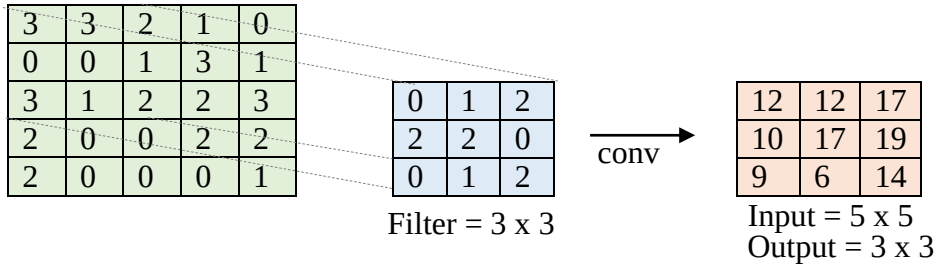


Figure 2.6 Illustration of how the convolution is done.

$$output\ size = \frac{input\ size - filter\ size + (2 * padding)}{strid} + 1 \qquad 2.5$$

Convolution layers can be stacked sequentially within a Convolutional Neural Network (CNN). When this occurs, the structure of the CNN becomes hierarchical, as the pixels in the receptive fields of earlier layers are accessible to subsequent layers. This hierarchical arrangement allows the CNN to progressively capture more abstract and complex features from the input data. Early layers typically detect simple patterns, such as edges or textures, while deeper layers identify more sophisticated structures, like shapes or objects. This layered approach is crucial for the effective performance of CNNs in various tasks, including image classification, object detection, and feature extraction.

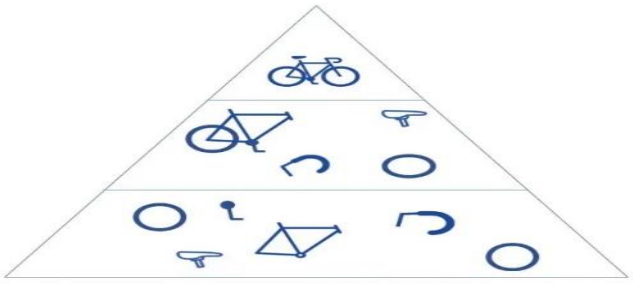


Figure 2.7 Hierarchical CNN structure

(Source: http://viraai.com/wp-content/uploads/2021/10/5.jpg).

The second key type of layer in Convolutional Neural Networks (CNNs) is the **Pooling Layer**, which is designed to downsize feature maps. This process reduces both the number of parameters and the spatial dimensions (width and height) of the data, making

computations more efficient. Pooling layers help the model focus on the most prominent features while discarding less critical details, thereby simplifying the overall structure.

There are two primary types of pooling: Max pooling and average pooling. **Max Pooling**: extracts the maximum value from each local region within the feature map. It effectively captures the most important feature (the strongest activation) in that region. **Average Pooling**: on the other hand, computes the average of the values in each local region, providing a more generalized representation of the region. (Zhao, L., Zhang, Z. A, 2023)

Pooling serves several important purposes, as presented below: **Robustness to Variations**: By focusing on key features, pooling makes the model less sensitive to small changes in the image, such as translations, distortions, or other minor variations. This ensures that the network maintains performance even if the input image undergoes slight transformations. **Overfitting Reduction**: By downsampling the data and reducing the overall complexity, pooling helps prevent the model from overfitting, ensuring better generalization to unseen data.

The pooling operation typically uses a fixed window size and stride, similar to convolutional layers, to systematically scan through the feature maps. Pooling layers are crucial for - building deeper networks that are computationally efficient and capable of extracting high-level abstractions from the input data.

The third type of layer in a Convolutional Neural Network (CNN) is the Fully Connected Layer (FC layer). This layer analyzes features extracted by previous layers and classifies the data. Each output node in the FC layer is connected to all other nodes, enabling the integration of information across the entire network. Typically, the SoftMax activation function is used for multiclass classification, while the sigmoid function is employed for binary classification. These activation functions transform the output into probability values ranging from 0 to 1, which represent the likelihood of each class or category. By using FC layers, CNNs can effectively perform tasks such as image classification, object detection, and feature extraction, delivering accurate and reliable results.

Convolutional Neural Networks (CNNs) have undergone substantial architectural evolution, transforming from relatively simple designs to highly complex and efficient models. A foundational milestone in this journey was LeNet, introduced by Yann LeCun in 1989. Designed for handwritten digit recognition, LeNet utilized a shallow seven-layer architecture

that operated on grayscale images of size 32x32 pixels. Its success demonstrated the potential of CNNs for automating feature extraction and achieving high accuracy on image-based tasks.

Following LeNet, the field witnessed a rapid acceleration in the development of more powerful CNN architectures. A turning point came with AlexNet, the winner of the 2012 ImageNet competition. AlexNet, featuring an eight-layer deep architecture, leveraged advances in computational resources like GPUs to showcase the power of deeper models. Its groundbreaking performance highlighted the potential of deeper networks for tackling complex vision tasks, sparking a wave of innovation. Subsequent architectures further pushed the boundaries of depth and efficiency; *VGG (2014)* Introduced a family of architectures that demonstrated the effectiveness of stacking numerous convolutional layers with small 3x3 filters. This approach emphasized simplicity and uniformity, making VGG a widely adopted baseline for research. *ResNet (2015):* Revolutionized deep learning with the introduction of residual connections, allowing networks to scale to unprecedented depths without suffering from vanishing gradients. *Inception (2014):* Pioneered the use of multi-scale feature extraction within a single layer through its "Inception modules," improving efficiency and performance. *DenseNet (2016):* Enhanced feature reuse by connecting each layer to every subsequent layer, leading to more efficient parameter utilization. *EfficientNet (2019):* Optimized network design by balancing depth, width, and resolution, achieving state-of-the-art performance with fewer parameters and lower computational costs.

These advancements have not only expanded the capabilities of CNNs but also inspired innovations in optimization techniques, hardware accelerations, and applications across diverse fields like medical imaging, autonomous vehicles, and natural language processing.

The training of Convolutional Neural Networks (CNNs) is accomplished using the backpropagation algorithm. This process involves calculating the weights and biases that best fit the model by minimizing the error, which is the difference between the predicted and actual values. Backpropagation updates the weights and biases of the network in a way that reduces this error, allowing the model to learn and improve its performance over time. By iteratively adjusting these parameters, the network can better capture the underlying patterns in the data, leading to more accurate predictions and classifications.

2.4 Explainable AI (XAI)

2.4.1 Introduction to XAI

The field of XAI (Explainable Artificial Intelligence) aims to make machine learning models more understandable and interpretable. This is particularly important in sensitive areas like healthcare and finance, where users need to trust and understand the decisions made by these models. The terminologies used and goals of XAI are as follows as given by (Ali et al. 2023; Barredo Arrieta et al., 2020; Conor O'Sullivan, n.d.)

One of the terminologies used is Explainability, which describes the active process of making a model's inner workings clear and easy to understand. The other is Interpretability, which shows the model's inherent characteristic of being understandable to humans. The third one is Understandability, the ability of a model to be understood without detailed explanations. Also, Comprehensibility is used to describe the ability of a model to express its knowledge in a way that humans can understand. Finally, the term Transparency expresses the ability of a model to be understood on its own.

As pointed out by (Ali et al., 2023 Barredo Arrieta et al., 2020 and Conor O'Sullivan, n.d.), the goals of XAI include the following. One is trustworthiness, which means that XAI enables the building of trust with domain experts and users affected by the model's decisions. Two, Causality, i.e., understanding the causal relationships between inputs and outputs. Third, Transferability; XAI ensures the model's performance generalizes to new data. Four, Informativeness: providing evidence to support the model's predictions. Five, Confidence; ensuring the model's predictions are reliable and stable. Six, Fairness: mitigating biases and ensuring fair outcomes for all. Seven, Accessibility: making the model's explanations accessible to different stakeholders. Eight, Interactivity, allowing users to interact with the model and explore its explanations. Finally, Privacy is protecting user privacy while providing explanations.

Explainable AI is focused on creating AI systems that are transparent and explainable to human users. This allows users to understand how decisions are being made by the system and to detect biases or errors. These aspects together define the scope of explainable AI, where decisions are made based on AI while also emphasizing human interaction and alignment with human values and goals.

XAI research seeks to achieve these goals through various techniques, ultimately aiming to improve the trust, reliability, and responsible use of machine learning models.

According to Kenton (2023), there are Black Box, White Box, and Gray Box Models in XAI.

- Black Box: Opaque inner workings. Difficult to understand and trust its decisions.
- White Box: Fully interpretable but often less accurate.
- Gray Box: Offers a balance between interpretability and accuracy, allowing partial understanding of the model's reasoning.

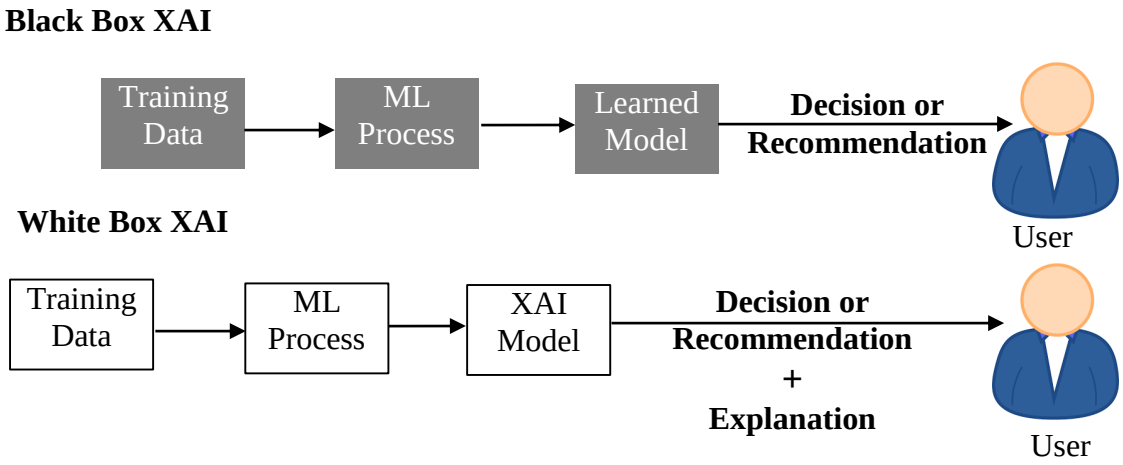


Figure 2.8 Black Box AI vs. White Box XAI.

XAI research suggests that it might be possible to overcome the traditional tradeoff between model complexity (and accuracy) and interpretability. This means we could potentially have models that are both accurate and understandable (Barredo Arrieta et al., 2020).

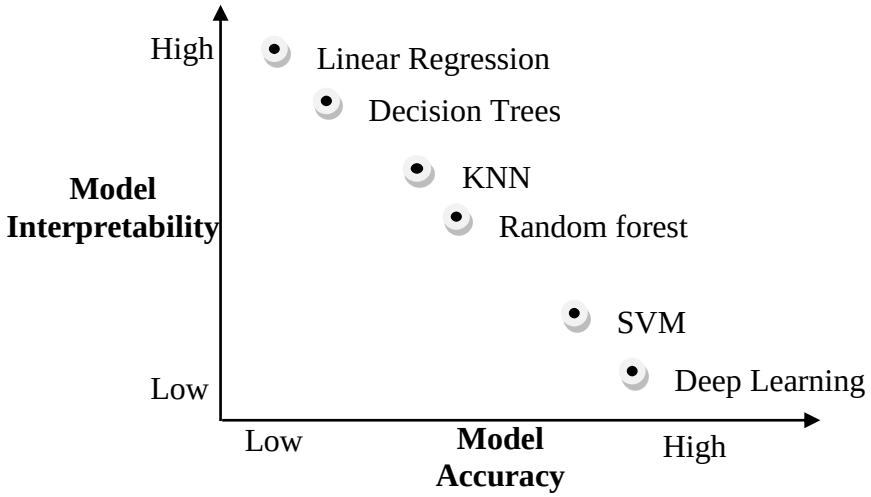


Figure 2.9 Interpretability vs. Accuracy.

Similarly, (Conor O'Sullivan, n.d.) Has put the interpretability spectrum as shown in Figure 2.11 below. As models become more complex, they become less interpretable. An ML model is a function. The input consists of the model's features, while the output is its predictions. A function that is too complex for a human to comprehend is a black-box model. To be able to peek inside the black box and comprehend how the model functions, we require an extra approach.

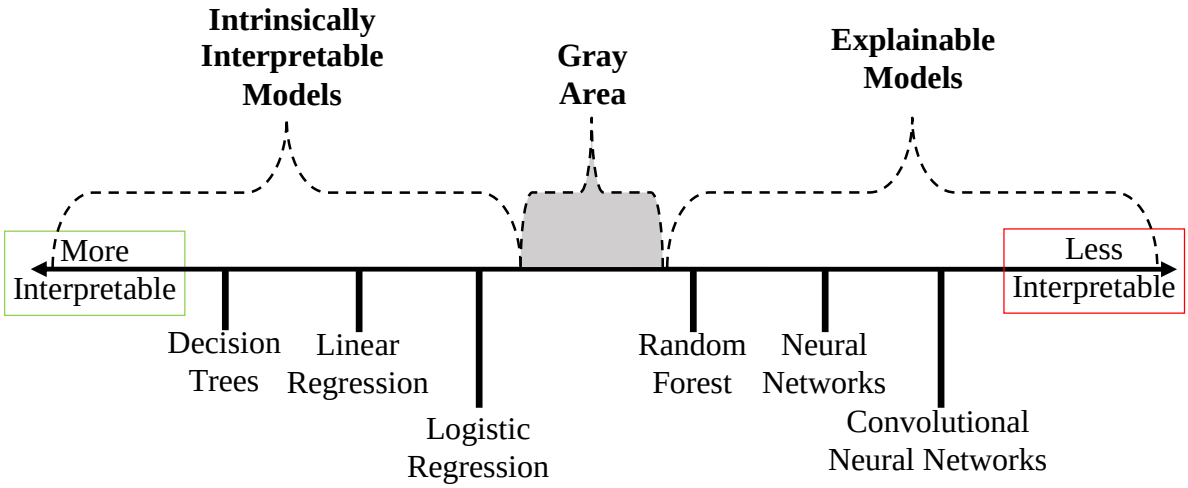


Figure.10 The Interpretability Spectrum (Interpretability vs. Explainability).

Intrinsic interpretability emphasizes creating simple models, such as decision trees or linear regression, that are naturally understandable because of their straightforward structure. However, these models often face limitations in capturing complex non-linear relationships within the data (Stiglic et al., 2020). This trade-off between simplicity and the ability to model intricate patterns highlights the ongoing challenge of achieving both high interpretability and performance in machine learning models.

2.4.2 Taxonomy of Explainability Methods

Taxonomies are indeed valuable for organizing discussions, but the sheer variety and complexity of Explainability techniques make it challenging to create a single, practically applicable taxonomy (Speith, 2022). The diverse methods used in Explainable Artificial Intelligence (XAI) span different domains and address various aspects of model interpretability, making it difficult to encapsulate them all under one unified framework. As a result, multiple taxonomies and categorization approaches are often employed to better address the specific needs and contexts of different Explainability techniques.

There are various classification criteria used to categorize techniques for ML interpretability. Some of them are as follows: as described by (Molnar, 2022).

Local vs. Global Explanations: This categorization criterion examines whether an Explainable AI (XAI) method explains a particular sample or the entire model. It asks whether the model's overall behavior or a specific prediction is elucidated by the interpretation method.

Local Explanations: These provide insights for individual samples. Results from local methods can be averaged across samples. Local interpretability highlights the importance of features for any single prediction, illustrating the model's behavior at a specific data point. This involves a sort of "what-if" analysis, where changes to feature values for selected data points are observed, and the resulting changes in the prediction value are noted. Local explanations can show how a prediction would change when a feature changes (Molnar, 2022).

Global Explanations: These provide insights for the entire model or groups of samples. In global interpretation, an overall view of the model is given, along with data predictions and explanations. This includes data exploration, which displays an overview of the dataset along with prediction values, and global importance, which aggregates feature importance values across individual data points and demonstrates how features affect changes in model prediction values (Molnar, 2022). Global methods will weigh input parameters the same way regardless of individual predictions.

Model Agnostic vs. Model Specific Explanations: This classification asks the question, does the interpretation depend on a particular Model? Model-specific interpretation tools are limited to specific models, while model-agnostic tools can be used on any ML model. Agnostic methods usually work by analyzing feature input and output pairs. These methods cannot have access to model architecture, such as layer weights or structural information (Molnar, 2022). Figure 2.12 shows some Explainability methods in their class.

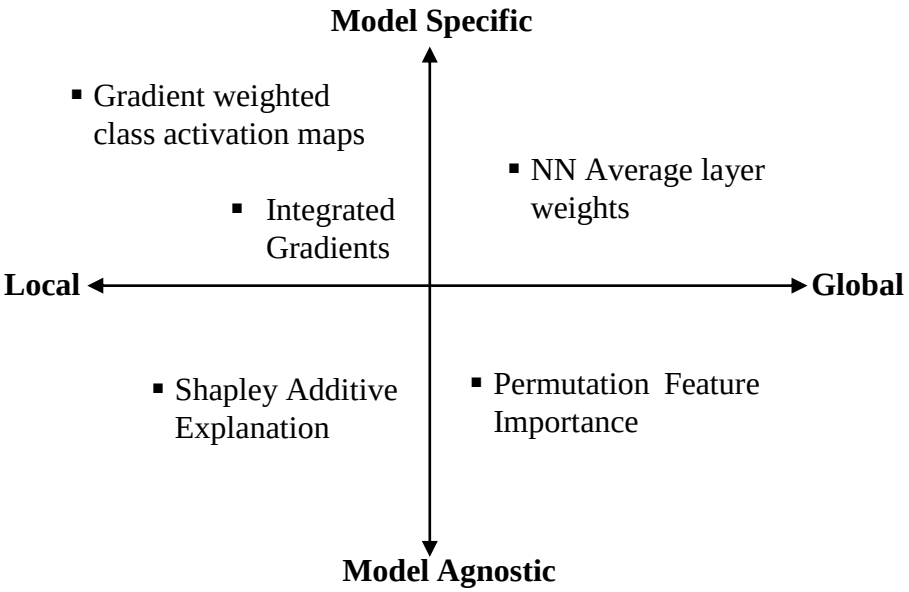


Figure 2.11 Examples of XAI methods in their class.

Ad-hoc vs Post-hoc Explanations ask the question, when does the explanation occur? In ad-hoc explanations, the model has been designed to be intrinsically explainable (Molnar, 2022).

Data modality-specific vs. Data modality-agnostic explanations

These are based on the type of data they handle. *Data Modality Specific Explanations are* methods that are tailored to a specific type of data, such as tabular data, time-series data, or imaging data. For example, Grad-CAM (Gradient-weighted Class Activation Mapping) is specifically applicable to data related to Convolutional Neural Networks (CNNs) like images.

On the other hand, *Data Modality Agnostic Explanations* are designed to be versatile and can work with any type of data. They are often model-neutral or model-agnostic, meaning they can be applied regardless of the underlying machine-learning model. These explanations provide flexibility in analyzing and understanding different data types without being limited to a specific modality (Molnar, 2022).

By understanding the differences between these approaches, users can choose the most appropriate Explainability method based on the type of data they are working with.

Surrogate Models vs. Attribution/Visualization Methods

Surrogate models use an interpretable model or a simpler one that simulates a black box's behavior. Attribution/ visualization methods attempt to visualize certain aspects of the model to allow an explanation of why and how the model reaches a decision (Molnar, 2022).

On the other hand, a work by Speith (2022) reviewed recent approaches and classified them into four approaches: The functioning-based approach, which is the way the Explainability methods extract information from the model, the other is result-based approach, which focuses on interpreting or extracting insights directly from the outputs or predictions of the model. The conceptual approach is to partition the field into several conceptual dimensions, and the mixed approach is a hybrid of the above three approaches. The paper's thorough taxonomy approaches are comprehensive and beginner-friendly the illustration below shows the mixed approach, which is likely good for newcomers in the field. It has Top level: scope distinction. Middle level: applicability distinction. And Bottom level: elements of other approaches.

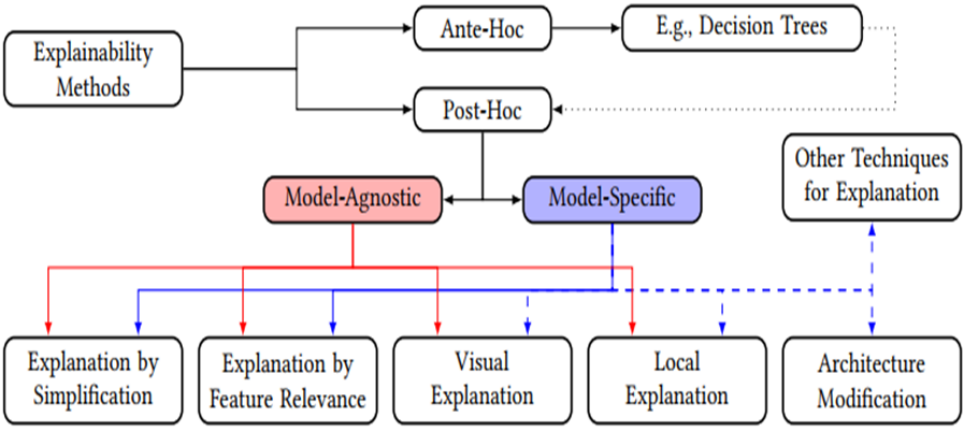


Figure 2.12 General taxonomy of the hybrid approach.

2.4.3 Overview of Popular XAI Methods

LIME (Local Interpretable Model-Agnostic Explanations) Creates simplified models around specific data points to understand their influence on predictions (Ribeiro et al., 2016).

SHAP (Shapley Additive exPlanations): Uses game theory concepts to attribute credit for predictions to individual features, providing detailed insights into their impact (Lundberg & Lee, 2017).

PFI (Permutation Feature Importance): Assesses feature importance by shuffling feature values and observing the model's performance change (Mi et al., 2021).

LRP (Layer-wise Relevance Propagation): Back propagates the prediction score through the model, assigning relevance scores to features and activations in each layer, providing a more granular understanding of the decision-making process (Bach et al., 2015; Sarker, 2021).

Grad-CAM, which stands for Gradient-weighted Class Activation Mapping, is a technique used in deep learning, especially with convolutional neural networks (CNNs). It helps us understand what parts of an image are most important to CNN's prediction for a particular class. According to (Selvaraju et al., 2017), it works with an input image and model (pre-trained CNN model). The model analyzes the image and calculates the gradients of the final classification score concerning the activations of the last convolutional layer. Gradients indicate how much each activation in the layer influences the final score (Grad Calculation).

Then, gradients are used to weigh the activations in the convolutional layer, creating a class activation map. This map highlights the image regions that have the most influence on the classification (Class Activation Map). The class activation map is overlaid on the original image, typically as a heatmap. Brighter areas represent image regions the model focused on for its prediction (Visualization). Grad-CAM offers a way to interpret the "black box" nature of deep learning models, particularly CNNs used in image recognition. By visualizing the important regions, we gain insights into why the model made a specific prediction.

Grad-CAM is versatile and can be applied to various CNN architectures without modifying the model itself. Grad-CAM is particularly useful for diagnosing model predictions and improving trust in their decisions. There are advanced libraries available, like Pytorch-grad-cam, that provide functionalities for implementing Grad-CAM (Selvaraju et al., 2017).

There are many other XAI techniques as well, XAI is a rapidly evolving field with a variety of techniques to understand and explain AI models. By choosing the right XAI method for your specific needs, you can unlock the full potential of AI while ensuring transparency and responsible use.

2.4.4 Explainability in Healthcare Applications

Explainable Artificial Intelligence (XAI) has become a crucial area of research in healthcare applications that utilize deep learning models. Despite the impressive accuracy these models can achieve in tasks like medical image analysis and disease prediction, their "black-box" nature raises significant concerns regarding transparency, accountability, and trust among medical professionals and patients. The importance of Explainability in healthcare applications includes the following:

Transparency: Providing clear and understandable insights into how deep learning models make decisions, enabling healthcare professionals to trust and validate the results.

Accountability: Ensuring that the models' predictions and decisions can be traced back and justified, which is essential for ethical considerations and regulatory compliance.

Trust: Building confidence among healthcare providers and patients in AI-driven outcomes, fostering acceptance and adoption of advanced technologies in medical practice.

Improved Decision-Making: Allowing clinicians to interpret and understand model predictions, leading to better-informed clinical decisions and personalized treatment plans.

Bias Detection: Identifying and mitigating potential biases in the models, ensuring fair and equitable treatment for all patient groups.

Regulatory Compliance: Meeting legal and ethical standards that require explanations for AI-driven medical decisions, particularly in sensitive and high-stakes scenarios.

By addressing these concerns, XAI can enhance the integration of deep learning models into healthcare, ultimately improving patient outcomes and the overall quality of care. Clinicians need to understand the rationale behind an AI model's recommendations to trust its output and integrate it into their decision-making process. Without explanations, clinicians might hesitate to adopt AI tools, hindering their potential benefits for patient care (McKelvey et al., 2018).

Explainable models can help identify and mitigate potential biases in the training data that could lead to unfair or discriminatory outcomes for certain patient groups. By understanding how the model arrives at its decision, developers can address these biases and ensure fair treatment for all patients (Whittaker, 2019).

Patients have a right to understand the reasoning behind AI-driven diagnoses or treatment recommendations. Explainability can facilitate better communication between patients and healthcare providers, fostering patient trust and engagement in their care (Guerra-Manzanares et al., 2023).

A study (Alghamdi, 2022) employed LIME to explain a deep learning model’s prediction for diabetic retinopathy from retinal fundus photographs. LIME identified the most relevant image regions (e.g., microvascular abnormalities, hemorrhages) influencing the model's decision, aiding ophthalmologists in understanding its reasoning.

A research project by (Jiang et al., 2021) developed an explainable AI system for personalized treatment recommendations for lung cancer. The system combined a deep learning model with SHAP to explain how factors like a patient's genomics and medical

history contributed to the suggested treatment plan. This transparency empowered oncologists to discuss treatment options with patients in a more informed way.

Investigators in a study by (McKelvey et al., 2018) built an explainable machine-learning model to predict hospital readmission risk for heart failure patients. The model used rule-based explanations, highlighting key factors like a patient's medications and lab test results that contributed to the predicted risk. This information helped healthcare providers tailor discharge plans and interventions for patients at higher risk.

These examples showcase how explainable AI can be implemented in various healthcare applications, fostering trust, improving communication, and ultimately leading to better patient care.

Challenges faced by XAI: While Explainable Artificial Intelligence (XAI) holds great promise for healthcare, it faces several challenges that must be addressed to realize its full potential. These challenges, along with future directions for XAI in healthcare applications, include:

Developing Effective Explanations: Creating explanations for complex deep learning models is inherently difficult due to their high dimensionality and non-linear nature. Ensuring that these explanations are both accurate and understandable is a major hurdle.

Tailoring Explanations for Diverse Users: The level of detail and technical complexity required in explanations varies between user groups. Clinicians may need in-depth insights into model mechanics and evidence for predictions, while patients often require simplified and accessible explanations. Striking this balance is challenging but critical.

Regulatory Ambiguity: The regulatory frameworks governing Explainability in AI-driven healthcare systems are still evolving. Guidelines for what constitutes adequate Explainability are often vague, leading to uncertainty in compliance and deployment.

Scalability of XAI Solutions: Implementing explainable systems at scale, particularly in diverse and resource-constrained healthcare settings, remains a logistical and technical challenge.

Bias and Fairness: While XAI can help identify biases in models, ensuring that these biases are effectively mitigated and explained transparently adds another layer of complexity.

Addressing these challenges will be pivotal in making XAI a cornerstone of trustworthy and effective AI applications in healthcare.

