Arab American University
Faculty of Graduate Studies

Lung Cancer Detection System Using Medical Image Processing, Machine Learning, and Deep Learning Approaches

By
Yara Zaher Mohammad Zayed
Supervisor
Prof. Dr. Mohammed Awad

This Thesis Was Submitted in Partial Fulfillment of the Requirements for the master's degree in data science and business analytics.

Jan 2024

Thesis Approval

Lung Cancer Detection System Using Medical Image Processing, Machine Learning, and Deep Learning Approaches

By

Yara Zaher Mohammad Zayed

This Thesis was Defended Successfully on **16/01/2024** and Approved By:

Committee Members	Signature
1. Supervisor: **Prof. Dr. Mohammed Awad**	
2. Internal Examiner: **Dr. Mahmoud Obaid**	
3. External Examiner: **Dr. Yousef Daraghmah**	

Declaration

I, Yara Zaher Mohammad Zayed, declare that this thesis/dissertation is entirely my own work and that all sources used are properly acknowledged.

Student Name	Yara Zaher Mohammad Zayed
University ID	202112254
Date	22/01/2024
Signature:	

Dedication

To the land of martyrs and heroes, to my homeland and its people, to Palestine, and the steadfast Palestinian people, I dedicate this thesis. May it serve as a small contribution to the aspirations and resilience of our beloved nation. Dedicated to my family, whose unwavering love and support have been my constant source of inspiration. Your belief in me has fueled this journey, and I am profoundly grateful for your encouragement and sacrifices. I also dedicate this thesis to my mentors whose contributions, knowledge, and wisdom have shaped my academic path. Thank you for being part of my educational path.

Acknowledgments

I would like to take this moment to extend my sincere appreciation to Prof. Mohammed Awad for his unwavering support, invaluable guidance, and invaluable insights. His assistance has been instrumental in helping me navigate the challenges I encountered throughout my thesis work.

Abstract

Early detection of lung cancer stands as a pivotal turning point in the realm of medical care, where the timely identification of this condition empowers healthcare professionals to prescribe the most effective treatments, ultimately leading to a reduction in mortality rates and the preservation of precious human lives. Countless dedicated researchers have explored this subject, employing a multifaceted approach to diagnosing lung cancer.

In response to the formidable challenge of selecting optimal methodologies to ensure consistently high performance, this thesis introduces two distinctive models, one based on machine learning (ML) and the other on deep learning (DL). Both models are designed to facilitate the diagnosis of lung cancer, employing clinical data and medical images.

In the domain of ML, a multitude of machine learning models, including eXtreme Gradient Boosting (XGB), Light Gradient Boosting (LGB), Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Gradient Boosting Machine (GBM), and Adaptive Boosting (AdaBoost), have been used. This model culminated in a remarkable achievement, with the XGB and Adaboost models achieving an accuracy of 99.07%.

Conversely, the DL was trained on CT scan images categorized under three distinctive labels: Normal, Benign, and Malignant. These images underwent meticulous preprocessing and training across eight deep learning algorithms, namely Convolutional Neural Network (CNN), MobileNet, Xception, DenseNet121, Visual Geometry Group 16 (VGG16), Visual Geometry Group 19 (VGG19), ResNet50, and EfficientNetB0. The best performance, which was achieved by the custom CNN model, boasts an extraordinary classification rate of 99.70% and perfect precision, recall, and f1-score.

Table of Contents

List of Figures

List of Tables

Table of Abbreviations

GLOBOCAN	Global Cancer Observatory
AIDS	Acquired Immunodeficiency Syndrome
LC	Lung Cancer
CT	Computed Tomography
DICOM	Digital Imaging and Communications in Medicine
AIDS	Acquired Immunodeficiency Syndrome
TB	Tuberculosis
HIV	Human Immunodeficiency Virus
DL	Deep Learning
ML	Machine Learning
ANNs	Artificial Neural Networks
NSCLC	Non-Small Cells Lung Cancer
SCLC	Small-Cell Lung Carcinoma
NET	Neuroendocrine Tumors
LDA	Linear Discriminate Analysis
ODNN	Optimum Deep Neural Network
PET	Positron Emission Tomography
MRI	Magnetic Resonance Imaging
UTE	Ultrashort Echo Time
MIT	Magnetic Induction Tomography
QALY	Quality-Adjusted Life Year
CAD/CADe	Computer-Aided Detection and Diagnosis
AI	Artificial Intelligence
DNN	Deep Neural Network
DCNN	Deep Convolutional Neural Network
FSL	Few-Shot Learning
GLCM	Gray Level Co Occurrence Matrix
LCC	Large Cell Carcinoma
SCC	Squamous Cell Carcinoma
DWT	Discrete Wavelet Transformation
MIP	Maximum Projection Intensity
LP	Laplacian Pyramid Decomposition
ASR	Adaptive Sparse Representation
DSC	Dice Similarity Coefficient
R-CNN	Region-Based Convolution Neural Network