2.5 Related Works

Search Strategy: Our initial information retrieval process focused on identifying relevant databases, drawing from both general scientific publishing platforms and domain-specific medical imaging resources. For general academic research, we utilized national repositories such as the National Academic Digital Repository of Ethiopia (NADRE), alongside globally recognized platforms like PubMed, ScienceDirect, arXiv, and databases from prominent academic publishers such as Frontiers, Hindawi, and Taylor & Francis.

To ensure comprehensive coverage of domain-specific content, particularly in medical imaging and related fields, we also explored specialized resources including ResearchGate, the ACM Digital Library (targeting computer science-related studies), and IEEE Xplore for cutting-edge publications in engineering and technology. These sources provided a robust foundation for retrieving highly relevant and diverse publications to support our research objectives.

To streamline the literature review process, we developed a search string comprising keywords closely aligned with our research topic. Examples of these keywords include "interpretable deep learning," "lung cancer detection," "lung cancer classification," "CT scans," “Early stage cancer classification” and "explainable AI (XAI)." Boolean operators such as AND, OR, and NOT were incorporated into the search string to refine and target the retrieval of the most relevant articles.

In addition to traditional database searches, we leveraged Research Rabbit, a web-based application specifically designed to optimize literature review for researchers, including students and academics. Research Rabbit functions as a citation-based literature mapping tool, utilizing citations and reference networks to establish connections between research articles. This approach significantly enhances the exploration of related works, enabling a more efficient and comprehensive review of relevant literature. The combination of well-constructed search strings and advanced tools like Research Rabbit ensured a thorough and systematic information retrieval process.

Inclusion/Exclusion Criteria: We established clear criteria for selecting relevant articles for our review. This involved focusing on recent advancements (within the past five years),

ensuring articles were published in peer-reviewed journals or reputable conference proceedings, selecting studies that directly address issues related to our study, and considering only studies written in English. These criteria helped us ensure that the literature we reviewed was both current and of high quality, providing a solid foundation for our research.

Selection Process: We employed a two-stage article screening process, consisting of title and abstract screening followed by full-text evaluation. *Title and Abstract Screening:* In the first stage, the titles and abstracts of the retrieved articles were evaluated. Articles deemed irrelevant to the research topic based on their titles and abstracts were excluded. This initial screening helped to efficiently narrow down the number of articles for further consideration. *Full-Text Evaluation:* Articles that passed the initial title and abstract screening stage proceeded to a more in-depth evaluation. Full-text versions of these articles were obtained and thoroughly examined to assess their alignment with the research question, methodology, and inclusion/exclusion criteria. This comprehensive analysis ensured the selection of high-quality and directly relevant studies for further analysis. By implementing this rigorous selection process, we were able to ensure the relevance and quality of the literature included in our review.

After completing the initial steps, we critically evaluated the selected articles to assess their quality, relevance, and methodological rigor. Key aspects considered included the following: one, *Study Design*: Whether the study was retrospective or prospective. Two, *Dataset Characteristics:* Size, source, and type of CT scans used. Three, *Deep Learning Model Architecture:* The specific architecture employed and any interpretability techniques utilized. Fourth, *Evaluation Metrics:* Measures such as accuracy and interpretability metrics. Five, *Strengths and Limitations:* An analysis of the strengths and limitations of the research presented.

Finally, we used the reference management software, Mendeley, to organize and manage our retrieved articles and references. This ensured a systematic and efficient approach to handling the literature, facilitating a thorough and well-structured review process.

As we have discussed previously, DL algorithms have extensive applications in healthcare, especially in informed clinical decision making which is when patients are able to weigh the pros, disadvantages, limitations, alternatives, and uncertainties of the clinical care they are considering for a certain disease or condition. There are a lot of successful applications of

DL in detecting and classifying lung cancer in terms of performance but most of them are done on 2D images or by converting the CT-Scans to ‘jpeg’ and ‘png’ formats. Some of them include (Sori et al., 2019), (B. C. & K. B., 2023), (Shafi et al., 2022), (Tandon et al., 2022), (Bushara & Kumar, 2022), (Ramana et al., 2022), and many more. Since this research study used 3D CNNs for the raw 3D medical image data we will focus on studies that employed 3D deep learning models, let’s see some of them next.

A research work on Jiang et al., (2022) demonstrated the effectiveness of a deep learning model for accurate and fast lung cancer prediction using CT scans. The model comprised three stages of work. The first one is lung nodule detection, which is identifying potential cancerous nodules. The second is the Benign vs. Malignant classification that is differentiating between Benign and Malignant nodules. The third is false positive reduction, which is filtering out non-cancerous nodules. The researchers employed various network architectures and loss functions for optimal performance. Notably, they introduced "Nodule-Net," a detection network combining U-Net and RPN (Region Proposal Network) for improved accuracy in lung nodule detection.

Causey et al., (2022) proposed "Deep Screener," a deep learning model for predicting lung cancer using volumetric CT scans. It combines CNNs for whole-image analysis and leverages Spatial Pyramid Pooling and 3D convolutions to capture spatial information at various scales, enhancing feature extraction. Trained on carefully curated datasets, Deep Screener achieved promising results on an independent test set, demonstrating its potential as a valuable tool for lung cancer detection and risk prediction.

A study conducted by Essaf et al., (2020) investigated the categorization and identification of CT lung images using convolutional neural networks. It used four different consecutive stages of work steps: (1) Using the database's description file CT images of malignant pulmonary nodules and normal nodules were located; (2) Using the maximum inter-class variance method CT image was Segmented to create a binary image; (3) A suitable CNN was designed; (4) CCN was trained and CT image of malignant lung nodules were classified. This study employed the LIDC dataset and the study's findings indicated that a CNN received the processed CT scans as input for both training and recognition. 95.85% was the average classification accuracy rate following six-fold cross-validation.

Sun et al., (2020) utilized a two-stage approach to classify lung nodules using Generative Adversarial Networks (GAN) and 3D CNN. In the first stage, the model employed transfer learning based on Deep Convolutional Generative Adversarial Networks (DCGAN) to preliminarily classify pulmonary nodules. The GAN model generated a dataset similar to the original data, which was then used to pre-train the nodule detection network based on AlexNet. The pre-trained network is fine-tuned using the original data to obtain a neural network with high accuracy. In the second stage, a 3D CNN was introduced to remove false positives in the classification process. They stated that this two-stage approach leveraged the strengths of GAN for data generation and transfer learning in the first stage, followed by the use of 3D CNN for accurate classification and false positives removal in the second stage. The accuracy achieved in the first stage is 94.50% and the highest accuracy registered because of the removal of false positives is 95.30%.

Bao et al. proposed a simple residual network for lung nodule classification. First, they suggested employing the Self-Attention technique to identify global features. Subsequently, a module called Inception-like was presented to extract multi-scale local features the identified local features were regarded as complementary parts of global features with residual connection. By combining the self-attention mechanism for global feature detection and the inception-like module for multi-scale local feature extraction, the network was able to effectively capture both global and local features essential for accurate lung nodule classification. The network achieved an AUC of 96.07%, indicating its high discriminatory power in distinguishing between benign and malignant lung nodules. The accuracy of the model was reported as 90.45%, and the precision and recall values were 0.8993 and 0.9015, respectively.

Lin et al., (2020) designed 3D CNNs with shortcut connections to improve lung nodules classification. As 3D CNN has many parameters, which leads to low model efficiency, they proposed a VGG-based 3D residual connection network. The VGG+ResCon network utilizes shortcut connections to improve the classification of lung nodules in CT images by addressing the issue of gradient disappearance, simplifying the learning process, and accelerating the convergence rate of the network. Additionally, the use of shortcut connections in the VGG+ResCon network enables the model to mine vertical information from tumor CT images more effectively.

This helps in capturing important spatial features of pulmonary nodules in 3D CT data, leading to improved accuracy in distinguishing benign from malignant nodules. The authors stated that the proposed methodology was evaluated on the LUNA16 dataset, achieving the best recall, precision, specificity, and f1-score of 92.48%, 93.62%, 96.83%, and 93.04%, respectively.

Table 2.1 summarizes the aforementioned works of literature. We included research gaps in each literature even though we focused on the interpretability aspect of these models,

Table 2.1 Summary of Related Works of DL on Lung Cancer

(Author, year)	Problem Investigated	Approach Followed	Results Achieved	Gaps found
(Jiang et al., 2022)	Lung region segmentation; candidate region feature extraction; lung nodule recognition and classification. Nodule net for lung nodule detection (3D U-Net).	LIDC IDRI, and LUNA16	FROC = 0.876 (Free-Response Receiver Operating Characteristic Curve)	▪ No interpretation of the results ▪ The research primarily addresses detection and classification at a single time point, lacking insights into the progression of lung nodules over time.
(Causey et al., 2022)	Deep screener based on spatial pyramid pooling with 3D convolution	LIDC, DSB2017, LUNA16, and NLST	Accuracy = 78.2 AUC = 85.8 AUPRC = 78.8	▪ High False Positive Rates ▪ No interpretation of the results
(Essaf et al., 2020)	The CT scans were segmented using the Otsu method, and the features were extracted using CNN then classified the CT scan images as benign and malignant.	LIDC IDRI	Accuracy = 95.85 AUC = 82.16	▪ No interpretation of the results
(Sun et al., 2020)	Generative Adversarial Networks (GAN) and 3D CNN	LIDS	Accuracy= 94.50	▪ No interpretation of the results
(Bao et al., 2020)	Simple residual network for lung nodule classification. Self-Attention technique to identify global features	LIDC	AUC = 96.07 Accuracy = 90.4 Precision = 89.93 Recall = 90.15	▪ The strength of the model's ▪ No interpretation of the results

(Author, year)	Problem Investigated	Approach Followed	Results Achieved	Gaps found
(Lin et al., 2020)	VGG-based 3D residual connection network, called VGG+ResCon	LUNA16	Accuracy = 93.6 Precision = 93.6 Recall = 92.48 Specificity = 96.83 F1-score = 93.04	▪ No interpretation of the results

The studies reviewed highlight significant advancements in lung nodule detection and classification using deep learning approaches, yet they reveal notable gaps that require further exploration. A common limitation across these studies is the lack of result interpretation, which is crucial for clinical applicability and understanding the decision-making process of the models. Additionally, most of the research focuses on single time-point detection, neglecting the progression and temporal changes of lung nodules, which are vital for early-stage cancer management. High false-positive rates, as observed in some studies, further hinder the reliability of these models in practical settings. Lastly, while various methodologies demonstrate impressive accuracy and performance metrics, the strengths and limitations of these approaches are often insufficiently analyzed, leaving room for improvement in model transparency and robustness. These gaps underscore the need for more comprehensive and clinically interpretable models that incorporate temporal analysis and address reliability issues effectively.

CHAPTER THREE

DESIGN AND METHODS

3.1 Overview

This chapter comprehensively details the materials and methods utilized throughout the deep learning (DL) workflow. It encompasses key stages such as data acquisition, where we gathered the necessary datasets for our study, and pre-processing, which involves cleaning and preparing the data to ensure optimal model performance. The chapter also delves into the design of the interpretable deep learning model architecture, highlighting how we structured the model to enhance its Explainability while maintaining high performance. Furthermore, it covers the training process, outlining the techniques and parameters used to train the model effectively. Lastly, the chapter discusses the evaluation metrics employed to assess the classification performance, ensuring that the model's predictions are both accurate and reliable. Each step in this workflow is meticulously crafted to build a robust and interpretable DL model, capable of delivering significant insights.

3.2 Proposed architecture

This study explored the performance of four 3D CNN architectures:

1) Baseline Model: A simple 3D CNN architecture comprising a series of convolutional layers, max-pooling layers, and fully connected layers. This model served as a baseline for comparison and evaluation.
2) 3D AlexNet-based Model: An adaptation of the well-known 2D AlexNet architecture to the 3D domain. This model incorporated 3D convolutional and pooling layers, mimicking the sequential structure of the original AlexNet.
3) Proposed 3D CNN Model:

Architecture: **Input;** 3D CT scan patches (e.g., 64x64x64 voxels). **Convolutional Layers;** Multiple 3D convolutional layers with varying filter sizes (e.g., 3x3x3, 5x5x5) and depths to extract features at different scales. **Inception Modules;** Incorporate parallel convolutional layers with different filter sizes (e.g., 1x1x1, 3x3x3, and 5x5x5) to extract features at multiple scales simultaneously. **Pooling Layers;** 3D max-pooling layers to downsample feature maps and reduce computational complexity. **Dropout Layers** to prevent overfitting and improve model generalization. **Fully Connected Layers;** Multiple

fully connected layers classify the extracted features into benign or malignant categories. **Rationale:** Inception modules enhance feature extraction by capturing information at multiple scales, improving the model's ability to detect subtle variations in nodule characteristics. Deeper convolutional layers allow the model to learn more complex and abstract features from the input data, and Dropout layers help to prevent overfitting by randomly dropping out neurons during training, reducing the model's reliance on specific neurons and improving its generalization ability.

4) 3D CNN with CBAM: This model incorporates the Convolutional Block Attention Module (CBAM) into the proposed 3D CNN architecture. CBAM consists of two attention modules: **Channel Attention Module:** Selectively emphasizes the most informative feature channels. **Spatial Attention Module:** Selectively emphasizes the most informative spatial locations within the feature maps. **By integrating CBAM**, the model can dynamically refine feature maps, focusing on the most relevant information for classification and potentially improving performance.

3.3 Materials

Considering that the core components of AI models include the platform for executing our code, the data used for training, and the diverse libraries and packages employed for various operations, we explore these aspects in the following subsection. This examination covers the computational environment and hardware specifications, the sources and preprocessing steps of the training data, and the specific software tools and frameworks that facilitate the implementation and optimization of our models. Understanding these fundamental elements is crucial for developing robust and efficient AI systems.

3.3.1 Platform

This study was conducted entirely on Kaggle and Google Colab, both cloud-based platforms that enable running Python code directly within a web browser. These platforms are particularly useful for machine learning, data analysis, and other computationally intensive tasks, as they eliminate the need for a high-performance local machine. One of their standout features is access to hardware accelerators, specialized processors designed to significantly enhance computation speed. Both Kaggle and Colab offer two primary types of hardware accelerators:

1) Graphics Processing Units (GPUs): GPUs excel at parallel processing, making them ideal for tasks such as training deep learning models.
    - Kaggle: Provides GPU options like GPU T4 x 2 and GPU P100, along with a Tensor Processing Unit (TPU) type called TPU VM v3-8.
    - Colab: Offers a variety of GPUs, including Tesla A100s, V100s, and T4s. Each GPU type has unique capabilities and performance levels.
2) Tensor Processing Units (TPUs): TPUs are custom-designed by Google specifically for machine learning tasks. They often outperform GPUs for particular workloads but may not be suitable for all types of computations.

The availability of these accelerators makes both platforms invaluable for efficiently handling resource-intensive processes in machine learning and related fields.

3.3.2 Data

Data can be categorized as either local or public based on its scope, accessibility, and origin. Public data refers to information that is freely accessible and typically provided by government agencies or public institutions for general use. In contrast, local data pertains to information specific to a particular geographic area, often gathered and maintained by local authorities or organizations for regional purposes.

Only 20% of lung cancer cases are reported in low- and middle-income countries. In Ethiopia, approximately 1.5% of all cancer cases involve the lung. However, the absence of a nationwide cancer registry in Ethiopia means that precise data on clinical history, histopathology, molecular characteristics, and risk factors for lung cancer remain unavailable (Gebremariam et al., 2021).

We chose to leverage publicly available datasets curated for research purposes. Among the datasets we explored were LIDC-IDRI, LUNA16, Kaggle Data Science Bowl, and LUNA22-ISMI. After conducting thorough exploratory data analysis on these and other additional datasets, we selected the most recent and computationally inexpensive dataset for our study: the LUNA22-ISMI dataset. This selection was based on its suitability for our research needs and its ability to provide relevant and high-quality data for our deep learning model.

The LUNA22-ISMI (Lung Nodule Analysis 2022 - Intelligent Systems in Medical Imaging) challenge was an educational initiative centered on classifying lung nodules in chest CT

scans. Its primary goal was to offer a platform for researchers and students to develop and test their algorithms for predicting nodule malignancy and classifying nodule types. Participants were provided with a dataset containing 1.7 GB of compressed files with 3D patches of nodules (Venkadesh & Jacobs, 2022). This challenge aimed to advance the field of medical imaging by encouraging innovative approaches and enhancing the understanding of lung nodule classification.

The LUNA22-ISMI dataset consists of 1176 lung nodule CT scan patches derived from the LIDC-IDRI dataset, specifically curated for the educational LUNA22-ISMI challenge. This dataset includes 3D patches of nodules, each sized 128 x 128 x 64 in x, y, and z dimensions. The nodules are consistently positioned in the center of each 3D patch. The dataset incorporates labels from at least three out of four radiologists, following the LUNA16 criteria, resulting in a total of 1186 labeled nodules. Additionally, the dataset provides radiologist scores for nodule type and malignancy, offering valuable information for developing and testing algorithms in nodule classification and malignancy prediction (Venkadesh & Jacobs, 2022).

The images are formatted in a compressed "nifti" format as ".nii.gz". The labels for each nodule can be found in the given Numpy file named "*LIDC-IDRI_1176.npy"*.

3.3.3 Software

To develop deep learning models, specialized software libraries and frameworks are essential. We utilized various software tools for data pre-processing, deep learning model development, and model interpretation. Additionally, we employed ITK-SNAP for data exploration and medical image viewing.

Data exploration and Data pre-processing:

ITK-SNAP: is an interactive software application that allows users to manually identify anatomical regions of interest and navigate through three-dimensional medical images, such as CT scans. For the data preprocessing step, we utilized various Python libraries for exploratory data analysis of different datasets. Let's take a look at them one by one.

Pydicom: Pydicom is a free, open-source Python library designed to handle DICOM (Digital Imaging and Communications in Medicine) files. It offers a wide range of functionalities, making it a valuable tool for medical imaging tasks. Key features include **Reading DICOM**

Files: PyDicom enables access to and parsing metadata from medical images, including patient information, acquisition parameters, and modality details. **Writing DICOM Files:** The library allows users to create and modify DICOM files, facilitating data sharing and archiving. **Processing Medical Images:** PyDicom provides basic tools for image processing on DICOM data, supporting further analysis (Mason et al., 2023).

NiBabel: Nibabel is a free and open-source Python library that provides tools for working with various neuroimaging data formats. It serves as a bridge between different neuroimaging file formats and analysis tools, streamlining the workflow. Key functionalities of Nibabel include:

Loading and saving neuroimaging data: Nibabel supports a wide range of neuroimaging file formats, such as fMRI (NIFTI, MGH), EEG (Brain Vision), and PET (ANALYZE). Data can be easily loaded from these formats for further analysis in Python. **Data manipulation**: Nibabel offers basic functionalities for manipulating neuroimaging data, such as cropping, masking, and reorientation. **Interaction with other neuroimaging libraries**: Nibabel seamlessly integrates with other popular neuroimaging libraries in Python, enabling a more comprehensive analysis pipeline (Brett et al., 2024). **dicom2nifti:** Python library for converting DICOM to Nifti format. dicom2nifti is a free and open-source Python library created to facilitate the conversion of DICOM files to the Nifti format. Additionally, we have utilized various other Python packages and modules for different computational tasks.

OS: The module offers a robust set of functions for interacting with the operating system. It enables Python programs to handle tasks such as file and directory management, process control, and environment variables. **Shuttle:** The shuttle module in Python provides a collection of high-level functions for working with files and collections of files. It offers functionalities that go beyond the basic file operations available in the built-in OS module, simplifying common tasks like copying, moving, and removing files and directories. **Numpy,** short for Numerical Python, is a fundamental library for scientific computing in Python. It provides a powerful set of tools for working with multidimensional arrays and matrices, along with mathematical functions for efficient numerical data operations. **Pandas:** is a powerful and popular library in Python that is specifically designed for data preprocessing and manipulation. It excels at working with tabular data, similar to spreadsheets or SQL tables, making it a reliable and useful tool for data scientists and analysts.

Scipy, pronounced "Sigh-pi," is another cornerstone library in the scientific Python ecosystem. Building on top of NumPy, SciPy extends its functionality by providing a vast collection of algorithms and tools for various scientific computing tasks. We used it for data augmentation and other Numpy array operations. **Matplotlib** is a fundamental library for creating static, animated, and interactive visualizations in Python. It offers a versatile and user-friendly toolkit for generating various plot types to explore and communicate data insights effectively.

Deep learning model development:

TensorFlow: It is a free and open-source software library that was developed by Google for numerical computation and large-scale learning using deep learning. It provides a flexible architecture for efficiently defining, training, evaluating, and deploying machine learning models. Key functionalities of TensorFlow include **Dataflow Programming**; TensorFlow utilizes dataflow graphs to represent computations. These graphs define the relationships between data elements and the operations performed on them. This approach facilitates efficient execution on various hardware platforms, including CPUs, GPUs, and TPUs. **Automatic Differentiation**: TensorFlow offers automatic differentiation capabilities, which are crucial for training machine learning models. It calculates the gradients of a function concerning its inputs, allowing optimization algorithms to effectively adjust the model's parameters. **Tensor Operations**: A core concept in TensorFlow is the "tensor," a multidimensional array of numerical data. TensorFlow provides a rich set of operations for manipulating and transforming tensors, forming the building blocks for machine learning algorithms. **Pre-built Libraries and Tools**: TensorFlow offers a wide range of pre-built libraries for specific tasks like natural language processing, computer vision, and recommender systems. Additionally, it provides tools for model deployment, serving, and visualization (TensorFlow, 2019). **Keras**: Keras is a high-level API designed to simplify the creation of deep learning models, operating atop frameworks like TensorFlow. It provides a user-friendly interface for building, training, and deploying deep neural networks. Key functionalities of Keras include **Model Building Blocks**: Keras offers pre-built layers for deep learning models, such as convolutional layers (CNNs) for image processing, recurrent layers (LSTMs) for sequence data, and dense layers for fully connected networks. **Sequential vs. Functional API**: Keras supports two main approaches for constructing deep learning models:

Sequential API: Ideal for building linear layer stacks, often used in basic deep learning models. **Functional API**: Allows for complex model architectures with branching, merging, and custom layers, offering more control for advanced tasks.

Loss Functions and Optimizers: Keras includes various pre-defined loss functions (e.g., mean squared error, categorical cross-entropy) for evaluating model performance during training and optimizers (e.g., gradient descent variants) for adjusting model parameters based on the loss function. **Ease of Use:** Compared to directly using TensorFlow, Keras makes deep learning development easier with its intuitive API and focus on essential model-building components (Keras, 2019).

3.4 Methods

Our methodological approach comprised four fundamental stages, each of which is elaborated upon in the subsequent subsections.

3.4.1 Exploratory Data Analysis

The initial phase of our study involved conducting a thorough exploratory data analysis (EDA) on potential datasets. This process began with data acquisition and focused on examining the characteristics of the data, including the specific attributes of medical imaging data and the labeling methodologies used. After visualizing the CT scan images with ITK Snap software, we utilized the Nibabel library to access the pixel data and retrieve the header information. Below is an example of a typical histogram representing the distribution of pixel values, also known as Hounsfield units (HU), for a single 3D patch of a nodule.

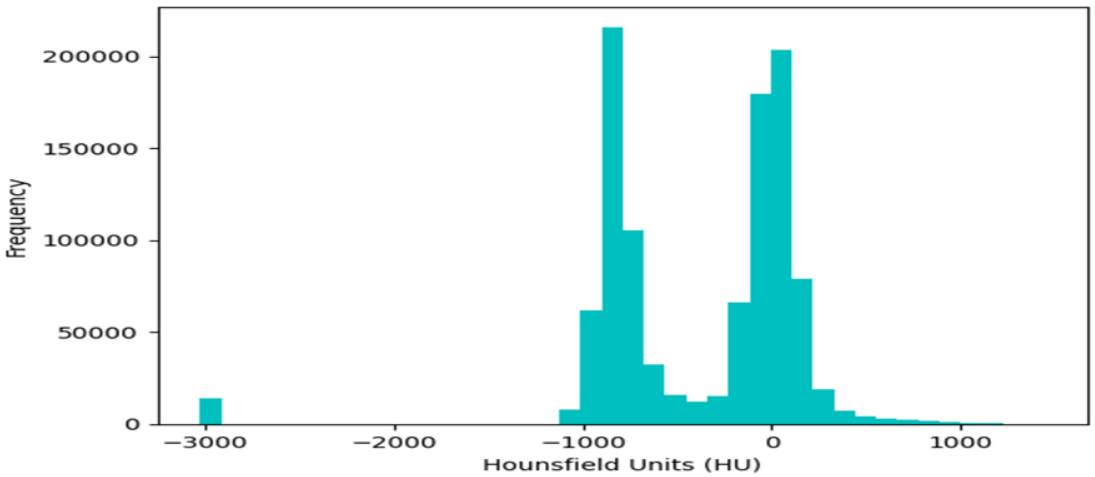


Figure 3.1 Histograms of Pixel values of a single 3D patch of nodule.

As we can see from the above figure, the pixel values for this sample CT scan range from more than +1000 to less than -3000, and each patch has a size of 128x128x64 in the x, y, and z directions. After going through all pixel values of each 3D nodule patch, we have found that the pixel values range from (-3,024.0) to (+6,054.0).

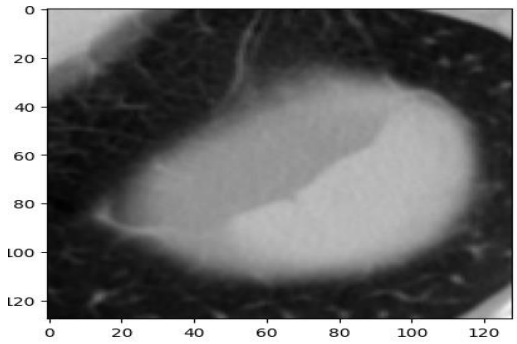


Figure 3.2 A CT scan slice of a sample 3D patch nodule

The labels were obtained from the "LIDC-IDRI_1176.npy" file. This file stores information for each nodule in a structured format using Python dictionaries. The file contains a NumPy array, where each element represents a single nodule. Each nodule dictionary includes the following key-value pairs relevant for label extraction and preprocessing:

```
{
'SeriesInstanceUID':
'1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860',
 'VoxelCoordX': 45,
 'VoxelCoordY': 211,
 'VoxelCoordZ': 77,
 'Diameter': [6.97167141, 6.97167141, 7.34878692, 5.94228451],
 'Texture': [5, 5, 5, 5],
 'Malignancy': [4, 2, 4, 2],
 'Calcification': [6, 6, 6, 6],
 'Filename':
'1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860_45_21
1_77_0000.nii.gz'
}
```

Where, *SeriesInstanceUID:* A distinctive identifier uniquely assigned to a specific CT scan series for each patient, ensuring accurate tracking and referencing of the imaging data, *VoxelCoordX*, *VoxelCoordY*, *VoxelCoordZ:* These denote the 3D coordinates pinpointing the exact center of the nodule within the CT scan volume, facilitating precise spatial localization for further analysis, *Diameter:* An array containing measurements that specify

the nodule's diameter across different axes, providing insights into its size and potential growth patterns, which are critical for assessing malignancy risks, *Texture (1–5):* A categorical variable that describes the internal radiographic solidity of the nodule, ranging from non-solid (1) to fully solid (5). This attribute is crucial in distinguishing between various types of nodules and aiding radiologists in determining potential malignancy, *Malignancy (1–5):* A categorical variable reflecting the estimated likelihood of malignancy, assuming the patient is a 60-year-old male smoker. This scale ranges from 1, indicating a very low probability, to 5, which represents a high suspicion of cancer. It serves as a vital input for risk stratification and clinical decision-making, *Calcification (1–6):* Another categorical variable indicating the presence and degree of calcification within the nodule. Calcification patterns can provide critical clues about the nature of the nodule, whether benign or malignant, and guide further diagnostic steps (Venkadesh & Jacobs, 2022).

This process involves transforming the original five-class malignancy labels into a binary classification problem, simplifying the model's task. Here's a breakdown of the steps:

Label Grouping: Benign: Labels 1 and 2 are combined to represent nodules with low malignancy suspicion. *Malignant:* Labels 3, 4, and 5 are combined to represent nodules with high malignancy suspicion, *Median Score Calculation:* A median score is calculated from the "malignancy" key-value pair to represent the overall malignancy assessment, *Binarization:* Nodules with a median score below three are classified as benign. Nodules with a median score of 3 or above are classified as malignant. This approach aims to consolidate potentially subjective assessments from multiple radiologists into a single, representative label for each nodule. By simplifying the classification problem, the model can focus on the key distinction between benign and malignant nodules.

However, this classification process revealed a class imbalance within the dataset. Out of the 1,176 labeled 3D nodule patches, 380 were categorized as benign, and 796 belonged to the malignant class. To address this imbalance and ensure the model learns effectively from both classes, appropriate data augmentation techniques were implemented during the preprocessing stage.

3.4.2 Data Preprocessing

In this stage, we have done data splitting and employed some preprocessing techniques on the dataset. To prevent data leakage, we split the dataset before augmentation.

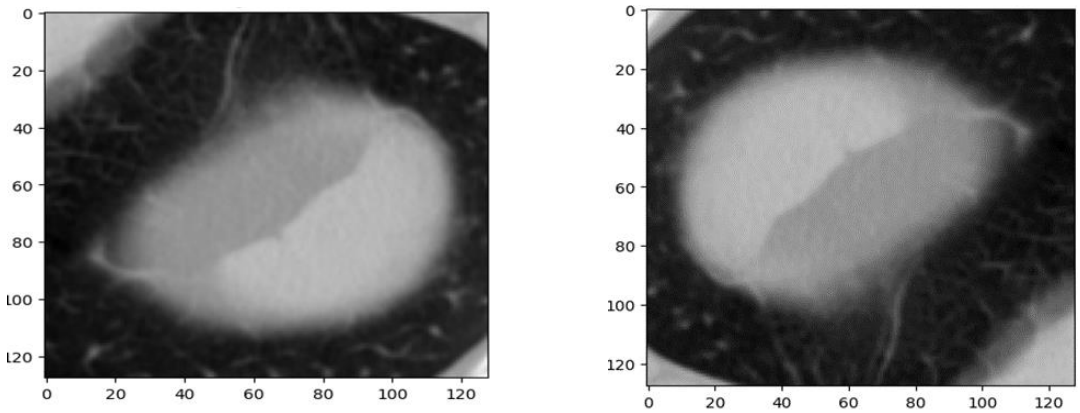
Data Splitting: we divided the dataset into 80% for training, 10% for validation, and 10% for testing to ensure proper model evaluation. The training set contained 304 benign and 637 malignant 3D CT scan patches of nodules. The validation set consisted of 38 benign and 79 malignant 3D CT scan patches used for testing the model on unseen data during training. Finally, the test set included 38 benign and 80 malignant 3D CT scan patches used to assess the model's performance on completely unseen data after training. This split ensured that the model was trained on a representative data portion, evaluated to avoid overfitting, and tested for a comprehensive performance review.

Data Augmentation: Data Augmentation was used to balance the original imbalanced dataset and create an equal number of 3D CT scan patches for both classes. Additionally, to improve the model's generalizability and prevent overfitting during training, data augmentation techniques were employed on the training set. This process artificially increases the training data by introducing controlled variations of existing samples, enhancing the model's ability to learn features robust to these variations. Several augmentation techniques were implemented, including Geometric Transformations: Flipping, Scaling, Translation, and Rotation. Noise Addition: Noise addition, Brightness variation, Contrast variation, Elastic deformation. Filtering: Median filtering. A combination of two or three techniques was also applied. All of these techniques were implemented using the SciPy library.

But even if we employed up to ten augmentation techniques, we only used the data that we have gotten from only seven techniques. This is done because of a shortage of RAM during training. We encountered RAM out of memory or "your notebook tried to allocate more memory than is available. It has restarted." during training. The seven augmentation techniques used are as follows with their visualizations.

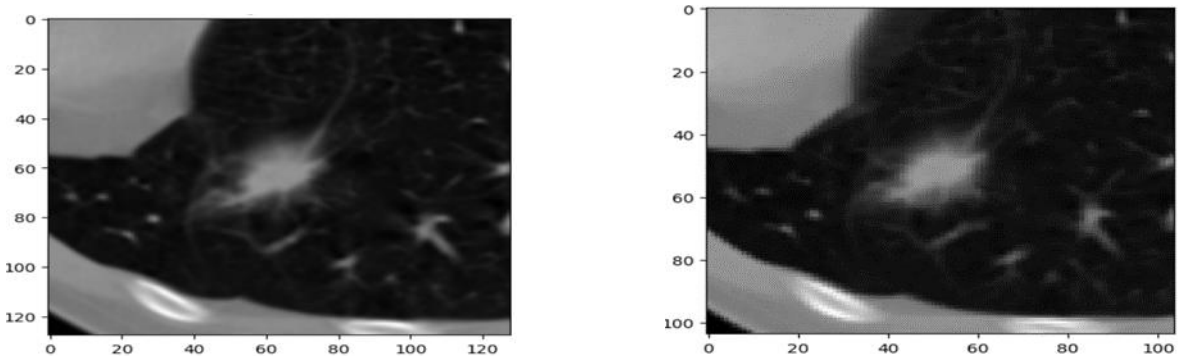
Flipping, a type of geometric transformation, involves flipping 3D nodule patches along two randomly selected axes out of the three dimensions. This mimics potential variations in nodule orientation within lung scans. Scaling changes the image size by a random factor between 0.8 and 1.2 along each axis independently. Translation shifts the image by a random amount within the range of -10 to 10 pixels along each axis.

Noise addition: Adds Gaussian noise to the image with a random standard deviation between 0.05 and 0.2. Contrast adjustment: Applies a shear transformation to the image with random shear factors between -0.2 and 0.2 along each axis. Brightness adjustment: Adjusts the image's brightness by a random factor, shifting the intensity values of a 3D image by -30 to 20 to make it brighter or darker. Elastic distortion (spline filtering): Introduces random, localized deformations to the 3D nodule patches, mimicking slight anatomical variations or scanner imperfections that might occur in real-world CT scans, enhancing the model's resilience to such artifacts. The SciPy library offered the essential functionalities for applying these data augmentation techniques to the pixel intensity arrays of the 3D nodule patches. Figures 3.3 and 3.4 demonstrate the impact of these augmentation techniques on sample nodule slices, showing how the original pixel intensity values were altered.



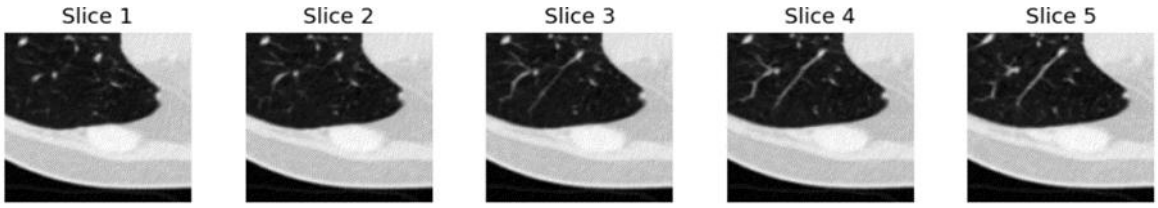
Original Flipped

(a) Randomly Flipped nodule patch CT scan slice

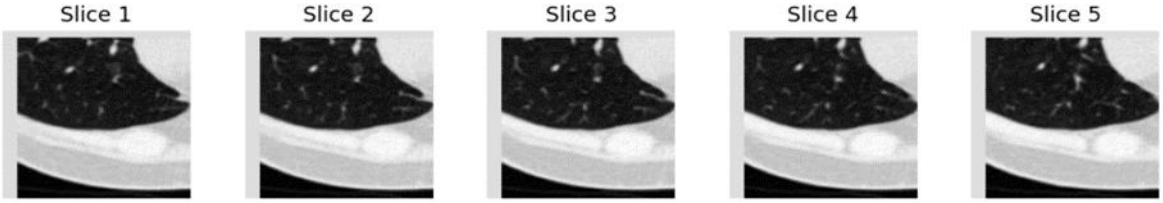


Original Scaled

(b) Randomly Scaled nodule patch CT scan slices and the size changed to (104, 104, 52)



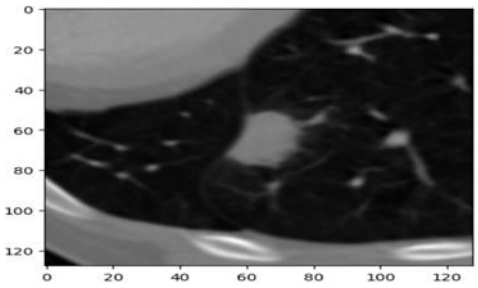
Original



Translated

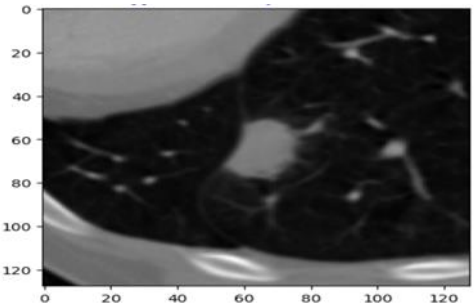
(c) Randomly translated nodule patch CT scan slices

Figure 3.3 The effect of Flipping, Scaling, and Translation on a sample CT scan of a nodule Patch from the Luna22 dataset.



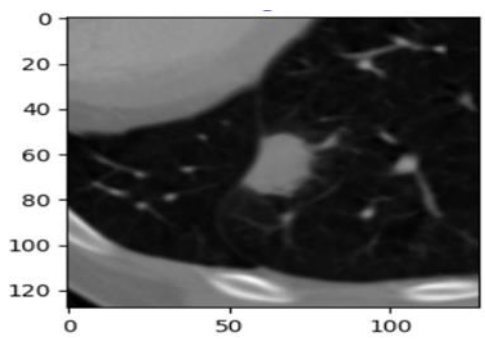
```
[[[ 4.300e+01 -2.700e+01  3.000e+00 ...  1.530e+02  1.280e+02  2.160e+02]
  [ 2.700e+01  9.000e+00  3.600e+01 ...  1.840e+02  1.870e+02  2.460e+02]
  [ 5.800e+01  5.100e+01  2.200e+01 ...  2.000e+02  2.420e+02  2.660e+02]
  ...
  [-5.800e+01 -7.200e+01 -1.220e+02 ... -8.840e+02 -9.000e+02 -8.740e+02]
  [-8.300e+01 -1.000e+00 -4.800e+01 ... -9.050e+02 -9.090e+02 -8.540e+02]
  [-2.400e+01  6.300e+01 -5.900e+01 ... -9.170e+02 -9.210e+02 -8.740e+02]]
```

(a) Original nodule CT scan slice and its pixel intensity values



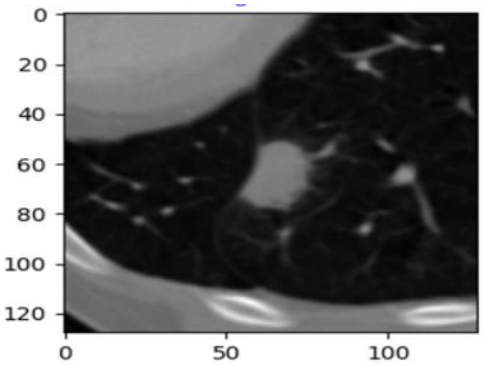
```
[[[ 4.29941813e+01 -2.69971504e+01  2.93395567e+00 ...  1.52987623e+02
    1.27847917e+02  2.15758995e+02]
  [ 2.71642656e+01  8.98332893e+00  3.59930802e+01 ...  1.83878053e+02
    1.86964549e+02  2.46127793e+02]
  [ 5.79288367e+01  5.08976397e+01  2.20644403e+01 ...  2.00137844e+02
    2.42068730e+02  2.66043488e+02]
  ...
  [-5.81856579e+01 -7.21031782e+01 -1.21897311e+02 ... -8.83966156e+02
   -8.99973692e+02 -8.74031995e+02]
  [-8.29618448e+01 -9.21269176e-01 -4.80088576e+01 ... -9.04899202e+02
   -9.09079907e+02 -8.54046207e+02]
  [-2.41342325e+01  6.30429400e+01 -5.90332393e+01 ... -9.17003015e+02
   -9.20954246e+02 -8.74112107e+02]]
```

(b) Random Gaussian noise added CT scan slice and its pixel intensity values



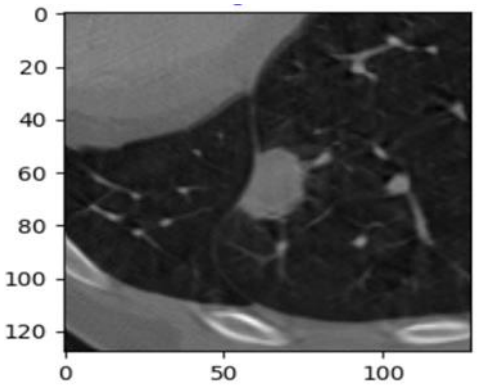
```
[[-3.24269492e+00 -2.26988644e+01  2.37797627e+01 ...  2.21584153e+02
    2.57253797e+02  2.64820085e+02]
  [ 3.78314407e+01 -3.67505424e+01  1.08089831e+00 ...  2.54011102e+02
    2.81033560e+02  2.86438051e+02]
  [ 3.56696441e+01 -2.48606610e+01  1.08089831e+01 ...  2.63739187e+02
    2.73467271e+02  2.84276254e+02]
  ...
  [-3.13460509e+01 -2.16179661e+01 -7.67437797e+01 ... -9.92264645e+02
   -9.99830933e+02 -9.50109611e+02]
  [-7.99864746e+01 -9.29572543e+01 -7.45819831e+01 ... -9.86860153e+02
   -1.01496351e+03 -9.77132069e+02]
  [-9.72808475e+01 -7.13392882e+01 -1.10251627e+02 ... -9.84698357e+02
   -9.95507340e+02 -1.01172081e+03]]
```

(c) Randomly Contrast adjusted CT scan slice and its pixel intensity values




```
[[[ 2.06897640e+01 -4.93102360e+01 -1.93102360e+01 ...  1.30689764e+02
    1.05689764e+02  1.93689764e+02]
  [ 4.68976402e+00 -1.33102360e+01  1.36897640e+01 ...  1.61689764e+02
    1.64689764e+02  2.23689764e+02]
  [ 3.56897640e+01  2.86897640e+01 -3.10235980e-01 ...  1.77689764e+02
    2.19689764e+02  2.43689764e+02]
  ...
  [-8.03102360e+01 -9.43102360e+01 -1.44310236e+02 ... -9.06310236e+02
   -9.22310236e+02 -8.96310236e+02]
  [-1.05310236e+02 -2.33102360e+01 -7.03102360e+01 ... -9.27310236e+02
   -9.31310236e+02 -8.76310236e+02]
  [-4.63102360e+01  4.06897640e+01 -8.13102360e+01 ... -9.39310236e+02
   -9.43310236e+02 -8.96310236e+02]]
```

(d) Randomly Brightness adjusted CT scan slice and its pixel intensity values




```
[[[ 3.81289466e+02 -2.28044292e+02 -1.29221900e+01 ...  1.21109220e+02
   -3.42336033e+00  2.18931214e+02]
  [ 3.14835091e+01  2.53331518e+01  1.41466328e+02 ...  1.72274462e+02
    1.41136998e+02  2.78112485e+02]
  [ 1.14301848e+01  1.36690634e+02  6.93935864e+01 ...  2.02265264e+02
    2.46698410e+02  2.93438078e+02]
  ...
  [-1.62561249e+02 -1.97911354e+02 -2.06231447e+02 ... -8.21561605e+02
   -9.05573576e+02 -8.86345983e+02]
  [-3.01264680e+02  1.92073593e+02  3.98830152e+01 ... -8.29082909e+02
   -9.45954483e+02 -7.49539883e+02]
  [-3.91628431e+01  1.49566897e+02 -1.17856453e+02 ... -9.17723828e+02
   -9.50644795e+02 -8.16071070e+02]]
```

(e) Spline-filtered or elastic distorted CT scan slice and its pixel intensity values

Figure 3.4 The effect of noise addition, contrast and brightness adjustment, and elastic distortion for augmentation (The pixel intensity values are shown to depict how the 3D image is changed).

Following data augmentation, we applied two essential preprocessing steps commonly used for chest CT scans before training a deep-learning model:

Windowing: Building on the established understanding of Hounsfield Units (HU) for representing CT scan voxel intensities (as discussed in Chapter 2), we applied specific data preprocessing steps to the lung nodule patches in this dataset. The raw HU values in our dataset range from -3024.0 to +6054.0. To focus on the lung parenchyma, the region of interest for nodule classification, we clipped the voxel intensities based on typical HU ranges for different tissue types in chest CT scans.

We employed a clipping threshold of -1000 HU for air (to filter out background noise) and +400 HU for bone (to exclude intensities that could represent unrelated anatomical structures or image artifacts). This clipping process ensures that Voxels with intensities above the bone threshold are set to 400 HU. Voxels with intensities below the air threshold are set to -1000 HU. These adjustments allow us to focus on the relevant data for lung nodule classification, improving the model's ability to interpret the CT scan images effectively.

Normalization (Min-Max Scaling): Following data augmentation, the other crucial technique used was normalization. This step is applied to the lung nodule patches before feeding them into the deep-learning model for training. Normalization aims to scale the pixel intensity values within a specific range, promoting numerical stability and improving the learning process. In this work, we employ a min-max scaling technique combined with zero-centering. The min-max scaling process is achieved using the following formula:

$$Volume_{normalize} = (Volume - HU_{Min})/(HU_{Max} - HU_{Min}) \quad 3.1$$

Where: volume represents the original voxel intensity array of the lung nodule patch. HU_{min} and HU_{max} denote the minimum and maximum HU values encountered within the 3D patch, respectively. This formula effectively scales each voxel intensity within the original range (HU_{min} to HU_{max}) to a new range between -1 and 1. This compressed range facilitates the learning process for the deep learning model by ensuring all features reside on a similar scale. By incorporating these data augmentation and preprocessing techniques, we aimed to create a more robust and generalizable model for lung nodule classification.

Following the preprocessing steps, the 3D lung nodule patches undergo a final resizing stage to match the input requirements of the chosen deep-learning model architecture. This resizing process ensures compatibility between the data and the model's expected input dimensions.

We employ a technique known as spline interpolated zoom (SIZ) for resizing the 3D patches. SIZ leverages spline interpolation, a mathematical method for creating smooth curves or surfaces that pass through a set of data points (Zunair et al., 2020). This approach offers advantages over simpler techniques like nearest neighbor interpolation, which can introduce artifacts during the resizing process. By using SIZ, we can preserve the fine details and structural information within the 3D patches, leading to more accurate and robust model performance.

After SIZ resizing, the 3D image voxel arrays are transformed into a format that is compatible with deep learning models. In this study, we utilize NumPy arrays to represent the preprocessed and resized 3D patches. Each voxel intensity within these arrays is encoded as a single numerical value, ensuring a standardized representation for model input. Additionally, binary labels are assigned to each patch, where "0" denotes benign nodules and "1" indicates malignant nodules. This clear binary labeling scheme is essential for facilitating the training process in our classification task, enabling the model to effectively distinguish between benign and malignant cases.

Visualization: During the preprocessing stage, data visualization techniques were utilized to gain valuable insights into the data distribution, detect potential anomalies, and evaluate the effectiveness of the preprocessing steps. These visualizations included Histograms To analyze the distribution of voxel intensities (Hounsfield Units) before and after applying clipping thresholds; 2D Slice Views To visually inspect individual slices from 3D lung nodule patches, ensuring that preprocessing steps such as normalization and augmentation were applied correctly; 3D Volume Renderings: To observe the spatial structure and placement of nodules within the CT patches, helping to identify any inconsistencies or artifacts; Comparison Charts: To compare data characteristics before and after preprocessing, such as changes in intensity ranges or augmented variations; These visualization efforts ensured the integrity of the data and validated the preprocessing pipeline's contribution to model readiness.

3.4.3 Deep Learning Model Development and Evaluation

This section addresses the development, training, and evaluation of a deep-learning model for lung nodule classification in chest CT scans. Deep Neural Networks (DNNs) have proven to be powerful tools for automated image analysis, showing impressive performance in various classification tasks. Given the three-dimensional nature of chest CT data, we chose to use 3D Convolutional Neural Networks (3D CNNs) to utilize the spatial information within the volumetric data. Unlike 2D CNNs, 3D CNNs employ 3D kernels or filters to capture spatial relationships across all three dimensions (depth, width, and height).

While 3D CNNs offer better performance in processing 3D data, they also come with increased computational complexity. The figure below illustrates 3D convolution.

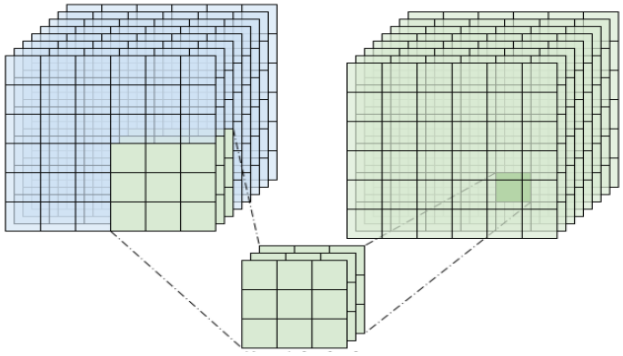


Figure 3.5 3D Convolution (https://www.researchgate.net).

Baseline Model Architecture and Rationale

To create a baseline for comparison and assess the feasibility of using simpler architectures, we implemented a straightforward 3D CNN model inspired by the work of Zunair et al. (2020), who explored methods for processing CT scans with 3D CNNs for tuberculosis prediction. Additionally, we examined the effectiveness of converting well-known 2D CNN architectures, such as AlexNet and VGG, into 3D CNN variants. Detailed configurations of these models and their training processes are discussed in the next chapter.

The rationale for this model selection strategy is twofold. First, due to the computational demands of 3D CNNs, we evaluated the performance of a relatively simple 3D CNN architecture as a baseline. Second, we investigated the potential of adapting successful 2D CNN architectures, like AlexNet and VGG, into the 3D domain for lung nodule classification.

Model Evaluation

Following the development and training of the deep learning models, a comprehensive evaluation process was conducted to assess their generalizability and performance on unseen data. This evaluation utilized a separate testing dataset that the models had not been exposed to during training. By relying on this independent dataset, we ensured an unbiased assessment of the model's capability to classify lung nodules accurately, reflecting its potential effectiveness in real-world scenarios. This rigorous evaluation process is critical for validating the model's reliability and robustness.

To comprehensively evaluate the models' performance, we employed a range of established performance metrics commonly used in binary classification tasks. The following were our performance metrics:

Accuracy: This is the most fundamental metric, and measures the overall proportion of correct predictions made by the model. It is calculated as the sum of true positives (TP) and true negatives (TN) divided by the total number of samples. While offering a basic overview of model performance, accuracy can be misleading in imbalanced datasets.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad 3.2$$

Precision: measures the model's ability to correctly identify true positive cases. For lung nodule classification, it represents the proportion of nodules classified as malignant by the model that are confirmed as malignant by radiologists. A high precision value indicates a low false positive rate, signifying the model's effectiveness in minimizing misclassification of benign nodules as malignant. This metric is particularly important in reducing unnecessary anxiety and interventions for patients.

$$Precision = TP/(TP + FP) \quad 3.3$$

Recall (Sensitivity): Recall, also referred to as sensitivity, highlights the model's capability to detect all actual positive cases, specifically malignant nodules in lung nodule classification. It is computed as the ratio of true positives identified by the model to the total number of actual positive cases in the testing data. As recall value increases false negative rate decreases, signifying the model's effectiveness in identifying most malignant nodules, which is critical for ensuring timely and accurate diagnosis in clinical settings.

$$Recall = TP/(TP + FN) \quad 3.4$$

F1-Score: The F1-score is the harmonic mean of recall and precision, offering a balance of a model's performance by combining recall and precision into a single metric (see equation 3.5). A high F1 score indicates a model that is effective at identifying true positives while minimizing false positives.

$$F1 - Score = 2 * (Precision * Recall)/(Precision + Recall) \quad 3.5$$

AUC-ROC (Area under the Receiver Operating Characteristic Curve): The AUC-ROC curve is a graphical plot of a model's performance at the given classification thresholds. The

AUC-ROC score measures the model's overall capability to differentiate among positive classes.

Confusion Matrix: This is a table that provides detailed information about the total number of true positives, false positives, true negatives, and false negatives. The confusion matrix enables to calculation of the performance of the classification model on a test data set.

By using such an evaluation strategy, we test the performance of deep-learning models for lung nodule classification. These insights can then be used to refine the model architecture, adjust training parameters, or improve data preprocessing techniques to achieve optimal performance in this critical medical application.

3.4.4 Deep Learning Model Interpretation Using Grad-CAM:

At this stage, we applied Gradient-weighted Class Activation Mapping (Grad-CAM) to interpret the predictions made by our DL model in lung nodule classification. Grad-CAM is a visualization technique designed to provide insights into the regions within a lung nodule image that most significantly influence the model's classification decision (malignant vs. benign). This interpretability enhances the model's transparency and allows medical professionals to better understand the reasoning behind the predictions.

The details of the implementation are elaborated upon in the subsequent chapter. A general block diagram illustrating the methodology for implementing the proposed model is shown in Figure 3.7.

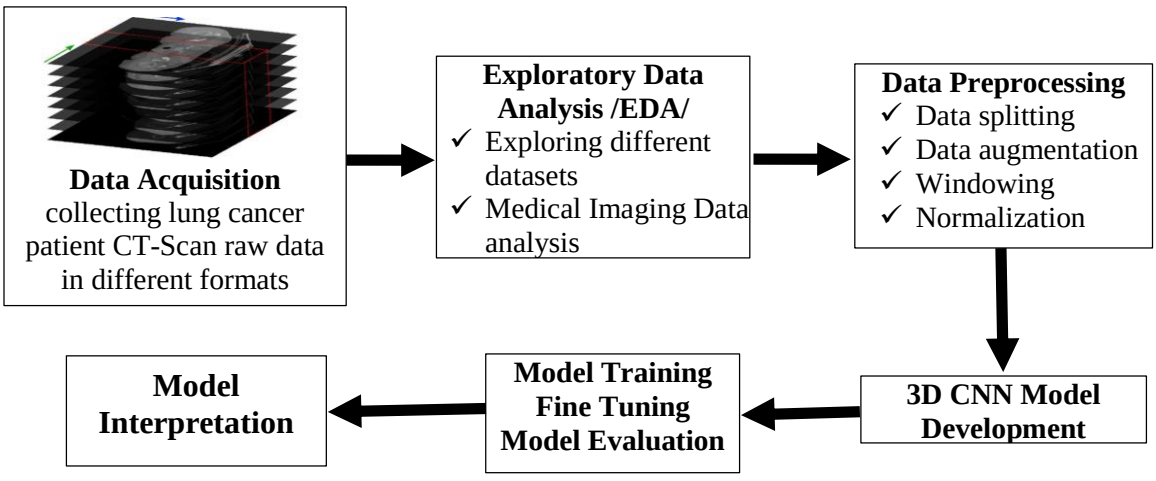


Figure 3.6 Block diagram summary of project work methodology.

CHAPTER FOUR

IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1 Overview

This chapter delves into the development and evaluation of four distinct 3D Convolutional Neural Network (CNN) models for lung nodule classification in chest CT scans. The chapter provides a detailed description of all four 3D CNN architectures. It outlines the number and type of convolutional layers, pooling layers, activation functions, and other relevant components used in the models.

The chapter also an overview of the training process for the 3D CNN models, detailing the experimental design and hyperparameter configurations (such as Optimizers, Learning Rate Scheduling Strategy, and Loss Function) employed during training, evaluation, and subsequent performance optimization.

Finally, the interpretability aspect of the 3D CNN models is presented, emphasizing the implementation and utility of Gradient-weighted Class Activation Mapping (Grad-CAM). Grad-CAM is employed to generate visual explanations for the model's predictions by highlighting the regions within lung nodule images that most significantly influence the classification decision (malignant vs. benign).