RPN	Region Proposal Network
RCNN	Residual Convolutional Neural Network
TPs	True Positives
FNs	False Negatives
FPs	False Positives
TNs	True Negatives
3DDCNN	Three-Dimensional-Deep Convolutional Neural Network
mRPN	Multi-Region Proposal Network
RoI	Region Of Interest
CC	Cloud Computing
HT	Historical Therapy
mRFCN	Multidimensional Region-Based Fully Convolutional Network
DSS	Decision Support System
TSCNN	Two-Stage Convolutional Neural Networks
GCPSO	Guaranteed Convergence Particle Swarm Optimization
BoVW	Bag Of Visual Words
CRNN	Convolutional Recurrent Neural Network
CLAHE	Contrast Limited Adaptive Histogram Equalization
BGR	Blue-Green-Red
SMOTE	Synthetic Minority Oversampling Technique
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic Curve
TNR	True Negative Rate
FPR	False Positive Rate
CNN	Convolutional Neural Network
VGG16	Visual Geometry Group 16
EfficientNetB01	EfficientNet B0
VGG19	Visual Geometry Group 19
ResNet50	Residual Network 50
DenseNet121	Dense Convolutional Network 121
Xception	Extreme Inception
DT	Decision Tree
RF	Random Forest
XGB	Extreme Gradient Boost
SVM	Support Vector Machine
Adaboost	Adaptive Boosting
GBM	Gradient Boosting Model

Chapter 1
Introduction

1.1 Introduction

The worldwide prevalence of cancer is steadily increasing, posing a significant global public health challenge. The condition is defined by an excessive growth of cells and tissues. Because of this, the occurrence of malignant or tumor growth is observed [1]. Cancer is recognized as the primary cause of mortality on a global scale, resulting in approximately 15 million new cases and 8.2 million deaths annually [2] [3]. Furthermore, it is projected that a total of 1,898,160 instances of cancer will be identified in the United States (US), leading to 608,570 deaths [4]. Additionality, taking Africa as an example there were 1,109,209 new cases and approximately 711,429 fatalities in 2020. Overall, the cancer mortality rates exceeded the death rates of acquired immunodeficiency syndrome (AIDS), Tuberculosis (TB), and malaria together [5]. One of the most common cancers is Lung cancer, which primary cause of death for individuals of both genders globally. In 2022, the global incidence of lung cancer reached around 2.2 million newly diagnosed cases, leading to an estimated 1.8 million fatalities [6]. Lung cancer has many prominent indications and symptoms including hemoptysis (coughing up blood), weight loss, and fatigue. Additionally, this disease has a lot of risk factors such as smoking, alcohol consumption, air quality, and diet [7].

Lung cancer incidence has increased rapidly in emerging economies, particularly in sub-Saharan Africa, where the human immunodeficiency virus (HIV) has also had a negative impact on other malignancies such as cancer as it is present in colorectal cancer (65%), breast cancer (90%), and prostate cancer (99%) etc. [8]. The 5-year overall survival is low (18%) compared with all types of cancers [1]. However, it is imperative that the medical, biological, and scientific fields prioritize their efforts towards addressing the complexities of lung cancer. This entails the development of innovative strategies aimed at promoting timely detection, facilitating informed medical

interventions, and assessing treatment outcomes to enhance the overall quality of healthcare. The utilization of the 15 molecular profile of tumor tissues facilitates the detection of driver mutations and the advancement of tailored therapeutic interventions for certain genotypes. Traditional chemotherapy kills all cells, regardless of whether they are normal or cancerous. Targeted therapy, on the other hand, targets specific regions of the body, engaging with cancer-driver genes to diminish malignant transformation [9]. Due to late or erroneous diagnoses, a considerable number of cases that are treatable in industrialized countries are diagnosed after they reach untreatable stages in third-world countries, according to the Health Organization. This has prompted scientists to evaluate existing procedures and propose novel ways of classifying and detecting lung cancer and its subtypes to improve early diagnosis and treatment options.