4.2 Preprocessing Techniques

We defined various functions for preprocessing, which include:

Read_nifti_file (filepath): Reads a Nifti file containing a 3D volume and returns the 3D volume data. *windowing_and_normalize (volume):* Performs windowing to limit intensity values within a specific range and normalizes the volume intensities to a defined range (e.g., -1 to 1). It returns the windowed and normalized volume. *Resize_volume (img):* Resizes the volume to the desired dimensions (e.g., 64x64x64) along the x, y, and z axes, returning the resized volume. *Process_scan (path):* Reads a Nifti file using read_nifti_file, performs windowing and normalization using windowing_and_normalize and resizes the volume using *resize_volume*. This function returns the processed volume.

Preprocessing (volume, label): Adds a channel dimension to the volume to ensure compatibility with deep learning models, returning the modified volume and its

corresponding label. *save_numpy (volume, path, filename):* Saves a NumPy array representing the volume to a specified file path. *save_niftifile (volume, path, filename):* Converts a NumPy array to a Nifti file and saves it to a specified file path.

plot_HU (volume, title='): (Function for visualization - not core processing) Plots the histogram of Hounsfield Units (HU) for a given volume, aiding in the visualization of the voxel intensity distribution.

To prevent data leakage, we split the data into training (80%), validation (10%), and test (10%) sets before augmentation. We performed data augmentation on the training set to address data imbalance and mitigate overfitting issues.

Labeling and Consistency across Datasets:

In the final step of data preparation, we assigned binary class labels to each case: 0 for benign (cancer-negative) cases and 1 for malignant (cancer-positive) cases. This consistent labeling approach was applied across all training, validation, and ensuring uniformity throughout the dataset. By maintaining this consistency, we ensured that the model's performance could be reliably evaluated, facilitating an accurate assessment of its predictive capabilities and its ability to distinguish between benign and malignant lung nodules.

To ensure transparency and reproducibility, the code for all the aforementioned techniques is provided in Appendix E. This appendix includes detailed code for various stages, such as data acquisition, binary classification (benign vs. malignant), dataset splitting (training, validation, testing), data augmentation, exploratory data analysis, and visualization. Additionally, it covers model building and training, as well as model interpretation. By providing this comprehensive code repository, we aim to facilitate the replication of our work and support further research in this field.

4.3 Model Architectures Used

This section outlines the development process of the proposed 3D Convolutional Neural Network (CNN) models for classifying lung nodules in chest CT scans. To construct these models, we utilized the Keras Functional API, which offers significant flexibility in handling complex architectures. The Keras Functional API is well-suited for creating deep learning models with non-linear topologies, shared layers, and multiple inputs and outputs, making it ideal for designing sophisticated models that can tackle a range of tasks, including 3D image classification. This approach allows for greater customization and scalability compared to

the Keras Sequential API, which is more limited in structure. The flexibility of the Functional API enables us to fine-tune the architecture, optimize the layers, and improve model performance in challenging medical image classification tasks (Fchollet & Team Keras, 2019).

The development process involved three key steps:

Define input to the model: This step included defining the model's input layer and specifying the dimensions of the input tensor, which represents a 3D lung nodule patch extracted from a chest CT scan.

Define a set of interconnected layers on the input: A series of interconnected layers were defined to build the core architecture of the model. These typically included convolutional layers for feature extraction, pooling layers for dimensionality reduction, and activation functions for introducing non-linearity. The specific types and configurations of these layers varied based on the chosen architecture.

Define the model using the input and output layers: This involved finalizing the model structure by connecting the input and output layers.

In this study, we employed *four distinct 3D CNN architectures* for the classification of lung nodules. The first architecture served as a baseline model, inspired by the work of Zunair et al. (2020), who applied 3D CNNs for tuberculosis prediction from CT scans. Their research provided a valuable reference for structuring a 3D CNN model to handle medical imaging data effectively. The baseline model helped establish a foundational architecture, providing a benchmark for evaluating the performance of more advanced or modified architectures. By comparing the results of this baseline model with those of the other 3D CNN models, we were able to assess the impact of different architectural designs on the accuracy and effectiveness of lung nodule classification. This approach allowed for a thorough investigation of the potential improvements and optimizations for nodule detection and classification in chest CT scans.

The second architecture involved the *3D adaptations of* AlexNet, *which is the well-established 2D CNN model*. These adaptations aimed to capitalize on AlexNet’s success in 2D image classification tasks while modifying the architecture to handle volumetric data, such as 3D lung nodule patches. By extending AlexNet's convolutional layers to 3D

convolutions, we were able to explore the potential for leveraging pre-existing architectures to process complex 3D medical data.

The third architecture was a *hybrid model* created by combining and modifying the first two architectures. This combination aimed to enhance performance during training and evaluation by integrating the strengths of both the baseline and the AlexNet-based 3D model. The fourth architecture incorporated the *CBAM (Convolutional Block Attention Module).* CBAM is an attention mechanism designed to improve the representational power of CNNs by focusing on the most informative regions of the input data. By adding this module, we sought to enhance the model's ability to focus on critical features within the 3D lung nodule patches, potentially improving classification accuracy. In the following subsections, we present model architecture diagrams and implementation code snippets for these four models, highlighting the specific configurations and modifications made to enhance their performance for lung nodule classification. All models are designed for an input shape of (64, 64, 64). Detailed model summaries are provided in Appendix D.

4.3.1 Model-1: 3D-CNN Baseline Model

The baseline model consists of a simple 3D CNN architecture with four convolutional layers. This architecture serves as the foundation for comparison with other models developed later in the study. To enhance the generalization performance and prevent overfitting during training, we applied L2 regularization using the kernel_regularizer=l2 (l2_reg) technique to all Conv3D and Dense layers. This regularization method helps prevent the model from overfitting to the training data by penalizing excessively large weights, encouraging the model to generalize better to unseen data.

- Conv3D layers: Four convolutional layers, each followed by activation functions and pooling layers to progressively extract spatial features from the 3D lung nodule data.
- L2 Regularization: Regularization was added to the convolutional and dense layers using the L2 penalty to control the magnitude of weights and improve model generalization.
- Activation Functions: ReLU activation functions were typically used after each convolutional layer to introduce non-linearity and improve learning capacity.

Figure 4.1 depicts the model architecture, illustrating the flow of data from the input layer through each convolutional block and eventually to the output layer for classification.

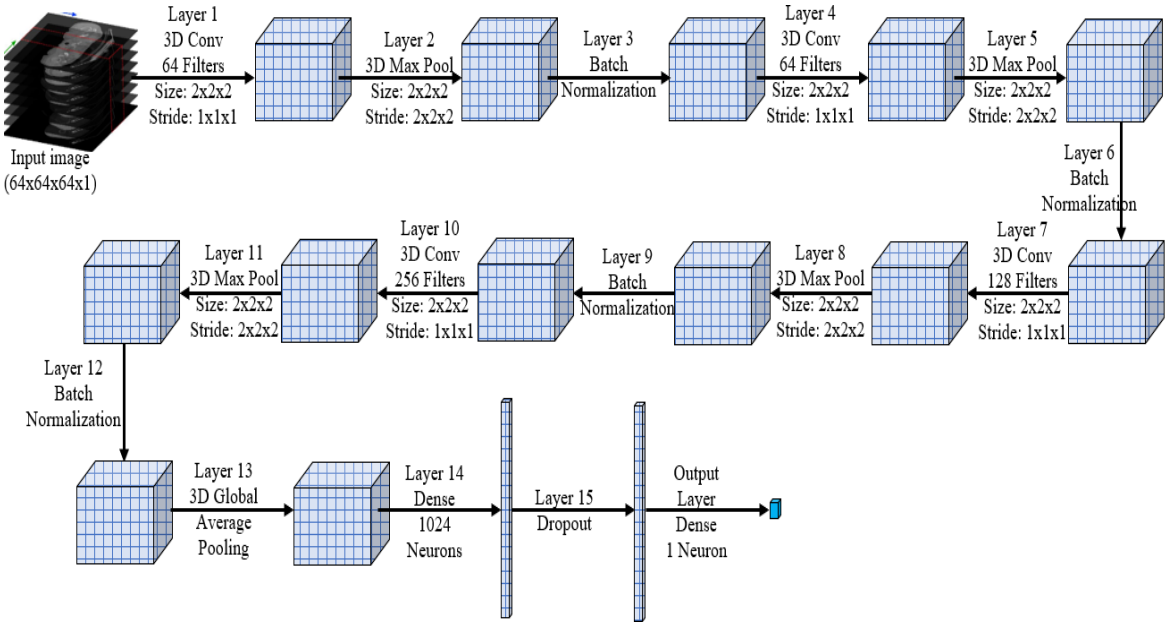


Figure 4.1 Baseline Model Architecture.

Key Considerations for the baseline model are the following. One, L2 Regularization (l2 (l2_reg)), Helps prevent overfitting by adding a penalty to the loss function that discourages large weights. MaxPooling3D Layers: These layers reduce the spatial dimensions of the data after each convolution, helping to retain important features and reduce computation. The other is the Fully Connected Layer: After flattening the output of the convolutional blocks, a dense layer is added to learn higher-level representations. Lastly, Output Layer: - where a sigmoid activation function is used in the final layer for binary classification (malignant vs. benign).

This baseline model serves as a simple yet effective approach for the initial lung nodule classification task, offering a strong foundation for comparing with more complex models later in the study.

4.3.2 Model-2: 3D-AlexNet Model

AlexNet, developed by Alex Krizhevsky in 2012, is a groundbreaking convolutional neural network (CNN) architecture that achieved outstanding results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Key features of AlexNet include convolutional layers equipped with filters designed to detect specific features within images. At the time, AlexNet was considered an intense network consisting of five convolutional layers followed by three fully-connected layers. This substantial depth enabled the model to learn intricate

relationships between features at various levels of abstraction, significantly advancing the field of computer vision and setting new benchmarks for image classification tasks.

AlexNet was indeed a groundbreaking achievement in the world of computer vision. Its architecture laid the foundation for many subsequent innovations in the field. Here are a few more interesting aspects of AlexNet. First, *Activation Function:* AlexNet used the ReLU (Rectified Linear Unit) activation function, which helped in faster training than traditional activation functions like tanh or sigmoid. Second, *GPU Utilization:* The training of AlexNet utilized GPUs (graphics processing units) to handle the intensive computations required for deep learning, significantly speeding up the process. Third, *Data Augmentation:* To enhance the model's generalization capability and to prevent overfitting, AlexNet employed techniques like image translations, horizontal reflections, and altering the intensities of the RGB channels. It's fascinating how a single architecture can profoundly impact the development of deep learning techniques. If you're interested, we can dive deeper into specific aspects or explore how AlexNet influenced subsequent models.

AlexNet significantly shifted neural network training by implementing the ReLU (Rectified Linear Unit) activation function instead of the more traditional sigmoid function. This change allowed for faster training and better performance in deeper networks. Additionally, AlexNet utilized dropout layers after some of the fully connected layers to combat overfitting. These dropout layers randomly deactivate a percentage of neurons during training, which helps the network learn robust features that aren't reliant on specific neurons. This approach enhances the model's generalization capability.

Although AlexNet achieved remarkable success, it is relatively computationally expensive compared to newer architectures. Modern convolutional neural networks (CNNs) often feature deeper structures and employ techniques like batch normalization to enhance training efficiency. Nevertheless, AlexNet remains a crucial milestone in the history of deep learning. Its pioneering architecture and its success in image recognition continue to shape advancements in computer vision.

The second model architecture employed in this study is adapted from the 3D AlexNet design proposed by Rani et al. (2022) for MRI brain tumor classification. This architecture was chosen for its efficiency in handling volumetric data and its demonstrated effectiveness in medical imaging tasks. The modifications tailored the architecture to the lung nodule classification task, including: 1) Batch Normalization: Added after convolutional layers to

stabilize and accelerate training by normalizing layer inputs and reducing sensitivity to initialization. 2) Dropout Layers: Incorporated after dense layers to mitigate overfitting by randomly deactivating neurons during training, promoting robustness.

These modifications aimed to improve the model's training efficiency and generalization capabilities.

Model Architecture: There are four major blocks in the architecture. It has multiple 3D convolutional layers to extract hierarchical features from the input data. The other is Batch Normalization, an intermediate output to maintain stable activations across layers. Further, Dropout Regularization randomly deactivates neurons in dense layers to prevent overfitting. Also, Fully Connected Layers encode high-level feature representations for final classification. Finally, Binary Classification outputs the final layer using a sigmoid activation function to distinguish between benign and malignant nodules.

Key Model Features: Input Size of (64, 64, 64) to match the preprocessed lung nodule patches. Regularization Techniques, such as Batch normalization and dropout layers, are added to enhance model robustness and training dynamics.

Training Enhancements: Batch Normalization improves convergence speed and reduces overfitting risks. Further, Dropout Regularization is applied to minimize model dependence on specific neurons, thereby enhancing generalization. Also, the Adam optimizer and binary cross-entropy loss function were utilized for effective training.

The architecture, depicted in Figure 4.2 below, showcases its layered complexity designed to exploit the 3D spatial context of lung nodule data. This AlexNet-inspired model provides a robust framework for volumetric medical image classification tasks.

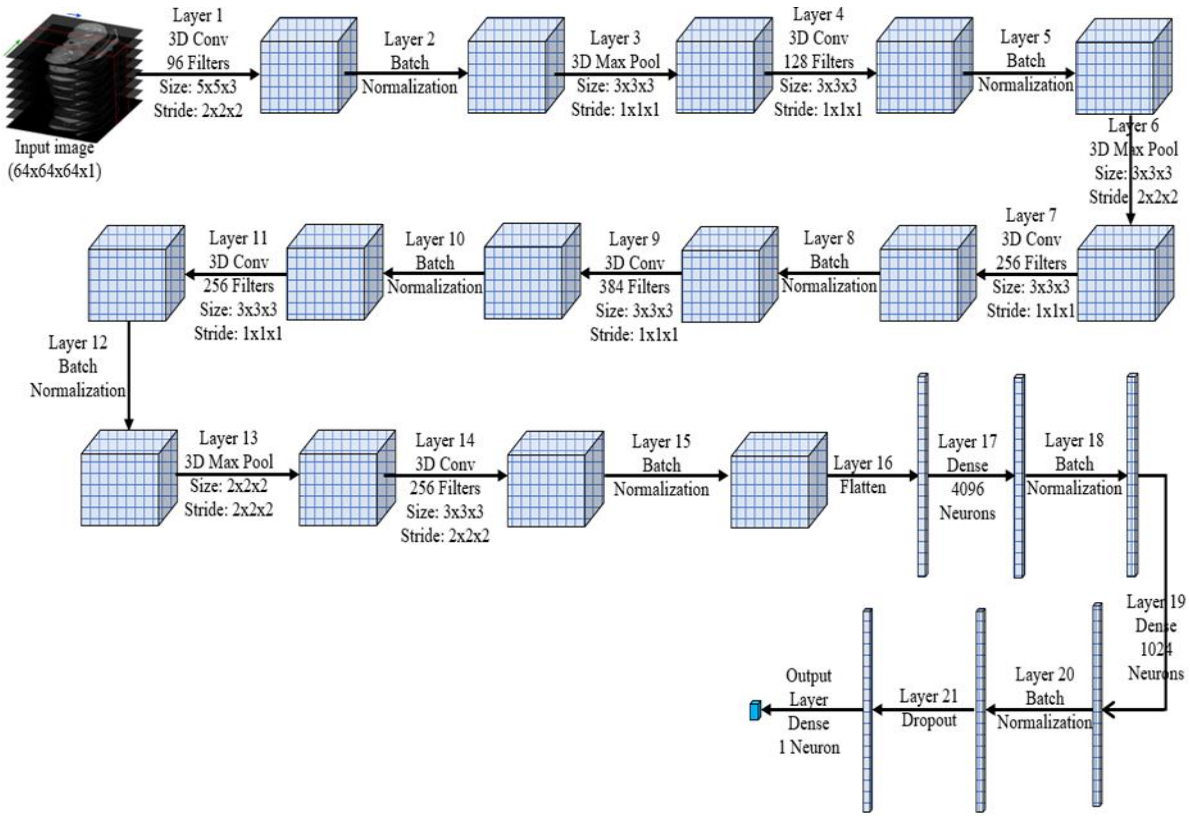


Figure 4.2 3D AlexNet Model Architecture.

4.3.3 Model-3: Proposed 3D CNN Model

While trying to achieve strong performance, we investigated various architectural configurations to combine the aforementioned model architecture layers. Through rigorous experimentation, we identified the following architecture, which yielded the most promising results.

After the input layer to extract spatial features from the input 3D data, we used 3D convolutional filters. Conv3D layers apply 3D convolution operations to the input data. Each Conv3D layer has a specified number of filters and a kernel size of 2. The activation function used is ReLU, which introduces non-linearity to the model. The L2 regularization helps prevent overfitting by penalizing large weights.

Then, the MaxPooling3D Layer performs down-sampling by taking the maximum value over a specified pool size (2 in this case). This reduces the spatial dimensions of the feature maps, which helps reduce computational complexity and control overfitting.

Then, we used BatchNormalization Layers to normalize the output of the previous layer. This helps stabilize and speed up the training process by reducing internal covariate shifts. After that, we used Flatten Layer: This layer converts the 3D feature maps into a 1D vector, which can then be fed into the fully connected layers. It essentially prepares the data for the dense layers. Then Dense Layers: These are fully connected layers with 4096 units each, using ReLU activation. They perform high-level processing on the features extracted by the convolutional layers. L2 regularization is applied to prevent overfitting. Then, Dropout Layer: This layer randomly sets 70% of the input units to zero during training. This helps avoid overfitting by ensuring the model does not rely too heavily on any particular set of features. Finally, the Output Layer: Dense Is the final layer with a single unit and sigmoid activation function. This layer outputs a value between 0 and 1, making it suitable for binary classification tasks. The model architecture is depicted in Figure 4.3.

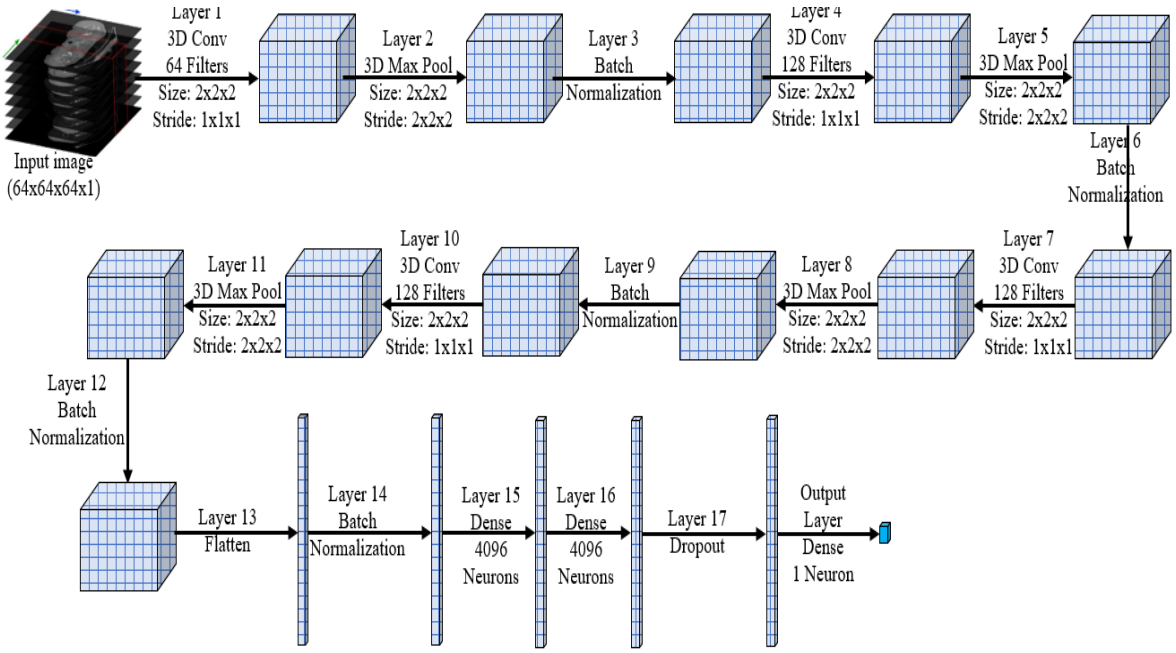
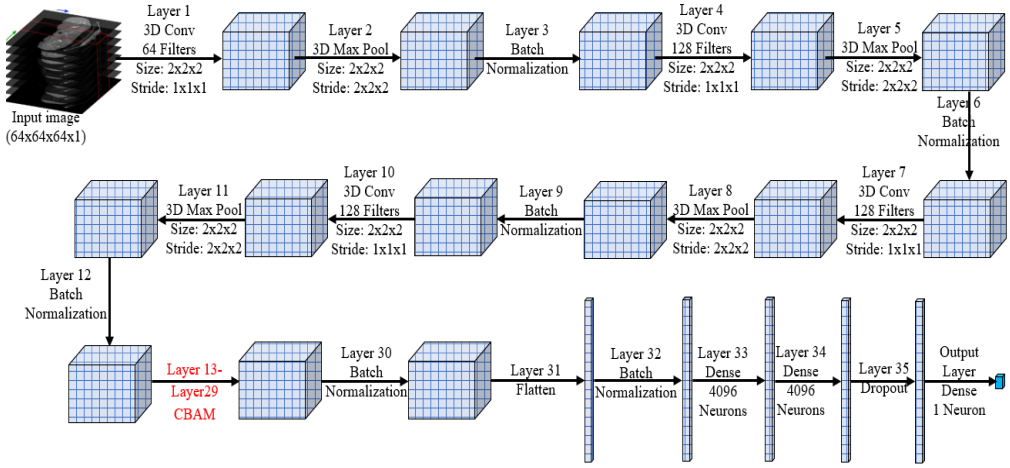


Figure 4.3 Proposed Model Architecture

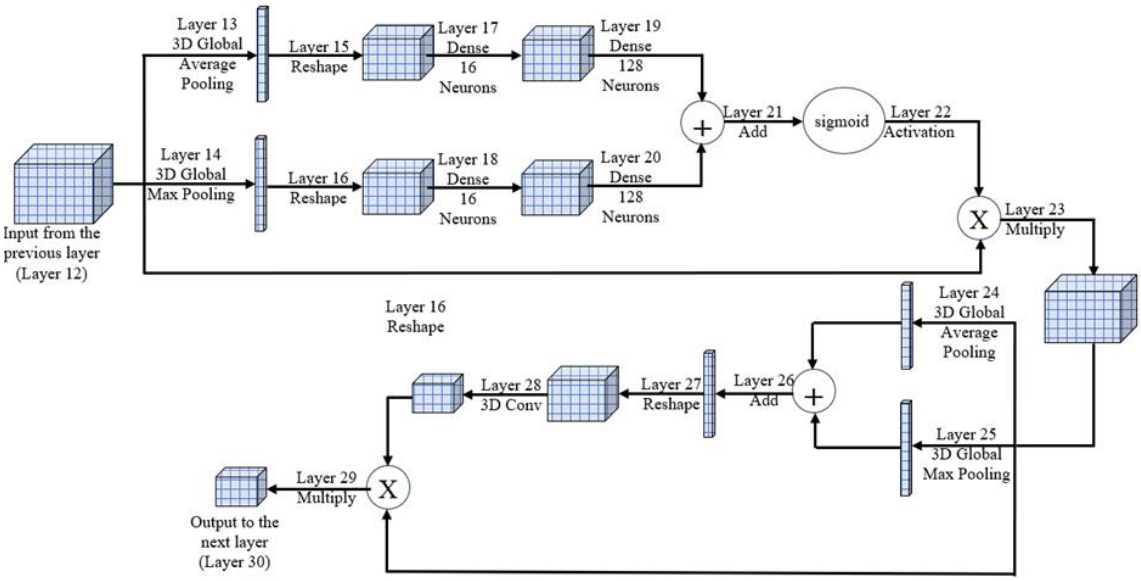
4.3.4 Model-4 / Proposed 3D CNN Model with CBAM

In our quest to enhance the performance of our proposed model, we conducted additional experiments and introduced the CBAM (Convolutional Block Attention Module). Placing the CBAM module after the last convolutional layer and before the flattening layer enables

The model concentrates on the most critical features before moving on to the dense layers. Although the overall architecture remains similar to our original model, it now gains the advantage of the attention mechanism provided by CBAM. Figure 4.4 illustrates the model architecture, accompanied by the corresponding code implementation.



(a) The overall model architecture



(b) CBAM Module

Figure 4.4 Proposed Model with CBAM Architecture.

4.4 Model Training and Evaluation

For all four models, we followed the steps depicted on the block diagram in Figure 4.5.

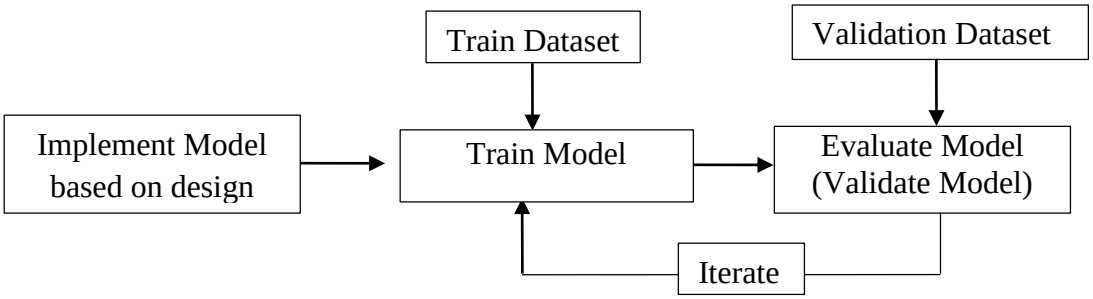


Figure 4.5 Block Diagram of Model Training Process

After the training process, the models are saved in the Hierarchical Data Format 5 (HDF5), commonly referred to with the .h5 extension. This format is popular for deep learning models as it stores not only the model architecture but also the trained weights and optimizer state. To ensure optimal performance, we employed a checkpointing mechanism during training using the 'ModelCheckpoint callback.' This technique periodically evaluates the model's performance on the validation set and saves a copy of the model with the best validation accuracy achieved so far. This allows for resuming training from the best-performing model if the training process is interrupted or if hyperparameter tuning requires restarting training.

Due to the binary classification nature of the problem, we initially used the Binary Cross-Entropy (BCE) loss function. However, after various experiments and considering that the validation data is not augmented, we opted for Binary Focal Crossentropy. This modified version of binary cross-entropy introduces a focusing parameter that down-weights easy examples and emphasizes hard-to-classify ones. It is particularly effective in scenarios with class imbalance, helping the model concentrate more on the minority class or difficult examples. The formula for binary focal cross-entropy is given in the equation. (4.1).

$$BFCL = -\frac{1}{N}\sum_{i=1}^{N}\alpha(1-p_i)^{\gamma}\left[y_i\log(p_i)+(1-y_i)\log(1-p_i)\right] \quad 4.1$$

Where: *BFCL* is Binary Focal Crossentropy Loss, α is a balancing factor to adjust the importance of positive/negative examples. γ is the focusing parameter that adjusts the rate at which easy examples are down-weighted. A higher value of γ increases the focus on hard examples.

4.5 Model Interpretation with GRAD-CAM

In this stage, we explored the use of Gradient-weighted Class Activation Mapping (Grad-CAM) to interpret the predictions of our deep-learning model for lung nodule classification. Grad-CAM is a technique that aims to provide visual insights into the regions of a lung nodule image that contribute most significantly to the model's classification decision (malignant vs. benign).

We focused on the last convolutional layer for Grad-CAM analysis, as deeper layers in convolutional neural networks tend to capture more abstract and semantic features, such as shapes and textures, which can be more relevant for classification. We identified the last convolutional layer using the model summary.

Grad-CAM leverages the class-discriminative nature of the network by generating separate visualizations for each class (benign and malignant). The heatmap is created by calculating the weighted sum of the feature map channels based on their importance for the predicted class. Finally, normalization (0-1) is applied for visualization purposes.

To visualize the results, we overlaid the generated heatmaps onto the original CT scan slices. This allowed us to visually inspect the regions of the nodule that the model focused on during the classification process. The code snippet for our Grad-CAM implementation is available in Appendix E. By incorporating these minor suggestions; you can further enhance the clarity and detail of this section.

CHAPTER FIVE

RESULTS AND DISCUSSION

5.1 Overview

This section delves into the results obtained during the training and evaluation of our four 3D Convolutional Neural Network (CNN) models. We utilize the performance metrics outlined in Chapter 3 to rigorously assess their effectiveness. In addition to the performance evaluation, we present Grad-CAM visualizations. This technique is employed to interpret and visualize the decision-making processes of each model, providing valuable insights into the regions within the images that most significantly influence the model's predictions.

5.2 Results

5.2.1 Experimenting Model 1

The training and validation result of Model-1 (Baseline Model) is presented in Figure 5.1 below.

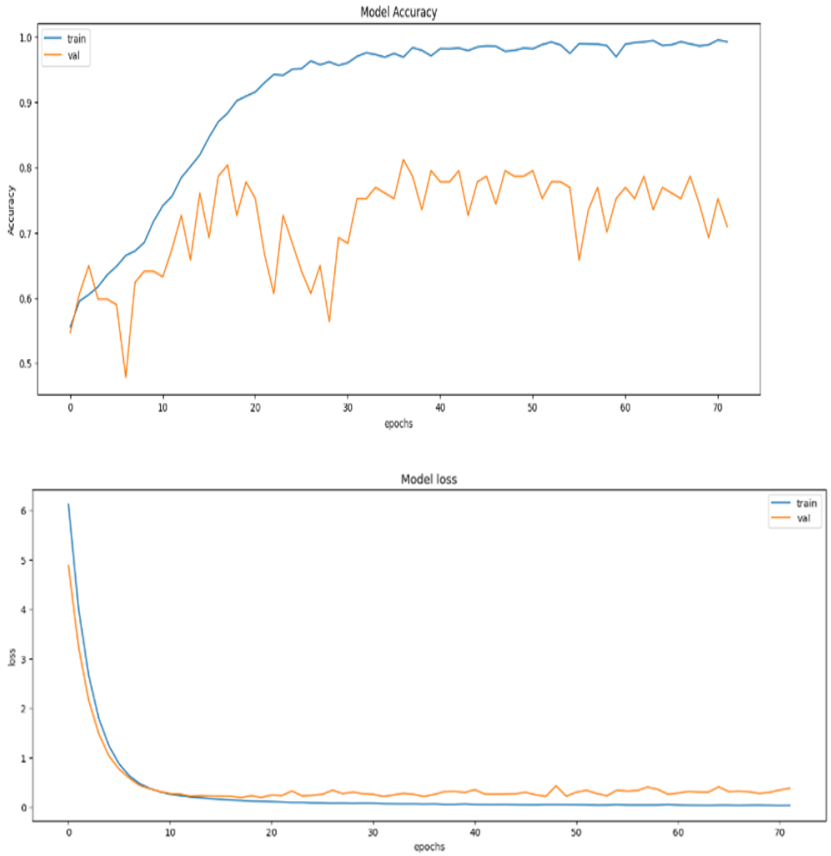


Figure 5.1 Baseline Model accuracy and loss during training and validation.

During the training process, we achieved a maximum validation accuracy of 81.19%. At this particular epoch, the model exhibited remarkable performance metrics during training, including an accuracy of 99.23%, precision of 99.25%, recall of 99.33%, AUC (Area Under the Curve) of 99.38%, and an F1-score of 99.29%. For the validation dataset, the results were also notable, with an accuracy of 81.19%, precision of 80.64%, recall of 94.93%, AUC of 74.76%, and an F1-score of 87.20%. These metrics highlight the model's robust performance and its ability to generalize well to unseen data, although there's room for improvement in certain validation metrics.

5.2.2 Experimenting Model -2

The training and validation result of model – 2 (3D AlexNet Model) is presented in Figure 5.2 below.

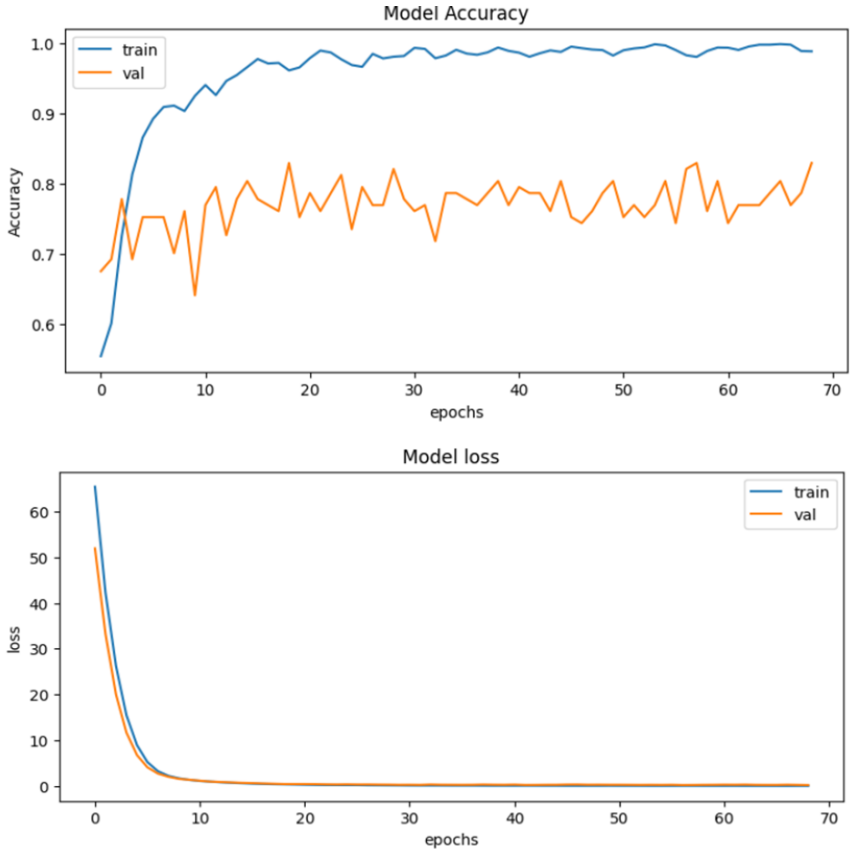


Figure 5.2 3D AlexNet Model Accuracy and Loss during Training and Validation.

During the training of the 3D AlexNet model, we achieved a maximum validation accuracy of 82.90%. At this epoch, the model's training performance metrics were an accuracy of 98.78%, precision of 98.70%, recall of 99.05%, AUC of 99.91%, and an F1-score of 98.88%.

For the validation set, the performance metrics included an accuracy of 82.90%, precision of 84.70%, recall of 91.13%, AUC of 84.42%, and an F1-score of 87.80%.

5.2.3 Experimenting model -3

The training and validation result of Model-3 / (Proposed 3D CNN Model) is presented in Figure 5.3 below.

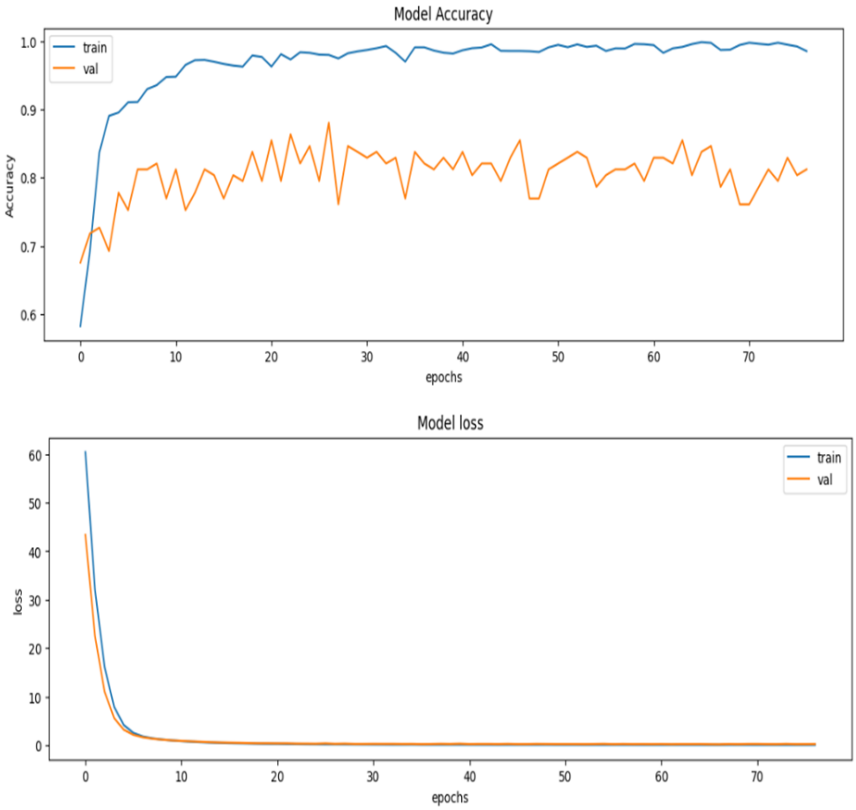


Figure 5.3 Proposed Model Accuracy and Loss during Training

During the training of Model-2, our proposed model, we achieved a maximum validation accuracy of 83.76%. At this epoch, the training performance metrics included an accuracy of 97.07%, precision of 97.58%, recall of 97.01%, AUC of 99.64%, and an F1-score of 97.92%. For the validation dataset, the model attained an accuracy of 83.76%, precision of 82.61%, recall of 96.20%, AUC of 82.15%, and an F1-score of 88.79%.

5.2.4 Experimenting model 4

The training and validation result of Model-4 / (Proposed 3D CNN Model) is presented in Figure 5.4 below.

During the training of the proposed 3D-CNN with the CBAM model, we achieved a maximum validation accuracy of 88.03%. At this epoch, the training performance metrics

were: accuracy of 97.95%, precision of 97.49%, recall of 98.84%, AUC of 99.75%, and F1-score of 98.16%. For the validation dataset, the model achieved an accuracy of 88.03%, precision of 88.24%, recall of 94.94%, AUC of 90.04%, and an F1-score of 91.46%.

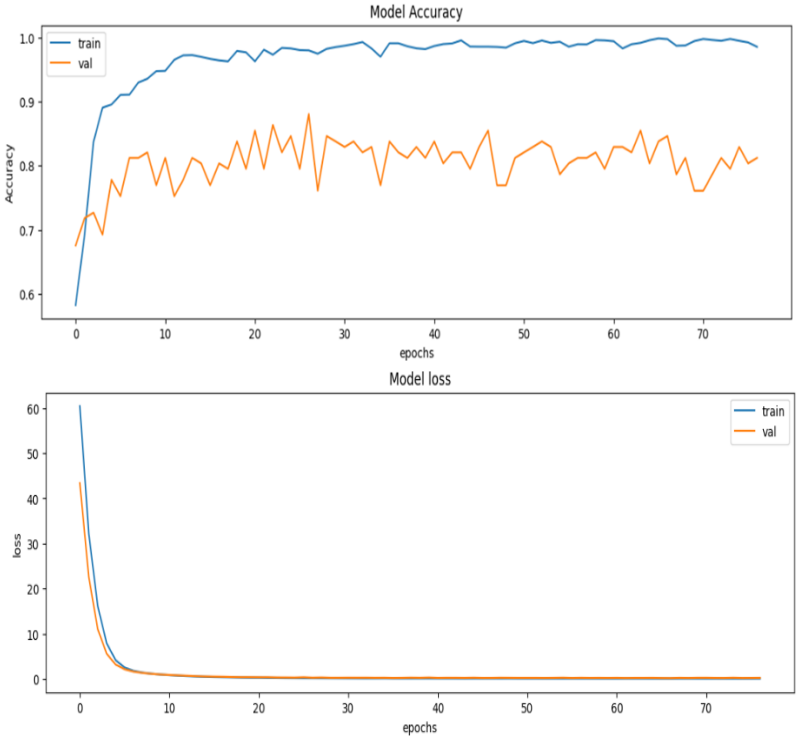


Figure 5.4 Proposed 3D-CNN with CBAM Model Accuracy and Loss during Training.

Training Accuracy indicates how well the model is performing on the training dataset. A high training accuracy suggests that the model is learning the patterns in the training data effectively. Validation Accuracy measures the model's performance on unseen data (validation set). It helps in assessing how well the model generalizes to new data. Ideally, validation accuracy should be close to training accuracy.

Training Error (Loss) represents the error or loss on the training dataset. A decreasing training error over time indicates that the model is learning and improving. Validation Error (Loss) shows the error on the validation dataset. Monitoring validation errors helps in detecting overfitting. If validation error starts increasing while training error decreases, it may indicate overfitting. Those figures show how these metrics change over epochs. Consistent improvement in both training and validation accuracy, along with decreasing errors, suggests a well-performing model.

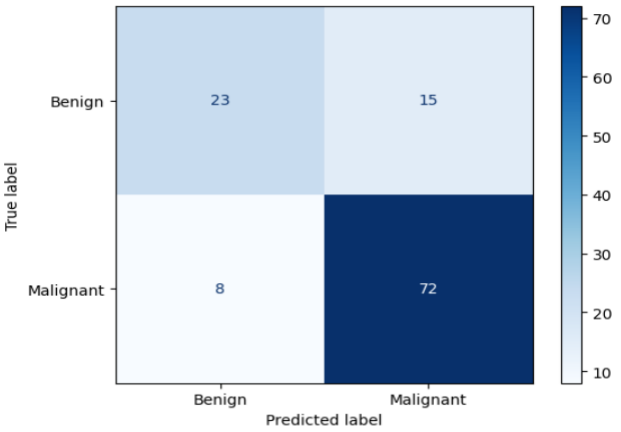
5.2.5 Testing Results

These results were obtained during model testing using the test dataset, which comprises 10% of the entire dataset. The test set includes 38 remaining test cases of the benign class and 80 test cases of the malignant class.

Table 5.1 Results of Performance Metrics for Testing Set

Models	**Performance Metrix**		
	Accuracy (%)	AUC (%)	F1-Score (%)
Baseline 3D CNN Model	80.50	85.26	86.75
3D ALEXNET	83.05	79.76	88.78
Proposed 3D CNN Model	88.98	97.91	91.50
Proposed 3D CNN Model with CBAM	**94.06**	**98.84**	**95.56**

Confusion Matrices and Classification Report for all Four Models: **For Model-1 / Baseline**



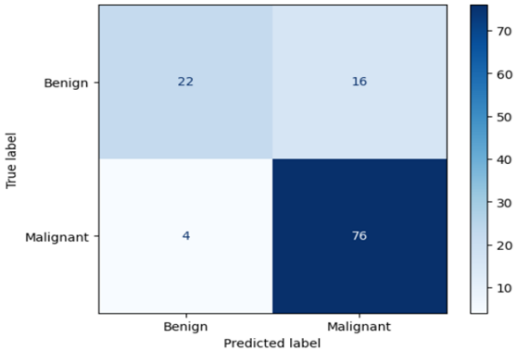
	precision	recall	f1-score	support
0.0	0.74	0.61	0.67	38
1.0	0.83	0.90	0.86	80
accuracy			0.81	118
macro avg	0.78	0.75	0.76	118
weighted avg	0.80	0.81	0.80	118

Figure 5.5 Confusion Matrix and Classification Report for the Baseline Model

The confusion matrix depicted in Figure 5.5 illustrates the performance of Model-1 in classifying lung nodules as either benign or malignant. *True Positives* represent malignant nodules that are correctly identified as malignant. *True Negatives* denote benign nodules that are accurately classified as benign. *False Positives* refer to benign nodules that are mistakenly classified as malignant, whereas *False Negatives* correspond to malignant nodules that are incorrectly classified as benign.

The model correctly predicted 72 malignant cases and 23 benign cases. The model incorrectly predicted 15 benign cases as malignant and eight malignant cases as benign.

For Model-2: 3D AlexNet model

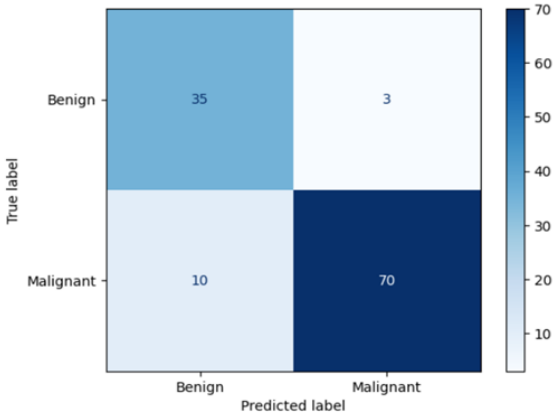


	precision	recall	f1-score	support
0.0	0.85	0.58	0.69	38
1.0	0.83	0.95	0.88	80
accuracy			0.83	118
macro avg	0.84	0.76	0.79	118
weighted avg	0.83	0.83	0.82	118

Figure 5.6 Confusion Matrix and Classification Report for the 3D AlexNet Model

The confusion matrix shown in Figure 5.6 illustrates the performance of Model 2 in classifying lung nodules as benign or malignant. The model accurately predicted 76 cases as malignant and 22 cases as benign. However, it misclassified 16 benign cases as malignant and four malignant cases as benign.

For Model-3: Proposed Model

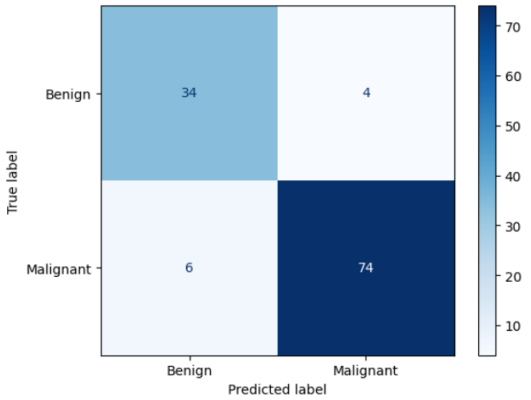


	precision	recall	f1-score	support
0.0	0.78	0.92	0.84	38
1.0	0.96	0.88	0.92	80
accuracy			0.89	118
macro avg	0.87	0.90	0.88	118
weighted avg	0.90	0.89	0.89	118

Figure 5.7 Confusion Matrix and Classification Report for Proposed Model

As shown in the confusion matrix in Figure 5.7, the model accurately identified 70 malignant cases and 35 benign cases. However, it incorrectly classified three benign cases as malignant and 10 malignant cases as benign.

For Model-4: Proposed Model with CBAM



	precision	recall	f1-score	support
0.0	0.85	0.89	0.87	38
1.0	0.95	0.93	0.94	80
accuracy			0.92	118
macro avg	0.90	0.91	0.90	118
weighted avg	0.92	0.92	0.92	118

Figure 5.8 Confusion Matrix and Classification Report for Proposed Model with CBAM

As illustrated in the confusion matrix in Figure 5.8, the model accurately identified 74 malignant cases and 34 benign cases. However, it misclassified four benign cases as malignant and six malignant cases as benign.

5.2.6 Interpretability Results

The interpretability results for all three models are depicted in the next table. We showed the visual explanations using the heatmap generated by Grad-CAM for a single slice from the whole 64 slices.

Table 5.2 Visual Insights into the Baseline 3D CCN Model's decision making

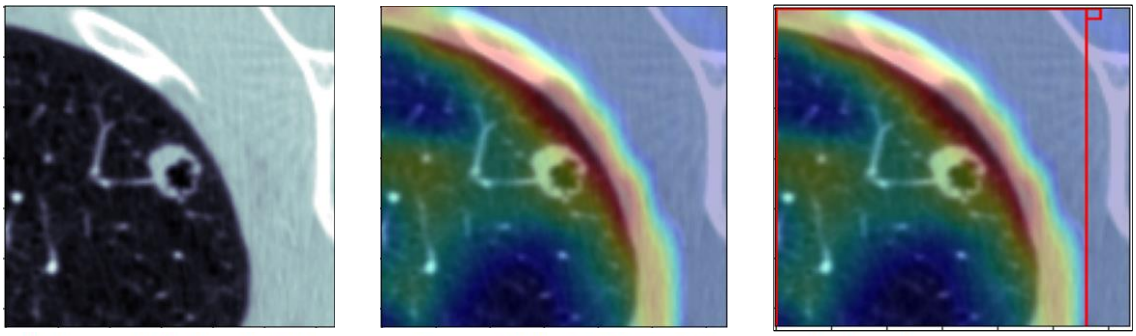
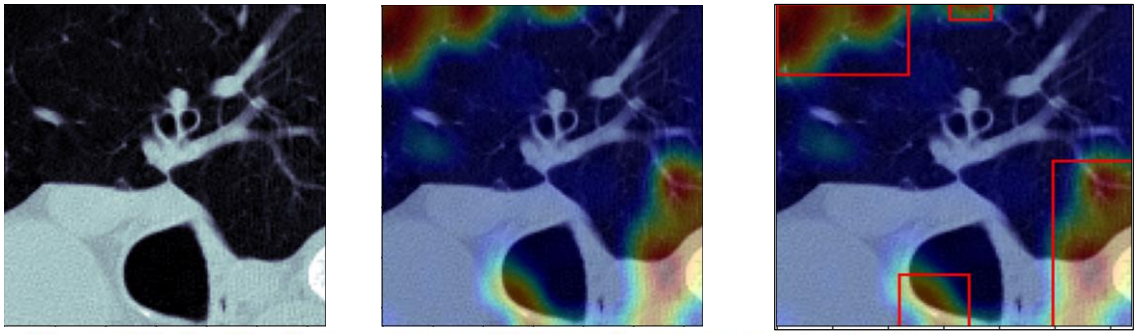
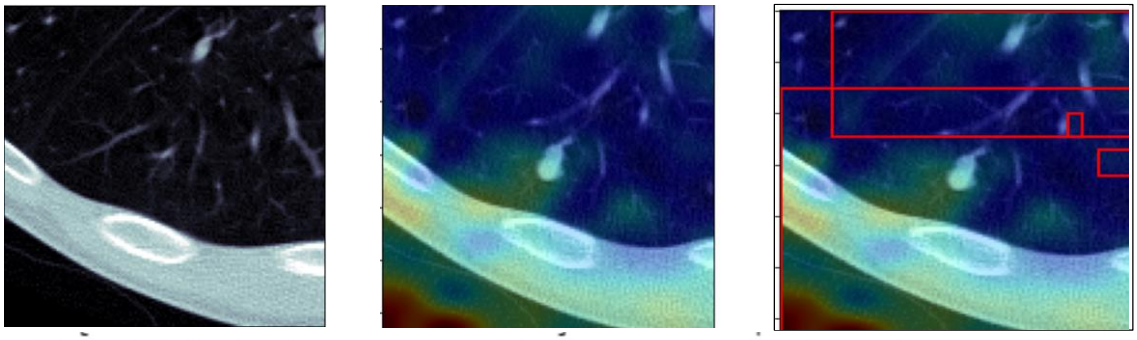
Original Slice	Generated Heatmap	Generated Heatmap with Bounding Box
 This model is 40.27 percent confident that CT scan is benign. This model is 59.73 percent confident that CT scan is malignant. True value = Malignant		
 This model is 95.72 percent confident that CT scan is benign. This model is 4.28 percent confident that CT scan is malignant. True value = Malignant		
 This model is 17.95 percent confident that CT scan is benign. This model is 82.05 percent confident that CT scan is malignant. True value = Malignant		

Table 5.3 Visual Insights into the 3D AlexNet model's decision making

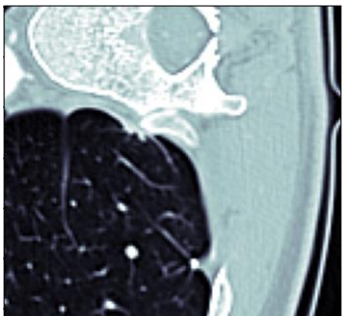
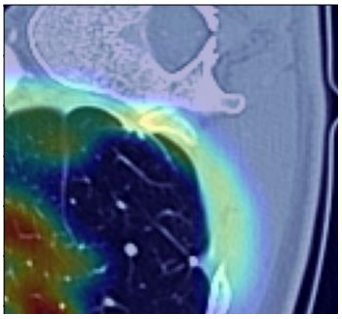
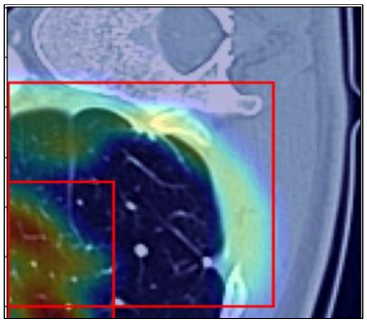
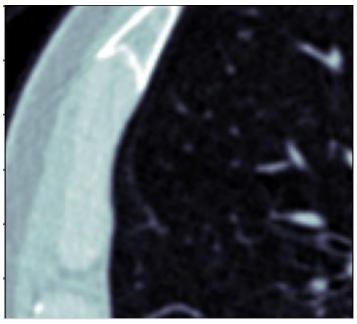
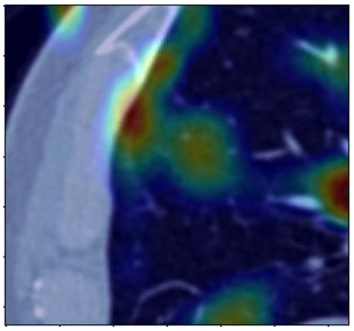
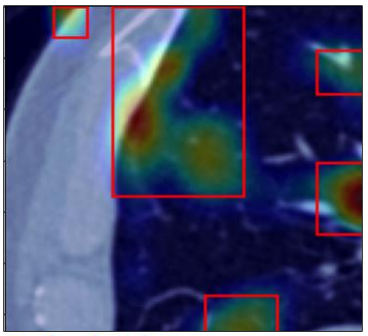
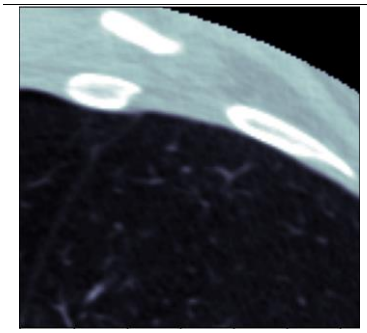
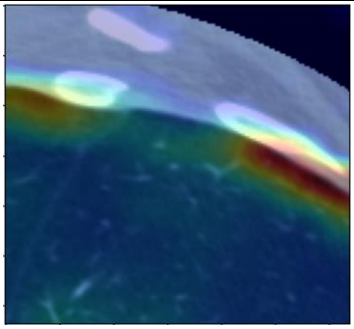
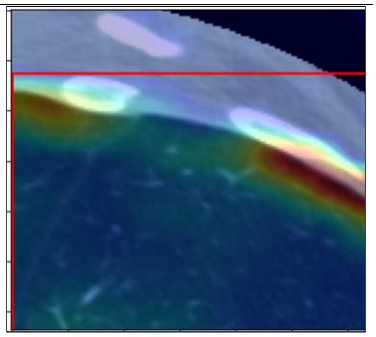
Original Slice	Generated Heatmap	Generated Heatmap with Bounding Box
		
This model is 38.51 percent confident that CT scan is benign. This model is 61.49 percent confident that CT scan is malignant. True value = Malignant		
		
This model is 49.34 percent confident that CT scan is benign. This model is 50.66 percent confident that CT scan is malignant. True value = Benign		
		
This model is 64.13 percent confident that CT scan is benign. This model is 35.87 percent confident that CT scan is malignant. True value = Malignant		

Table 5.4 Visual Insights into the proposed model's decision making

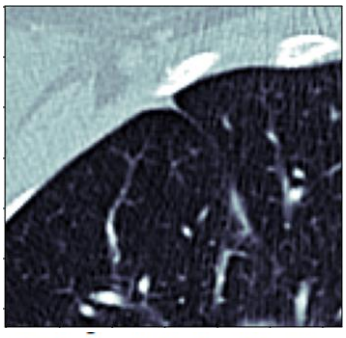
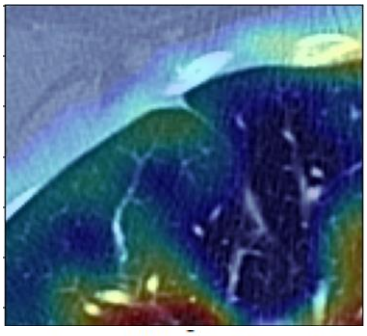
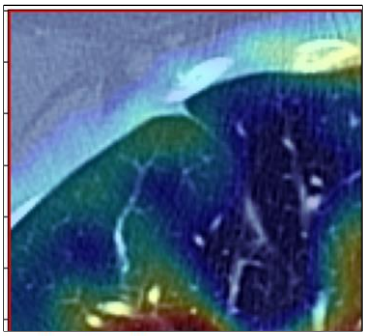
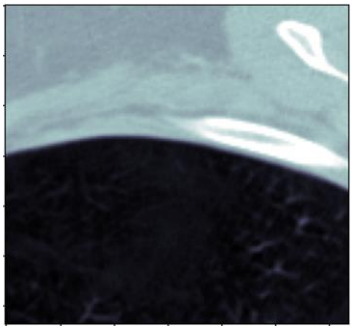
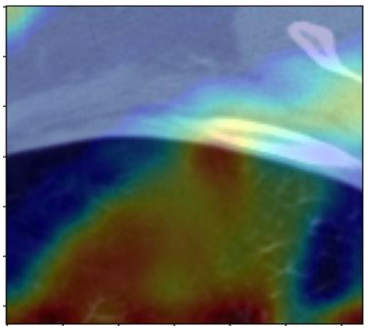
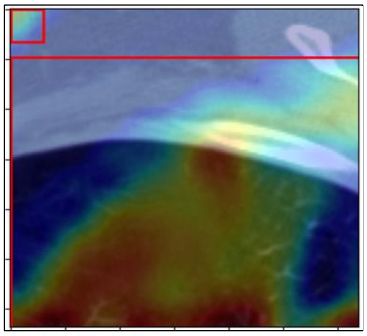
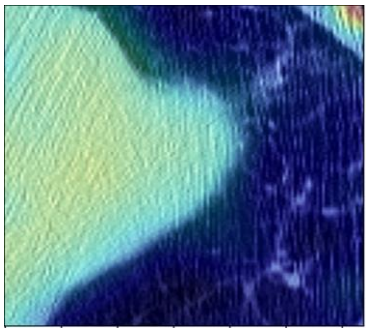
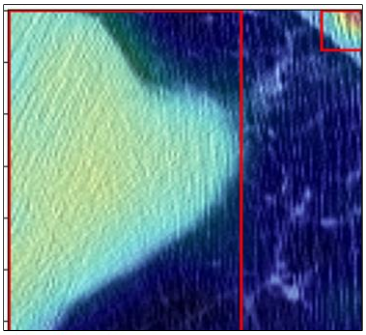
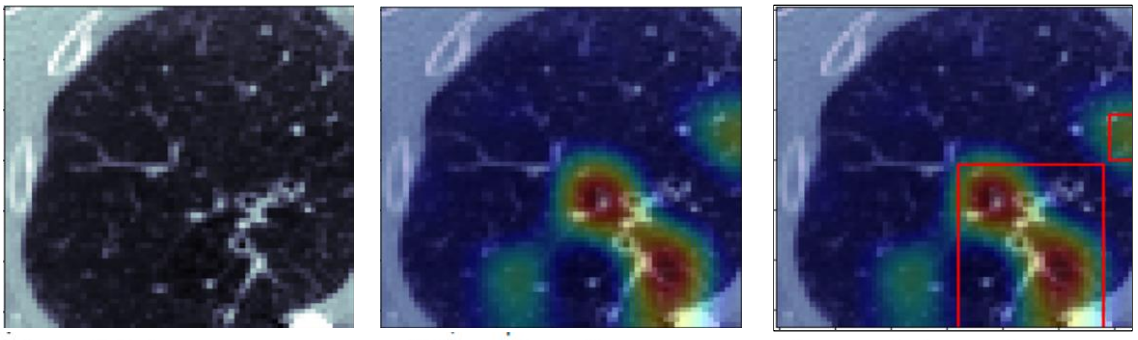
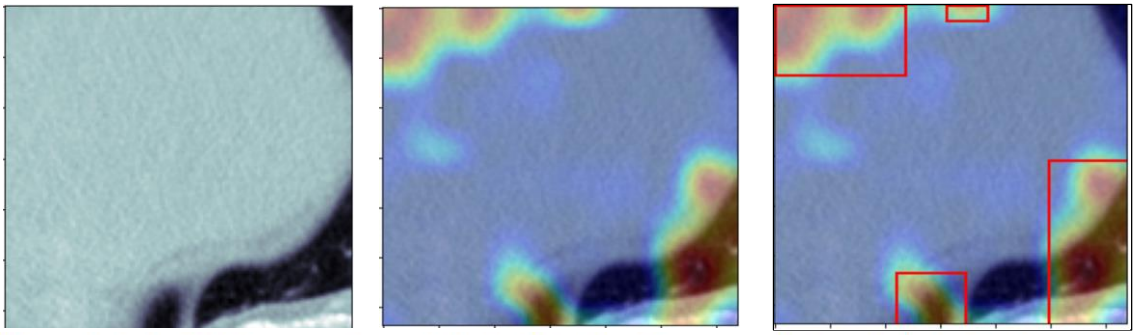
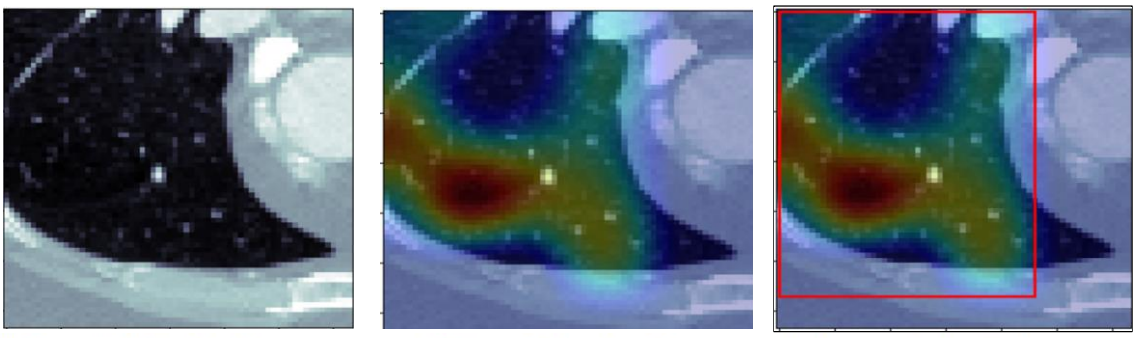
Original Slice	Generated Heatmap	Generated Heatmap with Bounding Box
		
This model is 50.21 percent confident that CT scan is benign. This model is 49.79 percent confident that CT scan is malignant. True value = Benign		
		
This model is 50.22 percent confident that CT scan is benign. This model is 49.78 percent confident that CT scan is malignant. True value = Benign		
		
This model is 7.39 percent confident that CT scan is benign. This model is 92.61 percent confident that CT scan is malignant. True value = Benign		

Table 5.5 Visual Insights into the Proposed Model with CBAM Model's decision making

Original Slice	Generated Heatmap	Generated Heatmap with Bounding Box
	This model is 6.98 percent confident that the CT scan nodule patch is Benign. This model is 93.02 percent confident that the CT scan nodule patch is Malignant. The Radiologists determined the nodule as (true class) is: Malignant.	
	This model is 40.11 percent confident that the CT scan nodule patch is Benign. This model is 59.89 percent confident that the CT scan nodule patch is Malignant. The Radiologists determined the nodule as (true class) is: Malignant.	
	This model is 38.77 percent confident that the CT scan nodule patch is Benign. This model is 61.23 percent confident that the CT scan nodule patch is Malignant. The Radiologists determined the nodule as (true class) is: Malignant.	

Lung tissue frequently appears darker on CT scans, with lung nodules manifesting as white spots. As a result, radiologists typically focus on these dark regions containing white spots. Ideally, our models should replicate this approach by concentrating solely on the lung areas. However, our visualizations revealed instances where certain models, such as Model 4, were drawn to non-lung regions, even when the classification was accurate. Figure 5.9 shows a sample test lung nodule CT scan image, demonstrating how the model might inadvertently focus on areas outside the lung regions.

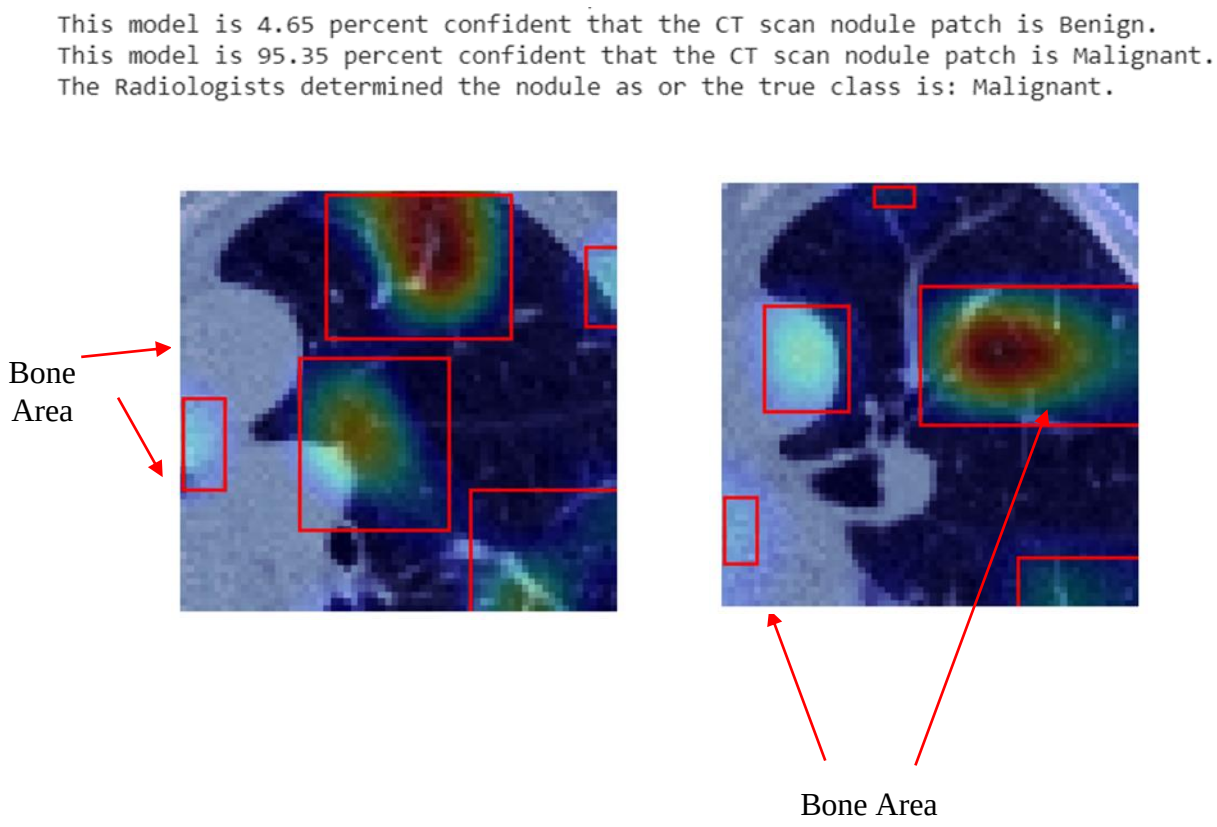


Figure 5.9 Sample Model-4 Result Interpretability Evaluation by the Radiologists

According to feedback from expert radiologists, interpretability visualizations should ideally display all three planes of the 3D CT scan image: axial, sagittal, and coronal. This approach corresponds with how radiologists typically view and analyze CT scans using medical imaging software. To thoroughly understand the model's focus, we generated 3D visualizations for all three planes, as shown in Figure 5.10.

Experts overwhelmingly preferred 3D visualization, which includes all three planes (axial, sagittal, and coronal) over single-plane or solely axial views. This comprehensive approach offered a more holistic understanding of the model's focus, aiding in interpreting its predictions. By presenting the model's attention from different perspectives, clinicians could

better evaluate the relevance and accuracy of the generated heatmaps, thereby enhancing the interpretability and reliability of the model's outputs.

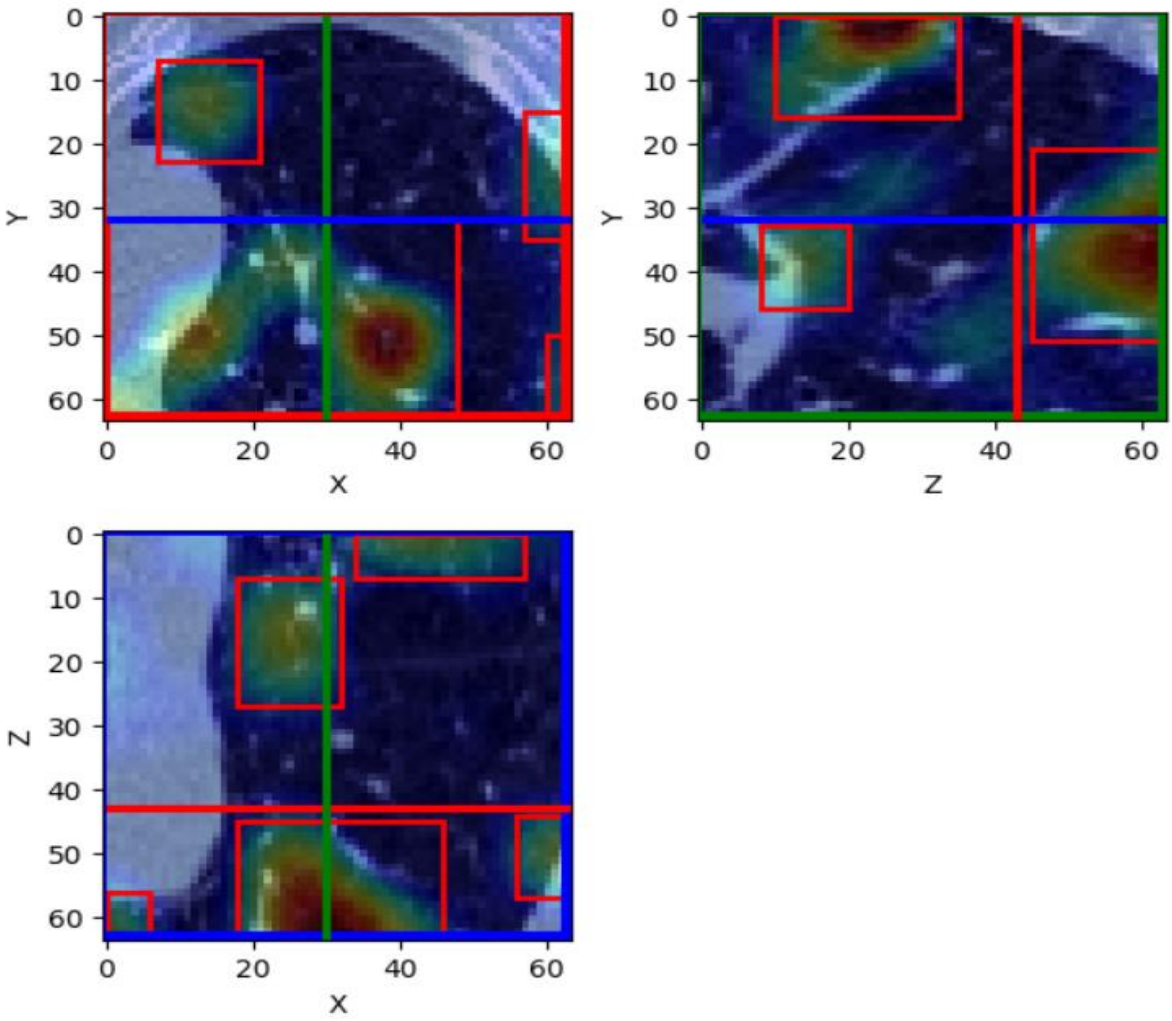


Figure 5.10 Sample 3D Interpretability Visualization Based on the Experts' Suggestions

5.3 Discussions

5.3.1 Results Discussion

The training history reveals that the initial 3D CNN models performed well in classifying lung nodules as benign or malignant. However, a noticeable decline in performance on the test and validation datasets suggested the presence of overfitting, where the models were overly tuned to the training data at the expense of generalizability. To address this issue, dropout layers and L2 regularization techniques were implemented across all models, with dropout rates varying from 60% to 80%. Among these, a dropout rate of 70% proved the most effective, achieving the highest validation accuracy and demonstrating its capability to curb overfitting while maintaining model robustness.

Additionally, an in-depth evaluation of different learning algorithms and optimizers was performed to enhance the training process. The Adam optimizer emerged as the most consistent performer across all 3D CNN models. Its adaptive learning rate and momentum-based approach effectively facilitated optimization, making it particularly suitable for this task. These findings underscore the importance of regularization techniques and algorithm selection in achieving both high accuracy and generalizability in the classification of lung nodules.

Given the substantial memory requirements associated with processing 3D data and implementing 3D CNN architectures, experimenting with larger batch sizes was impractical. To address this limitation, a batch size of 16 was selected for all four models. This configuration struck a balance between computational efficiency and resource constraints, enabling the models to be trained effectively while avoiding resource exhaustion or out-of-memory errors. By adhering to this setup, the training process remained stable, ensuring consistent progress without compromising model performance or hardware limitations.

These modifications, including the implementation of dropout layers and the selection of the Adam optimizer, effectively addressed the overfitting issue and resulted in improved performance on the validation set. Additionally, the chosen batch sizes ensured efficient training while staying within GPU memory constraints. Now, let's delve into the results of each model.

The LUNA22 ISMI dataset is imbalanced, containing nearly twice as many malignant nodules as benign ones. Despite our efforts to balance the dataset through augmentation, the models have learned to distinguish malignant class nodules more accurately than benign ones. To further investigate this, we calculated the specificity of the models for the training, testing, and validation sets.

Specificity: Specificity complements recall by assessing the model's capability to accurately classify negative cases (benign nodules). It is calculated as the ratio of true negatives to the total number of negative samples in the testing data.

$$Specificity = TN/(FP + TN) \qquad \textit{Equation. (5.1)}$$

When we calculated the specificity using Equation 5.1 for the test set, we found the baseline model had a specificity of 74.19%, the AlexNet model had 84.61%, the proposed model had 77.78%, and the proposed model with CBAM achieved 85%. The proposed model with

CBAM not only achieved higher training and testing accuracy but also demonstrated high specificity, indicating its strong ability to correctly identify benign nodules.

The findings of the current study, while showcasing the model's ability for lung nodule classification, do not necessarily reflect the highest performance achievable in this field. As noted in the literature review, other studies using different datasets have reported superior classification results.

A significant limitation of this research is the lack of a publicly available leaderboard for benchmarking performance on the specific dataset used. The absence of established benchmarks makes it difficult to directly compare our model's accuracy and generalizability with existing solutions.

The convolutional layers in the fourth model extract relevant features from the CT scan images. These features may include patterns linked to lung cancer nodules, such as variations in tissue density, texture, or shape.

Attention Mechanisms: The CBAM module significantly enhances feature learning for lung cancer nodule detection. Channel Attention: By concentrating on the most informative channels, the channel attention mechanism enables the model to identify specific features within CT scan images that are highly indicative of lung cancer nodules. This enhances the model's sensitivity to subtle variations in tissue characteristics.

Spatial Attention: By emphasizing the crucial spatial locations within CT scan images, the spatial attention mechanism allows the model to accurately identify regions where lung cancer nodules are likely present. This enhances the model's localization accuracy and helps reduce the number of false positives.

Improved Feature Learning: The CBAM module aids the model in concentrating on the most distinguishing features within CT scan images, resulting in more accurate and robust lung cancer nodule classification. Enhanced Localization: By pinpointing the exact spatial locations of lung cancer nodules, CBAM enhances the model's ability to localize these nodules within the CT scan images.

By suppressing irrelevant features and concentrating on the most informative regions, CBAM helps reduce the number of false positive detections, thereby improving the model's specificity (Reduced False Positives).

CBAM aids the model in learning more generalizable features, which reduces the risk of overfitting and enhances performance on unseen CT scan images (Improved Generalization).

Overall, the 3D CNN model with CBAM is well-suited for lung cancer nodule classification from CT scan images. By effectively leveraging the attention mechanisms and concentrating on the most relevant features and spatial locations, the model can achieve high accuracy and sensitivity in detecting lung cancer nodules.

In our implementation, Grad-CAM was applied to individual slices within CT scan patches. These patches, consisting of 64 axial slices of nodule images, serve as the model's input data. By generating heatmaps for each slice, Grad-CAM visualizes the regions the model focuses on when classifying a specific nodule patch as malignant or benign. These interpretable heat maps offer valuable insights for clinicians and radiologists. By visualizing the model's focus areas within each slice, these heatmaps can potentially *Enhance Decision Support and offer* insights that complement the radiologist's expertise, potentially aiding in the final classification of a nodule as malignant or benign. *Improve Explainability:* Increase the transparency of the model's predictions, allowing clinicians to better understand the rationale behind the model's classification decisions.

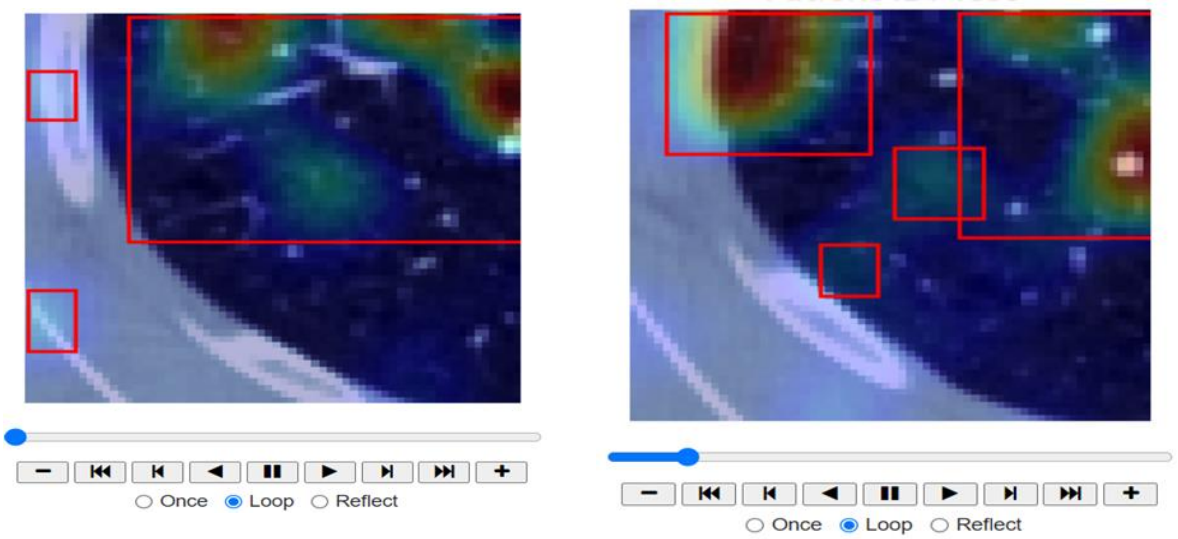
Identify Potential Biases: The heatmaps may uncover any biases within the model, such as an over-reliance on specific image artifacts or features that may not be clinically relevant. While Grad-CAM offers valuable interpretability, it has limitations. Grad-CAM mainly focuses on localized regions that contribute to the classification and might not capture the model's higher-level reasoning processes (Chattopadhyay et al., 2017; Selvaraju et al., 2017). Therefore, it is essential to explore more advanced interpretability techniques that delve deeper into the model's internal workings and feature interactions.

We chose to integrate CBAM into the best of the three models, which was Model 3. Grad-CAM (Gradient-weighted Class Activation Maps) and CBAM (Convolutional Block Attention Module) are powerful techniques that enhance the interpretability of deep learning models. Grad-CAM generates class-specific heatmaps highlighting the most critical regions in an input image that contribute to a given prediction. CBAM, on the other hand, dynamically adjusts feature responses by learning adaptive attention weights (Woo et al., n.d.). Combining Grad-CAM and CBAM provides a deeper understanding of how the model makes decisions. The input image passes through the model's layers, including the CBAM module. CBAM dynamically adjusts the feature responses based on their importance. The

gradient of the target class is computed and propagated backward through the model, with the gradients weighted by the activations of the final convolutional layer.

The weighted gradients are averaged to create a class-specific heatmap, highlighting the regions in the input image that contributed most to the prediction. The benefits of combining Grad-CAM and CBAM include *Enhanced Interpretability:* CBAM's attention mechanism helps Grad-CAM focus on the most relevant features, making the generated heatmaps more informative and easier to understand. *Improved Localization:* The attention weights learned by CBAM guide Grad-CAM to more accurately localize important regions in the input image. *Better Understanding of Model Behavior:* By combining these two techniques, we gain a deeper understanding of the model's decision-making process and the features it focuses on.

Additionally, incorporating these interpretable heatmaps into a user-friendly clinical interface could significantly enhance their practical utility in real-world medical practice. Since the interpretation is conducted for the 3D models, radiologists can thoroughly examine the entire series of CT scan nodules for a deeper investigation. This process is illustrated in Figure 5.9.



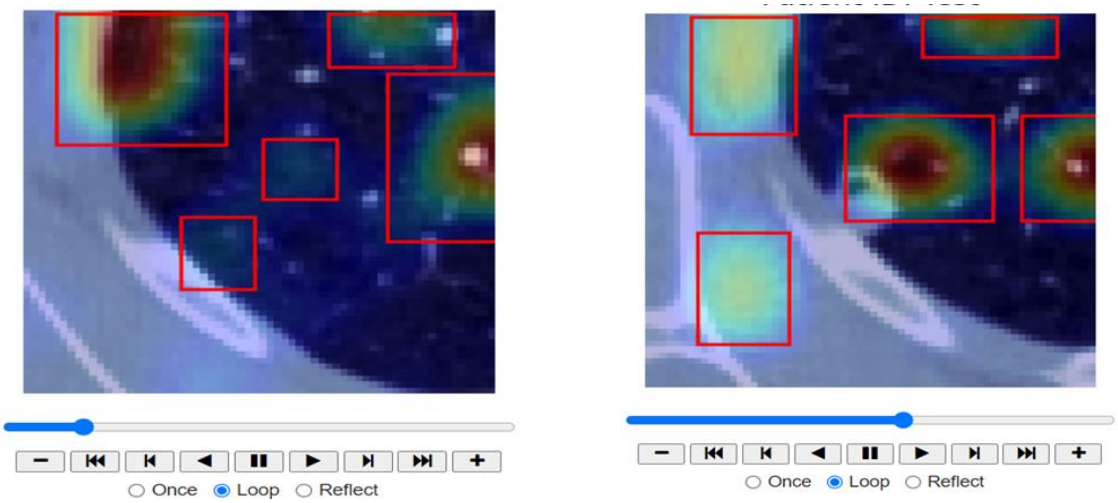


Figure 5.11 3D Grad-CAM Visualizations for the axial slices

5.3.2 Research Question Discussion

An answer for RQ1: This study delves into exploring various techniques and tools to utilize the rich information captured by chest CT scans. A primary focus was on leveraging the inherent 3D nature of this data modality. Unlike simpler image formats, CT scans provide volumetric information crucial for tasks such as lung nodule classification. By employing 3D data as input, we could harness the spatial relationships and contextual information existing between voxels (3D volume elements) within the CT scans. This 3D information holds valuable features for differentiating malignant from benign lung nodules.

To Effectively exploit this 3D information, we developed 3D convolutional neural networks (CNNs). These networks are specifically designed to process volumetric data, allowing them to learn features directly from the 3D CT scans. These learned features can then be used for various applications, including the classification of lung nodules in our case. The 3D CNN approach offers several advantages compared to traditional methods that might rely on pre-processing steps to convert 3D CT scans into simpler 2D representations. These pre-processing steps can potentially lead to information loss and hinder the model's ability to capture the full spectrum of features present in the 3D data. By directly using 3D CNNs, we can:

Preserve Spatial Relationships: Maintain the spatial information between voxels within the CT scan, potentially leading to more robust feature extraction for classification.

Learn 3D Features: The 3D CNN architecture allows the model to learn features directly from the 3D data, potentially capturing intricate or detailed relationships between anatomical structures within the lung. Based on our experiments, our proposed model with CBAM (Convolutional Block Attention Module) performed better than the other models in classifying lung nodules. CBAM helped the model focus on the most relevant features, leading to improved classification performance.

An answer for RQ2: The proposed model demonstrates strong performance for early-stage lung cancer detection and classification by leveraging a carefully designed neural network architecture tailored for medical imaging. Through our research, the **3D Convolutional Neural Network (3D CNN)** combined with **attention mechanisms** proved to be the most effective. This architecture is particularly adept at capturing spatial and volumetric features critical for identifying subtle early-stage lung nodules. Key performance metrics achieved: **Accuracy**: High precision in detecting and classifying malignant and benign nodules. **Sensitivity**: Reliable detection of even small nodules, crucial for early-stage diagnosis. **Specificity**: Minimization of false positives to prevent unnecessary interventions. The model integrates explainable AI (XAI) methods, such as Grad-CAM, to provide interpretable results, enhancing trust and usability in clinical settings. These results highlight the architecture's capability to meet the dual objectives of accuracy and transparency, making it a robust solution for early-stage lung cancer detection.

Answer for RQ3: Integrating Explainable AI (XAI) in Lung Cancer Detection Models Context: Developing robust lung cancer detection models faces the challenge of interpretability, especially in deep learning models, which often act as "black boxes." Integrating Explainable AI (XAI) methods is essential to build trust among clinicians and patients.

1. **Importance of Explainability: Trust and Adoption** Clinicians are more likely to adopt AI systems that can justify their decisions. **Error Analysis:** Identify weaknesses in model predictions, such as misclassified nodules. **Ethical Considerations:** Ensure compliance with ethical and regulatory standards.

2. **XAI Methods for Interpretability: Feature Attribution Methods; Grad-CAM:** Generates heatmaps highlighting regions of CT scans that influence the model's prediction. **Integrated Gradients:** Measures each pixel's contribution to the model's output. **Surrogate Models:** Simple models (e.g., decision trees) that approximate

complex models and provide human-readable logic. **Rule-Based Explanations:** Provide textual explanations based on extracted rules. **Counterfactual Explanations:** Highlight how slight input changes could alter predictions.

3. **Integration into Research Framework: Preprocessing Data;** Segment lung regions and standardize data for consistency. **Model Architecture:** Include attention mechanisms and interpretable constraints. **Post-Hoc Explanation:** Use Grad-CAM or Integrated Gradients for visual overlays and generate textual explanations. **User Interface:** Display explanations alongside predictions with visual outputs (e.g., heatmaps) and textual descriptions.

4. **Validation and Evaluation: Clinical Validation;** Collaborate with radiologists and oncologists. **Metrics for Evaluation:** Use faithfulness and comprehensibility metrics. **Real-World Testing:** Evaluate explanations in clinical scenarios.

5. **Addressing Challenges: Complexity and Simplicity;** Use a combination of visual and textual explanations. **Computational Cost:** Optimize workflows with parallel processing. **Clinical Relevance:** Focus on medically relevant features with domain expert consultation.

6. **Benefits of Integration: Improved Trust;** Gain confidence in the AI model's predictions. **Enhanced Decision-Making:** Validate or refine diagnoses. **Regulatory Compliance:** transparent models are likely to pass evaluations. **Wider Adoption:** Explainable systems are more likely to be used in healthcare settings.

Conclusion: Integrating XAI methods enhances the interpretability of AI-driven medical systems, bridging the gap between AI capabilities and clinical applicability and ensuring accurate and actionable model predictions.

CHAPTER SIX

CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

This research explored the use of 3D convolutional neural networks (CNNs) for classifying lung nodules. By utilizing a 3D CNN architecture, we were able to extract features directly from the volumetric CT scan data to achieve reliable and accurate differentiation between malignant and benign nodules.

To understand the decision-making processes of the 3D CNN model, we integrated Gradient-weighted Class Activation Mapping (Grad-CAM) for interpretability analysis. Grad-CAM produces heatmaps that highlight the specific regions within a CT scan slice that play a crucial role in the model's classification of a particular nodule patch. These visual insights can assist healthcare professionals in comprehending the model's reasoning behind its predictions, thereby enhancing trust and facilitating informed decision-making.

To address the challenge of class imbalance and improve the model's performance, we utilized data augmentation and fine-tuning techniques. While there is still room for further refinement, our proposed 3D CNN model demonstrated promising results, achieving accuracies of 94.06% on the testing set and 88.06% on the validation set.

It's essential to recognize the inherent challenges in medical image analysis. Medical images, such as CT scans, are often detailed and show subtle variations that can be challenging for models to differentiate, especially between benign and malignant nodules. Furthermore, the 3D nature of CT scans adds a layer of complexity compared to simpler 2D images, which may require more advanced models and larger datasets to achieve optimal performance.

The proposed 3D CNN model with CBAM achieved an impressive AUC of 98.84 and an F1-score of 95.56 on the test set. These metrics indicate the model's strong potential for accurate lung nodule classification. Importantly, the high recall value of 94.99 signifies a low rate of missed malignant nodules, which is critical in clinical settings. The performance on the test set demonstrates good generalizability. The consistency between validation and test set results suggests that the model is not overfitting to the training data and can perform well on unseen data. While these metrics are encouraging, it's important to recognize potential limitations. A larger dataset could further improve generalizability. Additionally,

comparing these results with those from other models or established benchmarks would provide a more comprehensive understanding of the 3D CNN model's relative strengths and weaknesses.

6.2 Future Work

Building on the promising outcomes demonstrated by the 3D CNN models, future research can delve into several strategies to enhance their performance and generalizability further. A more comprehensive analysis of the class activation maps (CAMs) generated using Grad-CAM could significantly contribute to the explainability of the model. Collaborating with radiologists to scrutinize these activation maps can provide critical insights into the regions the model emphasizes during lung nodule classification. Understanding these focal areas can uncover the model's strengths and highlight potential limitations. This valuable feedback can then be used to optimize the network architecture or adjust the training process, ultimately aiming to improve classification accuracy and reliability.

Leveraging Local Datasets and Expert Knowledge: The availability of a locally collected dataset, gathered in collaboration with healthcare professionals, offers a valuable opportunity. By strategically combining this local dataset with publicly available datasets, we can create a more diverse and robust data pool for model training. This diversification can enhance the model's ability to generalize to unseen data and improve its performance across various nodule types and appearances.

Addressing Class Imbalance: As noted earlier, class imbalance within the training data can negatively affect the model's ability to effectively learn features from the under-represented class (benign nodules). To improve classification accuracy for both malignant and benign nodules, balancing the class distribution during training could be a key strategy. By ensuring a more equal representation of both classes, the model may develop a more comprehensive understanding of the distinguishing features for each type of nodule, leading to better overall performance.

The field of deep learning is continuously advancing, with new and enhanced architectures regularly being developed. Exploring alternative 3D CNN architectures specifically designed for medical image analysis could significantly improve feature extraction capabilities for lung nodule classification. Furthermore, integrating attention mechanisms

within the network architecture might enable the model to focus on the most discriminative regions within the CT scans, thereby boosting classification accuracy.

Close collaboration with radiologists throughout the research process is essential. Their expertise plays a vital role in interpreting Class Activation Maps (CAMs), as they can help identify the image features that the model prioritizes for classification. With their clinical knowledge, radiologists can assess the relevance of these features, ensuring that the model's focus aligns with what is important for accurate diagnosis. This collaboration not only enhances the model's interpretability but also ensures its practical applicability in clinical settings.

Validating Model Performance: Engaging radiologists in the evaluation process can provide essential feedback on the model's performance within a real-world clinical context.

Guiding future research directions through collaboration with healthcare professionals is crucial. Their insights can help pinpoint areas where the model can be further optimized, ensuring that it meets the practical needs of clinical environments. By working together, we can refine the model’s accuracy, interpretability, and overall utility, ultimately making it a more effective tool for early-stage lung cancer detection and improving patient care outcomes.

By applying these strategies and collaborating with healthcare professionals, we aim to develop a highly accurate and generalizable 3D CNN model for lung nodule classification. This effort ultimately enhances diagnosis and improves patient care in the future.

REFERENCES

Alghamdi, H. S. (2022). Towards Explainable Deep Neural Networks for the Automatic Detection of Diabetic Retinopathy. *Applied Sciences (Switzerland)*, *12*(19). https://doi.org/10.3390/app12199435

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., Guidotti, R., Del Ser, J., Díaz-Rodríguez, N., & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, *99*, 101805. https://doi.org/https://doi.org/10.1016/j.inffus.2023.101805

American College of Radiology. (2023). *Lung cancer screening (LCS) - LungRADS.* https://www.acr.org/Clinical-Resources/Reporting-and-Data-Systems/Lung-Rads

B. C., K., & K. B., N. (2023). An Approach of AlexNet CNN Algorithm Model for Lung Cancer Detection and Classification. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(11s), 49–54. https://doi.org/10.17762/ijritcc.v11i11s.8069

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, *10*(7). https://doi.org/10.1371/journal.pone.0130140

Bao, L., Bao, T., Zheng, Y., & Xia, J. (2020, July 21). A Simple Residual Network for Lung Nodule Classification. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3403782.3403808

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012

Brett, M., Markiewicz, C. J., Hanke, M., Côté, M.-A., Cipollini, B., McCarthy, P., Jarecka, D., Cheng, C. P., Larson, E., Halchenko, Y. O., Cottaar, M., Ghosh, S., Wassermann, D., Gerhard, S., Lee, G. R., Baratz, Z., Wang, H.-T., Papadopoulos Orfanos, D.,

Kastman, E., … freec84. (2024). *nipy/nibabel: 5.2.1*. Zenodo. https://doi.org/10.5281/zenodo.10714563

Bushara, A. R., & Kumar, R. S. V. (2022). Deep Learning-based Lung Cancer Classification of CT Images using Augmented Convolutional Neural Networks. *Electronic Letters on Computer Vision and Image Analysis*, *21*(1), 130–142. https://doi.org/10.5565/REV/ELCVIA.1490

Causey, J. L., Li, K., Chen, X., Dong, W., Walker, K., Qualls, J. A., Stubblefield, J., Moore, J. H., Guan, Y., & Huang, X. (2022). Spatial Pyramid Pooling With 3D Convolution Improves Lung Cancer Detection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(2), 1165–1172. https://doi.org/10.1109/TCBB.2020.3027744

Chattopadhyay, A., Sarkar, A., Balasubramanian, V., Howlader, P., & Balasubramanian, V. N. (2017). *Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks*. https://doi.org/10.48550/arXiv.1710.11063

Conor O'Sullivan. (n.d.). *XAI with Python*. A Dat Odyssey. Retrieved January 27, 2024, from adataodyssey.com

Dr. Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python Starter Bundle 1st Edition (1.1.0)*.

Essaf, F., Li, Y., Sakho, S., & Gadosey, P. K. (2020). Improved Convolutional Neural Network for Lung Cancer Detection. *ACM International Conference Proceeding Series*, 48–54. https://doi.org/10.1145/3398329.3398337

Fchollet, & Team Keras. (2019). Keras documentation: The Functional API. In *Keras, Complete guide to the functional API.* https://keras.io/guides/functional_api/

Gebremariam, T. H., Haisch, D. A., Fernandes, H., Huluka, D. K., Binegdie, A. B., Woldegeorgis, M. A., Ergetie, W., Worku, A., Zerihun, L. M., Cohen, M., Massion, P. P., Sherman, C. B., Saqi, A., & Schluger, N. W. (2021). Clinical Characteristics and Molecular Profiles of Lung Cancer in Ethiopia. *JTO Clinical and Research Reports*, *2*(7). https://doi.org/10.1016/j.jtocrr.2021.100196

Giard, C. (2023). *Radiology Workflow | Efficiencies*. https://enlitic.com/blogs/radiology-workflow-efficiencies/

Gierada, D. S., Black, W. C., Chiles, C., Pinsky, P. F., & Yankelevitz, D. F. (2020). Low-dose CT screening for lung cancer: Evidence from 2 decades of study. In *Radiology: Imaging Cancer* (Vol. 2, Issue 2). Radiological Society of North America Inc. https://doi.org/10.1148/rycan.2020190058

Guerra-Manzanares, A., Lopez, L. J. L., Maniatakos, M., & Shamout, F. E. (2023). *Privacy-preserving machine learning for healthcare: open challenges and future perspectives*. https://doi.org/10.1007/978-3-031-39539-0_3

Heindl, Dr. A. (2022). *What's the Difference Between DICOM and NIfTI?* https://encord.com/blog/dicom-and-nifti-comparison/

IARC, W. H. O. (2022). *Cancer*. https://www.who.int/news-room/fact-sheets/detail/cancer

IARC, W. H. O. (2024). *Cancer Tomorrow*. https://gco.iarc.who.int/tomorrow/en

Islam, S. K. M. S., Nasim, M. A. Al, Hossain, I., Ullah, Dr. M. A., Gupta, Dr. K. D., & Bhuiyan, M. M. H. (2023). *Introduction of Medical Imaging Modalities*. http://arxiv.org/abs/2306.01022

Jiang, W., Cai, G., Hu, P., & Wang, Y. (2021). Personalized medicine of non-gene-specific chemotherapies for non-small cell lung cancer. In *Acta Pharmaceutica Sinica B* (Vol. 11, Issue 11, pp. 3406–3416). Chinese Academy of Medical Sciences. https://doi.org/10.1016/j.apsb.2021.02.003

Jiang, W., Zeng, G., Wang, S., Wu, X., & Xu, C. (2022). Application of Deep Learning in Lung Cancer Imaging Diagnosis. In *Journal of Healthcare Engineering* (Vol. 2022). Hindawi Limited. https://doi.org/10.1155/2022/6107940

Kenton, W. (2023). *What Is a Black Box Model? Definition, Uses, and Examples*. https://www.investopedia.com/terms/b/blackbox.asp#citation-5

Keras. (2019). *Home - Keras Documentation*. https://keras.io/

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. https://doi.org/10.1038/nature14539

Li, X., Morgan, P. S., Ashburner, J., Smith, J., & Rorden, C. (2016). The first step for neuroimaging data analysis is DICOM to NIfTI conversion. *Journal of Neuroscience Methods*, *264*, 47–56. https://doi.org/10.1016/j.jneumeth.2016.03.001

Lin, Z., Zheng, J., & Hu, W. (2020). Using 3D convolutional networks with shortcut connections for improved lung nodules classification. *ACM International Conference Proceeding Series*, 42–49. https://doi.org/10.1145/3404512.3404525

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. http://arxiv.org/abs/1705.07874

Luo, G., Zhang, Y., Etxeberria, J., Arnold, M., Cai, X., Hao, Y., & Zou, H. (2023). Projections of Lung Cancer Incidence by 2035 in 40 Countries Worldwide: Population-Based Study. *JMIR Public Health and Surveillance*, *9*, e43651. https://doi.org/10.2196/43651

Markotić, V., Pojužina, T., Radančević, D., Miljko, M., & Pokrajčić, V. (2021). The Radiologist Workload Increase: Where Is the Limit?: Mini Review and Case Study. *Psychiatria Danubina*, *33*, 768–770. https://pubmed.ncbi.nlm.nih.gov/34718316/#:~:text=Daily%20average%20of%20analyzed%20imaging

Mason, D., scaramallion, mean-Bremen, Braxton, Suever, J., Vanessasaurus, Orfanos, D. P., Lemaitre, G., Panchal, A., Rothberg, A., Herrmann, M. D., Massich, J., Kerns, J., van Golen, K., Robitaille, T., Biggs, S., Moloney, Bridge, C., Shun-Shin, M., … Wortmann, J. (2023). *pydicom/pydicom: Pydicom v2.4.0*. Zenodo. https://doi.org/10.5281/zenodo.8034250

Mayekar, N., Pattewar, S., Patil, S., & Dhruv, A. (2022). Preliminary Lung Cancer Detection using Deep Neural Networks. *International Research Journal of Engineering and Technology*. www.irjet.net

Mayo, J. R. (2009). *CT Evaluation of Diffuse Infiltrative Lung Disease Dose Considerations and Optimal Technique*. www.thoracicimaging.com

McKelvey, T., Ahmad, M., Teredesai, A., & Eckert, C. (2018). *Interpretable Machine Learning in Healthcare*.

Mi, X., Zou, B., Zou, F., & Hu, J. (2021). Permutation-based identification of important biomarkers for complex diseases via machine learning models. *Nature Communications*, *12*(1). https://doi.org/10.1038/s41467-021-22756-2

Molnar, C. (2022). *Interpretable machine learning : a guide for making black box models explainable* (2nd ed.). christophm.github.io/interpretable-ml-book/

National Cancer Institute, National Institutes of Health, & U.S. Department of Health and Human Services. (2021, October 11). *What Is Cancer?* National Cancer Institute: Patient Education Publications. https://www.cancer.gov/about-cancer/understanding/what-is-cancer

NEMA PS3 / ISO 12052, Digital Imaging and Communications in Medicine (DICOM) Standard, N. E. M. A., & National Electrical Manufacturers Association. (2024). *About DICOM: Overview*. ISO 12052 Standard. http://www.dicomstandard.org/

Panesar, A. (2019). Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes. In *Machine Learning and AI for Healthcare: Big Data for Improved Health Outcomes*. Apress Media LLC. https://doi.org/10.1007/978-1-4842-3799-1

Ramana, K., Kumar, M. R., Sreenivasulu, K., Gadekallu, T. R., Bhatia, S., Agarwal, P., & Idrees, S. M. (2022). Early Prediction of Lung Cancers Using Deep Saliency Capsule and Pre-Trained Deep Learning Frameworks. *Frontiers in Oncology*, *12*. https://doi.org/10.3389/fonc.2022.886739

Rani, S., Ghai, D., Kumar, S., Kantipudi, M. P., Alharbi, A. H., & Ullah, M. A. (2022). Efficient 3D AlexNet Architecture for Object Recognition Using Syntactic Patterns from Medical Images. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/7882924

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 6). Springer. https://doi.org/10.1007/s42979-021-00815-1

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-October*, 618–626. https://doi.org/10.1109/ICCV.2017.74

Shady Hermena, & Michael Young. (2023, August 8). *CT-scan Image Production Procedures*. National Library of Medicine. https://www.ncbi.nlm.nih.gov/books/NBK574548/

Shafi, I., Din, S., Khan, A., Díez, I. D. L. T., Casanova, R. del J. P., Pifarre, K. T., & Ashraf, I. (2022). An Effective Method for Lung Cancer Diagnosis from CT Scan Using Deep Learning-Based Support Vector Network. *Cancers*, *14*(21). https://doi.org/10.3390/cancers14215457

Sori, W. J., Feng, J., & Liu, S. (2019). Multi-path convolutional neural network for lung cancer detection. *Multidimensional Systems and Signal Processing*, *30*(4), 1749–1768. https://doi.org/10.1007/s11045-018-0626-9

Speith, T. (2022). A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods. *ACM International Conference Proceeding Series*, 2239–2250. https://doi.org/10.1145/3531146.3534639

Stiglic, G., Kocbek, P., Fijacko, N., Zitnik, M., Verbert, K., & Cilar, L. (2020). Interpretability of machine learning-based prediction models in healthcare. In *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (Vol. 10, Issue 5). Wiley-Blackwell. https://doi.org/10.1002/widm.1379

Sujatha, M., & Prabhakar, S. (2019). Issue 3 www.jetir.org (ISSN-2349-5162). In *JETIRAU06101 Journal of Emerging Technologies and Innovative Research* (Vol. 6). JETIR. www.jetir.org

Sun, B., Liu, F., Zhou, Y., Jin, S., Li, Q., & Jin, X. (2020, October 20). Classification of Lung Nodules Based on GAN and 3D CNN. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3424978.3425094

Tandon, R., Agrawal, S., Chang, A., & Band, S. S. (2022). VCNet: Hybrid Deep Learning Model for Detection and Classification of Lung Carcinoma Using Chest Radiographs. *Frontiers in Public Health*, *10*. https://doi.org/10.3389/fpubh.2022.894920

TensorFlow. (2019). *TensorFlow*. Google. https://www.tensorflow.org/

Venkadesh, K. V., & Jacobs, C. (2022). *LUNA22-ISMI*. Zenodo. https://doi.org/10.5281/zenodo.6559584

Whittaker, M. (2019). *Disability, Bias, and AI*.

WHO, I. (2022). *Cancer*. https://www.who.int/news-room/fact-sheets/detail/cancer

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (n.d.). *CBAM: Convolutional Block Attention Module*.

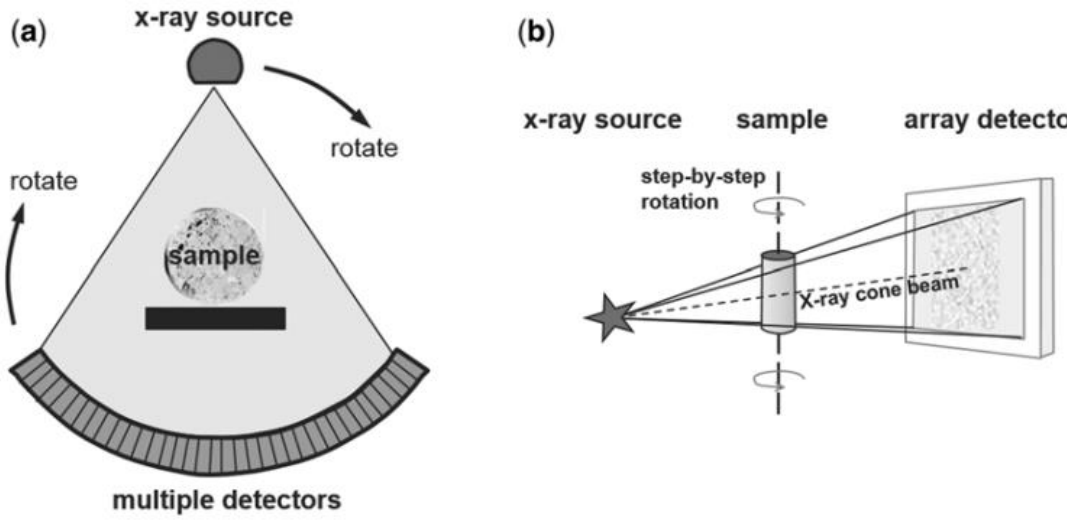
Zunair, H., Rahman, A., Mohammed, N., & Cohen, J. P. (2020). *Uniformizing Techniques to Process CT scans with 3D CNNs for Tuberculosis Prediction*. http://arxiv.org/abs/2007.13224

APPENDICES

A. CT Scan image - Advanced X-ray Imaging

CT scans employ advanced detectors to simultaneously capture numerous X-ray images from different angles, enabling the creation of highly detailed 3D representations of the body. Enhancements such as multi-energy imaging and spatial filtering significantly improve image clarity, while dose modulation techniques are used to reduce radiation exposure effectively (Islam et al., 2023). The resulting images can be presented in diverse formats, including cross-sectional views, 3D visualizations, and even immersive virtual reality models.

The CT scanner features a donut-shaped structure with a motorized table where the patient lies. This table moves into the scanner, where an internal X-ray tube rotates around the patient, emitting X-ray beams from multiple angles. As the beams pass through the body, they are absorbed at varying levels by different tissues, creating a pattern of attenuated X-rays. Detectors located opposite the X-ray tube capture these patterns and convert them into electrical signals. These signals are then processed by a computer using advanced algorithms to construct a detailed three-dimensional image of the body (Islam et al., 2023).



Principle of Computed Tomography Imaging

CT scan images differ significantly from standard JPEG images, as they are stored in specialized formats such as DICOM and NIfTI. These formats are designed to handle the complex data captured by CT scanner detectors. The detectors in a CT scanner collect raw imaging data by measuring the attenuated X-rays after they pass through the body. This data

is then transmitted to the image processor, where it is converted into meaningful images using advanced computational techniques, ensuring accurate representation and compatibility with medical imaging standards.

CT scan Hardware:

- **X-ray Generator:** Produces the X-ray beam.
- **X-ray Tube:** Converts electricity into X-rays.
- **Gantry:** Houses the X-ray tube and detectors, rotating around the patient for 360-degree views.
- **Patient Table:** Moves the patient through the gantry during the scan.
- **Photon Detectors:** Capture and measure X-rays passing through the body.
- **Shielding Elements:** Absorb scattered X-rays, minimizing noise and radiation exposure.
- **Image Processor:** Processes the raw scan data.
- **Console:** Controls the entire scanning process.

B. Important Factors to Consider with CT-scan Images

Image Windowing: When viewing CT images, the human eye can differentiate fewer shades of gray, even though Hounsfield Unit (HU) values are expressed on a 5000-unit scale. The displayed greyscale brightness and contrast can be adjusted by modifying the number of included HUs (referred to as the "window") and/or the HU value set as the central or middle value (referred to as the "level"). The image viewer can manually change these settings by interacting with the image software using a mouse. "Windowing" or "changing the window" means altering what is shown in the image by shifting the visible HU range (Shady Hermena & Michael Young, 2023). The window width determines the image contrast. As the width increases, the contrast decreases. The range of HUs displayed in the image is determined by the window width. For structures with similar attenuation values, a small window (50-350 HU) is appropriate. For structures with significantly varied attenuation values, a large window (400-2000 HU) is suitable.

Level: The brightness of the image depends on the window level. As the level increases, so does the brightness. By adjusting the window level, the central or midpoint gray value for the HU range displayed in the image can be set. The brightness of the image increases with higher window levels and decreases with lower levels. The window level also affects the

optimal imaging of various tissues (Shady Hermena & Michael Young, 2023). Denser tissues require a higher level, while less dense tissues require a lower level.

The wider the window, the more densities are seen but less contrast. In a narrow window, fewer densities are seen but more contrast. Window (W) is how many HU within 256 shades of grey, while Level (L) is where the window is centered. Anything lower than the window is going to appear black, and anything higher than the window is going to look white. {i.e. < L – 1/2W = black and > L + 1/2W = white}. For the lung, the window is commonly +1500, and the level is commonly -700.

Planes and Orientations: The capacity to fully rebuild the pictures in the **Axial or Transversal, sagittal,** and **Coronal or Frontal** planes is one benefit of obtaining a volume acquisition CT scan. When assessing the degree of disease in a patient, it is especially useful to view the anatomy and pathology in all three planes (Mayo, 2009). While

Anterior refers to the front of the body, closer to the head.

Posterior refers to the back of the body, closer to the rear.

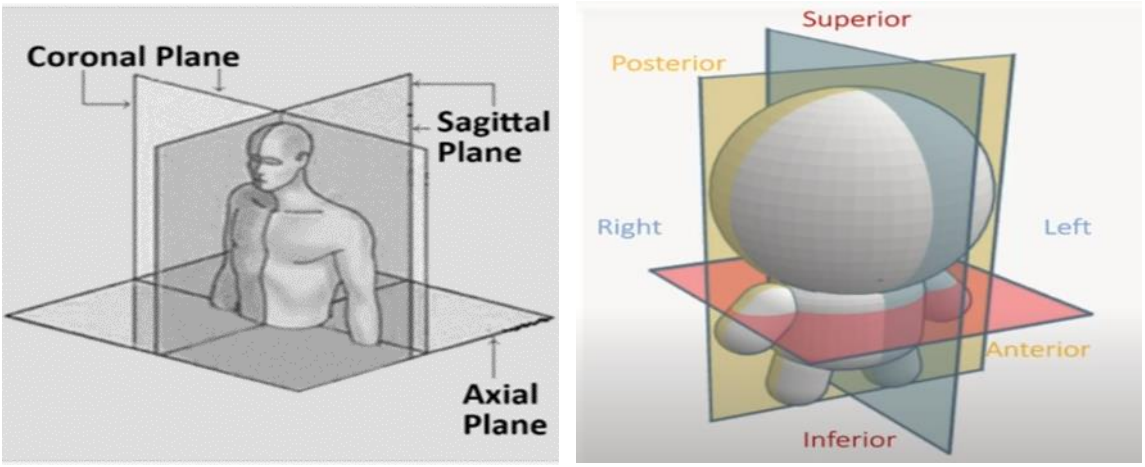
CT scan images can be viewed in various formats:

Axial Slices: These are cross-sectional images, like slices of bread, providing a horizontal view of the body at different levels.

Coronal Slices: These slices are oriented vertically, providing a view from the front to the back of the body.

Sagittal Slices: These slices are also vertical but oriented from side to side, offering a view from the left to the right of the body.

3D Reconstructions: These are computer-generated 3D models of the scanned body, allowing for visualization from any angle.



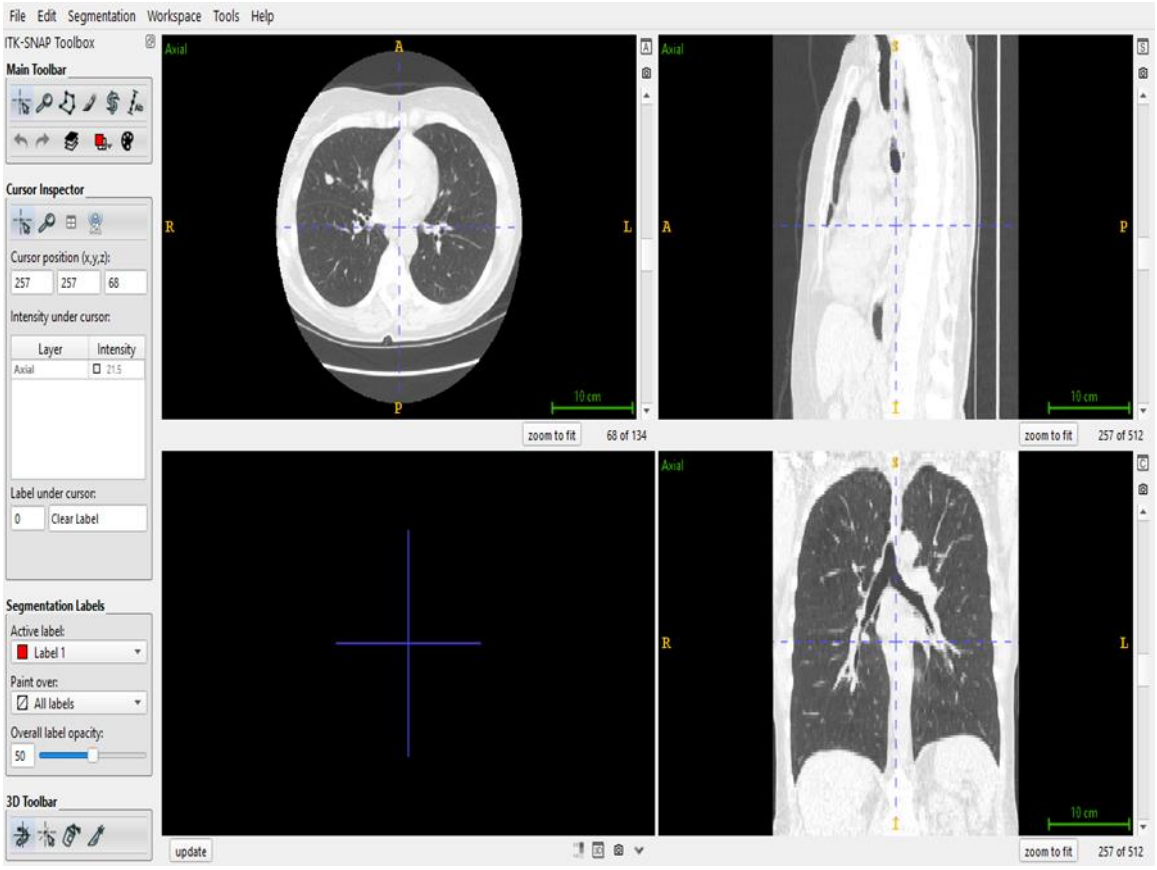
Planes and Orientations and Anatomical Orientation

C. How Do Low-Dose CT Scans Show Whether We Have Lung Cancer?

Low-dose CT (LDCT) scans, unlike traditional CT scans, use a significantly lower radiation dose, making them a safer option for screening (National Cancer Institute et al., 2021). These scans are particularly effective at visualizing small nodules in the lungs, which may indicate early-stage lung cancer (Gierada et al., 2020). Standard chest X-rays often lack the sensitivity to detect such subtle nodules. The ability of LDCT to identify these early lesions allows for intervention at a more treatable stage, potentially improving patient outcomes. While LDCT aids in visualization, interpreting the findings requires a structured approach. Radiologists rely on established guidelines, such as the Lung-RADS classification system, to ensure consistent interpretation across healthcare settings. These guidelines provide a framework for evaluating nodule characteristics, including size, number, and morphology (solid, part-solid, or non-solid). Based on this evaluation, a Lung-RADS score (1-4) is assigned to the scan (American College of Radiology, 2023).

- Lung-RADS 1: No nodules were identified, indicating a low suspicion of cancer. Follow-up screening is typically recommended in one year (National Cancer Institute et al., 2021).
- Lung-RADS 2: A small, non-concerning nodule is present. Similar to Lung-RADS 1, follow-up screening at one year is often advised (National Cancer Institute et al., 2021).
- Lung-RADS 3: A small nodule with features that warrant closer monitoring. In this case, a follow-up scan at six months might be recommended (National Cancer Institute et al., 2021).
- Lung-RADS 4: This category encompasses nodules with a higher suspicion of malignancy based on their appearance on the CT scan.

While LDCT can provide valuable information regarding nodule characteristics, it's important to recognize its limitations. Distinguishing benign from malignant nodules based solely on imaging can be challenging, and additional investigations, such as biopsies, may be necessary for a definitive diagnosis (National Cancer Institute et al., 2021). The textural characteristics of nodules, categorized as solid, part-solid, or non-solid on LDCT scans, can offer clues regarding their potential malignancy (Sujatha & Prabhakar, 2019). Solid nodules, appearing opaque on the scan, are generally considered more suspicious for cancer compared to non-solid nodules, where underlying lung tissue can be visualized (American College of Radiology, 2023). Part-solid nodules exhibit a mixed appearance, with both solid and non-solid components. These require careful evaluation and may warrant further investigation depending on their specific features (Gierada et al., 2020).



D. Model Architecture Summaries of the Four Models Used

The number of parameters for each layer for all three models can be calculated using the formulas in *equation (F.1)* and *equation (F.2).*

✦ **For the convolutional layer**:

$$\#\ param = (w * h * d * Pf * + 1) * Cf \qquad \textit{Equation (F.1)}$$

✦ **For FC-layer:**

$$\#\ param = (Pn + 1) * Cn \qquad \textit{Equation. (F.2)}$$

Where "# param" is the total number of parameters in the layer, "w" is filter width, "h" is filter height, "d" is filter depth, "Pf" is number of filters of the previous layer, "Cf" is number of filters of the current layer, "Pn" is previous layer number of neurons, "Cn" is current layer number of neurons. ('None' in the output shape denotes the batch size)

Baseline Model / Model-1 Model Summary:

Layer	**Output Shape**	**Param #**
Input Layer	[(None, 64, 64, 64, 1)]	0
Conv3D	(None, 63, 63, 63, 64)	576
Max Pooling 3D	(None, 31, 31, 31, 64)	0
Batch Normalization	(None, 31, 31, 31, 64)	256
Conv3D	(None, 30, 30, 30, 64)	32,832
Max Pooling 3D	(None, 15, 15, 15, 64)	0
Batch Normalization	(None, 15, 15, 15, 64)	256
Conv3D	(None, 14, 14, 14, 128)	65,664
Max Pooling 3D	(None, 7, 7, 7, 128)	0
Batch Normalization	(None, 7, 7, 7, 128)	512
Conv3D	(None, 6, 6, 6, 256)	262,400
Max Pooling 3D	(None, 3, 3, 3, 256)	0
Batch Normalization	(None, 3, 3, 3, 256)	1,024
Global Average Pooling 3D	(None, 256)	0
Dense	(None, 1024)	263,168
Dropout	(None, 1024)	0
Dense	(None, 1)	1,025
Total params: 627,713 (2.39 MB) Trainable params: 626,689 (2.39 MB) Non-trainable params: 1,024 (4.00 KB)		

3D AlexNet Model / Model -2 Model Summary:

Layer	Output Shape	Param #
Input Layer	[(None, 64, 64, 64, 1)]	0
Conv3D	(None, 30, 30, 31, 96)	7,296
Batch Normalization	(None, 30, 30, 31, 64)	384
Max Pooling 3D	(None, 28, 28, 29, 96)	0
Conv3D	(None, 26, 26, 27, 128)	331,904
Batch Normalization	(None, 26, 26, 27, 128)	512
Max Pooling 3D	(None, 12, 12, 13, 128)	0
Conv3D	(None, 12, 12, 13, 256)	884,992
Batch Normalization	(None, 12, 12, 13, 256)	1,024
Conv3D	(None, 12, 12, 13, 384)	2,654,592
Batch Normalization	(None, 12, 12, 13, 384)	1,536
Conv3D	(None, 12, 12, 13, 256)	2,654,464
Batch Normalization	(None, 12, 12, 13, 256)	1,024
Max Pooling 3D	(None, 6, 6, 6, 256)	0
Conv3D	(None, 3, 3, 3, 256)	1,769,728
Batch Normalization	(None, 3, 3, 3, 256)	1,024
Flatten	(None, 6912)	0
Dense	(None, 4096)	28,315,648
Batch Normalization	(None, 4096)	16,384
Dense	(None, 1024)	4,195,328
Batch Normalization	(None, 1024)	4,096
Dropout	(None, 1024)	0
Dense	(None, 1)	1,025
Total params: 40,840,961 (155.80 MB) Trainable params: 40,827,969 (155.75 MB) Non-trainable params: 12,992 (50.75 KB)		

Proposed 3D CNN Model / Model-3 Model Summary:

Layer	Output Shape	Param #
Input Layer	[(None, 64, 64, 64, 1)]	0
Conv3D	(None, 63, 63, 63, 64)	576
Max Pooling 3D	(None, 31, 31, 31, 64)	0
Batch Normalization	(None, 31, 31, 31, 64)	256
Conv3D	(None, 30, 30, 30, 128)	65,664
Max Pooling 3D	(None, 15, 15, 15, 128)	0
Batch Normalization	(None, 15, 15, 15, 128)	512
Conv3D	(None, 14, 14, 14, 128)	131,200
Max Pooling 3D	(None, 7, 7, 7, 128)	0
Batch Normalization	(None, 7, 7, 7, 128)	512
Conv3D	(None, 6, 6, 6, 128)	131,200
Max Pooling 3D	(None, 3, 3, 3, 128)	0
Batch Normalization	(None, 3, 3, 3, 128)	512
Flatten	(None, 3456)	0
Batch Normalization	(None, 3456)	13, 824
Dense	(None, 4096)	14,159,872
Dense	(None, 4096)	16,781,312
Dropout	(None, 4096)	0
Dense	(None, 1)	4,097
Total params: 31,289,537 (119.36 MB) Trainable params: 31,281,729 (119.33 MB) Non-trainable params: 7,808 (30.50 KB)		

Proposed 3D CNN Model with CBAM / Model-4 Model Summary:

Layer	Output Shape	Param #	Connected to
Input Layer	[(None, 64, 64, 64, 1)]	0	-
Conv3D-1	(None, 63, 63, 63, 64)	576	Input Layer
Max Pooling 3D-1	(None, 31, 31, 31, 64)	0	Conv3D-1
Batch Normalization-1	(None, 31, 31, 31, 64)	256	Max Pooling 3D-1
Conv3D-2	(None, 30, 30, 30, 128)	65,664	Batch Normalization-1
Max Pooling 3D-2	(None, 15, 15, 15, 128)	0	Conv3D-2
Batch Normalization-2	(None, 15, 15, 15, 128)	512	Max Pooling 3D-2
Conv3D-3	(None, 14, 14, 14, 128)	131,200	Batch Normalization-2
Max Pooling 3D-3	(None, 7, 7, 7, 128)	0	Conv3D-3
Batch Normalization-3	(None, 7, 7, 7, 128)	512	Max Pooling 3D-3
Conv3D-4	(None, 6, 6, 6, 128)	131,200	Batch Normalization-3
Max Pooling 3D-4	(None, 3, 3, 3, 128)	0	Conv3D-4
Batch Normalization-4	(None, 3, 3, 3, 128)	512	Max Pooling 3D-4
Global Average Pooling 3D-1	(None, 128)	0	Batch Normalization-4
Global Max Pooling 3D-1	(None, 128)	0	Batch Normalization-4
Reshape-1	(None, 1, 1, 1, 128)	0	Global Average Pooling 3D-1
Reshape-2	(None, 1, 1, 1, 128)	0	Global Max Pooling 3D-1
Dense-1	(None, 1, 1, 1, 16)	2,064	Reshape-1
Dense-2	(None, 1, 1, 1, 16)	2,064	Reshape-2
Dense-3	(None, 1, 1, 1, 128)	2,176	Dense-1
Dense-4	(None, 1, 1, 1, 128)	2,176	Dense-2
Add-1	(None, 1, 1, 1, 128)	0	Dense-1 and Dense-2
Activation	(None, 1, 1, 1, 128)	0	Add-1
Multiply-1	(None, 1, 1, 1, 128)	0	Batch Normalization-4 and Activation
Global Average Pooling 3D-2	(None, 128)	0	Multiply-1
Global Max Pooling 3D-2	(None, 128)	0	Multiply-1
Add-2	(None, 128)	0	Global Average Pooling 3D-1 and Global Max Pooling 3D-2
Reshape-3	(None, 1, 1, 1, 128)	0	Add-2
Conv3D-5	(None, 1, 1, 1, 1)	43,905	Reshape-3
Multiply-2	(None, 3, 3, 3, 3)	0	Multiply-1 and Conv3D-5
Batch Normalization-5	(None, 3, 3, 3, 128)	512	Multiply-2
Flatten	(None, 3456)	0	Batch Normalization-5
Batch Normalization-6	(None, 3456)	13, 824	Flatten
Dense-5	(None, 4096)	14,159,872	Batch Normalization-6
Dense-6	(None, 4096)	16,781,312	Dense-5
Dropout	(None, 4096)	0	Dense-6
Dense-7	(None, 1)	4,097	Dropout
Total params: 31,342,434 (119.56 MB) Trainable params: 31,334,370 (119.53 MB) Non-trainable params: 8,064 (31.50 KB)			

Code Snippets

Download and Extract LUNA22-ISMI Dataset

```
!wget --no-check-certificate https://zenodo.org/records/6559584/files/instructions.md
!cp /content/instructions.md /content/drive/MyDrive/Dataset/LUNA22-ISMI
!wget --no-check-certificate https://zenodo.org/records/6559584/files/LIDC-IDRI_1176.npy
!cp /content/LIDC-IDRI_1176.npy /content/drive/MyDrive/Dataset/LUNA22-ISMI
!wget --no-check-certificate https://zenodo.org/records/6559584/files/LIDC-IDRI_1176.zip
!cp /content/LIDC-IDRI_1176.zip /content/drive/MyDrive/Dataset/LUNA22-ISMI
!unzip "/content/drive/MyDrive/Dataset/LUNA22-ISMI/LIDC-IDRI_1176.zip" -d "/content/drive/MyDrive/Dataset/LUNA22-ISMI"
```

```
import nibabel as nib
import os
import numpy as np
```

```
patch_path = '/content/drive/MyDrive/Luna22/Dataset/LUNA22-ISMI/LIDC-IDRI'
patches = os.listdir(patch_path)
print(f"Total number of nodules = {len(patches)}")
print(f"List of nodules = {patches}")
```

To know the range of pixel/voxel intensity values

```
min = 0
max = 0
for i in patches:
  scan = nib.load(os.path.join(patch_path, i))
  scan = scan.get_fdata()
  if min > np.min(scan):
    min = np.min(scan)
  if max < np.max(scan):
    max = np.max(scan)
print(f'The minimum pixel values is {min} and maximum pixel values is {max}.')
```

```
The minimum pixel values is -3024.0 and maximum pixel values is 6054.0.
```

For Visualization

```
# To show the histogram of HU for a patient.
import matplotlib.pyplot as plt
def plot_hu(volume, title=""):
  plt.title(title, fontdict = font)
  plt.hist(volume.flatten(), bins=40, color='c')
  plt.xlabel("Hounsfield Units (HU)")
  plt.ylabel("Frequency")
  plt.show()
```

```
# To Visualize a single CT-Scan image slice.

def plot_a_slice(slice, title=''):
  plt.title(title, fontdict = font)
  plt.axis('on')
  plt.imshow(slice, cmap='gray')
  plt.show()
```

For Visualization

```
# To Visualize series number of CT-Scan image slices.

def plot_slices_nifti(volume, start_slice=0):#, save_path='/content'):
  volume = np.rot90(np.array(volume))
  fig, axis = plt.subplots(8, 8, figsize=(15, 15))
  slice_counter = start_slice

  for i in range(8):
    for j in range(8):
      axis[i][j].imshow(volume[:, :, slice_counter], cmap="gray")
      axis[i][j].axis("off")
      slice_counter += 1
      axis[i][j].set_title(f"Slice {slice_counter}")
  fig.suptitle('CT Scan Patch Slices', fontsize=16)
  # plt.savefig(save_path, format="png")  # You can specify the filename and format
  plt.tight_layout()
```

Reading and resizing

```
import os
import numpy as np

import nibabel as nib

import random

from scipy import ndimage

def read_scan(filepath):
    """Read and load volume"""
    # Read file
    scan = nib.load(filepath)
    # Get raw data
    scan = scan.get_fdata()
    return scan
```

```
def resize_volume(img):
    """Resize across z-axis"""
    # Set the desired depth
    desired_depth = 64
    desired_width = 64
    desired_height = 64
    # Get current depth
    current_depth = img.shape[-1]
    current_width = img.shape[0]
    current_height = img.shape[1]
    # Compute depth factor
    depth = current_depth / desired_depth
    width = current_width / desired_width
    height = current_height / desired_height
    depth_factor = 1 / depth
    width_factor = 1 / width
    height_factor = 1 / height
    # Rotate
    img = ndimage.rotate(img, 90, reshape=False)
    # Resize across z-axis
    img = ndimage.zoom(img, (width_factor, height_factor, depth_factor), order=1)
    return img
```

For splitting the data into benign and malignant

```
lidc_ann = '/content/drive/MyDrive/M_Sc_Thesis/Datasets/LUNA22-ISMI/LIDC-IDRI_1176.npy'
dataset = np.load(lidc_ann, allow_pickle=True)
print(dataset)
```

```
[{'SeriesInstanceUID': '1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860', 'Voxe
 {'SeriesInstanceUID': '1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860', 'Voxe
 {'SeriesInstanceUID': '1.3.6.1.4.1.14519.5.2.1.6279.6001.100398138793540579077826395208', 'Voxe
 ...
```

```
dataset[0]
```

```
{'SeriesInstanceUID': '1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860',
 'VoxelCoordX': 45,
 'VoxelCoordY': 211,
 'VoxelCoordZ': 77,
 'Diameter': [6.97167141, 6.97167141, 7.34878692, 5.94228451],
 'Texture': [5, 5, 5, 5],
 'Malignancy': [4, 2, 4, 2],
 'Calcification': [6, 6, 6, 6],
 'Filename': '1.3.6.1.4.1.14519.5.2.1.6279.6001.100225287222365663678666836860_45_211_77_0000.nii.gz'}
```

```
len(dataset)
```

```
1176
```

```
b_c = 0
m_c = 0
for i in dataset:
  filename = i['Filename']
  m = np.median(i['Malignancy'])
  # t = np.median(i['Texture'])
  if m < 3:
    b_c += 1
  else:
    m_c += 1
print(f"Benign total = {b_c}, Malignant total = {m_c}")

  #print(filename, malignancy, texture)
```

```
Benign total = 380, Malignant total = 796
```

```
import shutil
from shutil import copyfile
import os
luna22 = '/content/drive/MyDrive/M_Sc_Thesis/Data4Model/Luna22_2'
os.makedirs(os.path.join(luna22, 'Benign'))
os.makedirs(os.path.join(luna22, 'Malignant'))
```

```
for luna22, dirs, files in os.walk(luna22):
  for subdir in dirs:
    print(os.path.join(luna22, subdir))
```

```
/content/drive/MyDrive/M_Sc_Thesis/Data4Model/Luna22_2/Benign
/content/drive/MyDrive/M_Sc_Thesis/Data4Model/Luna22_2/Malignant
```

```
patch_path = '/content/drive/MyDrive/M_Sc_Thesis/Datasets/LUNA22-ISMI/LIDC-IDRI'
benign_path = '/content/drive/MyDrive/M_Sc_Thesis/Data4Model/Luna22_2/Benign'
malignant_path = '/content/drive/MyDrive/M_Sc_Thesis/Data4Model/Luna22_2/Malignant'
```

```
for i in dataset:
    filename = i['Filename']
    s_path = os.path.join(patch_path, filename)
    m = np.median(i['Malignancy'])
    t = np.median(i['Texture'])
    if m < 3:
        d_path = os.path.join(benign_path, filename)
    else:
        d_path = os.path.join(malignant_path, filename)
    copyfile(s_path, d_path )

print(f"Benign total = {len(os.listdir(benign_path))},
 Malignant total = {len(os.listdir(malignant_path))}")
```

```
Benign total = 380, Malignant total = 796
```

For splitting the data into train, validation, and test set (similarly for test and Val)

```
benign_scan_paths = [
    os.path.join(
        os.getcwd(),
        "/content/drive/MyDrive/MScThesis/Luna22/Backup/Dataset/Binary_class/Benign",
        x)
    for x in os.listdir(
        "/content/drive/MyDrive/MScThesis/Luna22/Backup/Dataset/Binary_class/Benign")
]
malignant_scan_paths = [
    os.path.join(
        os.getcwd(),
        "/content/drive/MyDrive/MScThesis/Luna22/Backup/Dataset/Binary_class/Malignant",
        x)
    for x in os.listdir(
        "/content/drive/MyDrive/MScThesis/Luna22/Backup/Dataset/Binary_class/Malignant")
]
print("CT scan patches with benign lung tumor: " + str(len(benign_scan_paths)))
print("CT scans patches with malignant lung tumor: "+ str(len(malignant_scan_paths)))
```

```
CT scan patches with benign lung tumor: 380
CT scans patches with malignant lung tumor: 796
```

```
# For train
# copy

for i in benign_scan_paths[:304]:
    filename = i.split('/')[10]
    d_path = os.path.join(train_dir, 'benign', filename)
    copyfile(i, d_path)

for i in malignant_scan_paths[:637]:
    filename = i.split('/')[10]
    d_path = os.path.join(train_dir, 'malignant', filename)
    copyfile(i, d_path)
print('Done Copying.')
```

```
Done Copying.
```

For Augmentation

```python
from scipy import ndimage
# 1
def flip(volume):
    # Flips the image along two randomly selected axes out of the three dimensions.
    axes = tuple(random.sample(range(3), k=2))
    augmented_image = np.flip(volume, axis=axes)
    return augmented_image
# 2
def scale(volume):
    # Scales the image by a random factor between 0.8 and 1.2 along each axis independently.
    scales = random.uniform(0.8, 1.2)
    zoomfactor = (scales, scales, scales)
    augmented_image = ndimage.zoom(volume, zoomfactor, order=1)
    return augmented_image
# 3
def translate(volume):
    # Translates the image by a random amount within a range of -10 to 10 pixels along each axis.
    shifts = tuple(random.randint(-10, 10) for _ in range(3))
    augmented_image = ndimage.shift(volume, shifts)
    return augmented_image
# 4
def addNoise(volume):
    # Adds Gaussian noise to the image with a random standard deviation between 0.05 and 0.2.
    noise_level = random.uniform(0.05, 0.2)
    augmented_image = volume + np.random.normal(0, noise_level, volume.shape)
    return augmented_image
# 5
def contrast(volume):
    # Applies shear transformation to the image with random shear factors between -0.2 and 0.2 along each axis.
    contrast_factor = random.uniform(0.9, 1.1)
    augmented_image = volume * contrast_factor
    return augmented_image
# 6
def brightness(volume):
    # Adjusts the brightness of the image by a random factor.
    # Shifts the intensity values of a 3D image by -30 to 20 factor to make it brighter or darker
    brightness_factor = random.uniform(-30, 20)
    augmented_image = volume + brightness_factor
    return augmented_image
# 7
def spline_filter(volume):
    # Simulate slight deformations in the image
    volume = ndimage.spline_filter(volume)
    return volume
```

Windowing and Normalizing

```
def windowing_and_normalize(volume):
    """Windowing the volume"""
    min = -1000
    max = 400
    volume[volume < min] = min
    volume[volume > max] = max

    """Normalize the volume"""
    volume = (volume - min) / (max - min)
    volume = volume.astype("float32")
    return volume
```

```
def process_scan(path):
    """Read and resize volume"""
    # Read scan
    volume = read_scan(path)
    # Windowing and Normalize
    volume = windowing_and_normalize(volume)
    # Resize width, height and depth
    volume = resize_volume(volume)
    return volume
```

```
# Training data

ben_train_paths = [
    os.path.join(os.getcwd(),
                 "/content/drive/MyDrive/MScThesis/Luna22/Dataformodel/train/benign", x)
    for x in sorted(os.listdir(
        "/content/drive/MyDrive/MScThesis/Luna22/Dataformodel/train/benign"))
]

# Folder "Malignant" consist of CT scans having
# patches of lung nodules that are malignant.

mal_train_paths = [
    os.path.join(os.getcwd(),
                 "/content/drive/MyDrive/MScThesis/Luna22/Dataformodel/train/malignant", x)
    for x in sorted(os.listdir(
        "/content/drive/MyDrive/MScThesis/Luna22/Dataformodel/train/malignant")[:1520])
]

print("benign in training: " + str(len(ben_train_paths)))
print("malignant in training: " + str(len(mal_train_paths)))
```

Converting and Saving as NumPy array for loading the data to the model

(Similar for test and validation

```
# Read and process the scans.
# Each scan is resized across height, width, and depth and rescaled.
mal_train_scans = np.array([process_scan(path) for path in mal_train_paths])
ben_train_scans = np.array([process_scan(path) for path in ben_train_paths])

# assign 1, for the malignant and assign 0 for benign.
mal_train_labels = np.array([1 for _ in range(len(mal_train_scans))])
ben_train_labels = np.array([0 for _ in range(len(ben_train_scans))])

x_train = np.concatenate((mal_train_scans, ben_train_scans), axis=0)
y_train = np.concatenate((mal_train_labels, ben_train_labels), axis=0)
```

```
save_data = '/content/drive/MyDrive/MScThesis/Luna22/Dataformodel/nump'
np.save(os.path.join(save_data,'x_train') , x_train)
np.save(os.path.join(save_data,'y_train') , y_train)
```

```
def train_preprocessing(volume, label):
    """Process training data by adding a channel."""
    volume = tf.expand_dims(volume, axis=3)
    return volume, label
```

```
# Define data loaders.
train_loader = tf.data.Dataset.from_tensor_slices((x_train, y_train))
validation_loader = tf.data.Dataset.from_tensor_slices((x_val, y_val))

batch_size = 16
# Augment the on the fly during training.
train_dataset = (
    train_loader.shuffle(len(x_train))
    .map(train_preprocessing)
    .batch(batch_size)
    .prefetch(2)
)
# Only rescale.
validation_dataset = (
    validation_loader.shuffle(len(x_val))
    .map(validation_preprocessing)
    .batch(batch_size)
    .prefetch(2)
)
```