There is a massive collection of CT scan image data and other medical data related to lung cancer, which could aid in the detection of the disease. ML and DL can make use of this data to improve cancer diagnosis and detection as soon as possible and provide the best course of action. Machine learning algorithms fall into two main categories: supervised and unsupervised. Supervised algorithms utilize labeled input and output data, such as images labeled as malignant or benign scans. On the other hand, unsupervised algorithms aim to uncover previously unknown relationships and patterns in data without explicitly classifying them. While Deep learning is a subfield of machine learning that derives its principles from the complex structure of the human nervous system. With DL machines can process high dimensionality data like images and videos. To improve recognition skills, DL employs a combination of Artificial Neural Networks (ANNs) [10]. The primary objective of employing both machine and deep learning is to acquire the ability to classify input samples, like medical MRI scans or pathological images, into predefined categories. These categories may include specific sub-type diagnoses, such as tumors, or broader diagnostic groups, like distinguishing between malignant and benign cases [11].

1.1.1 Types of Lungs Cancer

Lung cancer is histologically categorized into three classes of non-small cell lung cancer (NSCLC), small-cell lung carcinoma (SCLC), and neuroendocrine tumors (NET). NSCLC has the highest percentage of 85%-90% of all lung cancer cases and has main subtypes including squamous-cell carcinoma, adenocarcinoma, and large-cell carcinoma. Squamous cell and adenocarcinoma, both collectively account for 80% of the total cases. The Large-cell carcinomas are relatively rare, accounting for approximately 5% of all lung cancer cases, and they do not display squamous or glandular characteristics.

Squamous cell carcinomas are found to be strongly correlated with the habit of smoking and tend to develop as large tumors in the central region of the lung. On the contrary, adenocarcinomas have emerged as the predominant form of lung cancer among individuals who do not engage in smoking, while concurrently experiencing a rise in prevalence among smokers in recent times.

Typically occurs at the periphery of the lung. They can be further categorized into acinar, papillary, bronchiole-alveolar carcinoma (BAC), and solid adenocarcinoma with mucin production. Nevertheless, many cases exhibit mixed histologic patterns. SCLC stands out as the most aggressive form of lung tumor due to its high metastatic potential in comparison to other lung cancer types. Its strong association with active smoking is evident, with over 95% of SCLC patients being either current or former smokers [5] [12].

1.1.2 Lung Imaging Diagnosis Techniques

Radiologists use medical imaging technologies to diagnose lung cancer. Among the various medical imaging modalities, Computed Tomography (CT) stands out for its numerous advantages in terms of size, location, characterization, and lesion growth. These attributes make CT essential for identifying lung cancer and nodule information. The implementation of 4D CT further

enhances precision in radiation targeting, exerting a significant effect on the care of individuals with lung cancer [13] [14].

Positron Emission Tomography (PET) exhibits significantly higher sensitivity and specificity in identifying lung nodules compared to CT, particularly in cases involving reactive or granulomatous nodal disease [15]. PET demonstrates a strong correlation with prolonged progression times and overall survival rates. Specifically, 18F-FDG Solitary pulmonary nodules are effectively diagnosed using PET [16] .The utilization of 18F-FDG PET is crucial as an in-patient selection tool for advanced NSCLC patients undergoing intensive radiation. PET-assisted radiation contributes to enhanced accuracy [17], and it is applied in the treatment of approximately 32% of patients with stage IIIA lung cancer [17].

Magnetic resonance imaging (MRI) is a non-ionizing radiation lung imaging technology; however, it gives insufficient information at a high cost and with time constraints. It missed approximately 10% of tiny lung nodules (4-8 mm in diameter) [18]. Ultrashort echo time (UTE) MRI can improve the signal intensity and decrease lung susceptibility artifacts. Small lung nodules (4-8 mm) can be detected with MRI and UTE [19]. MRI detects lung nodules more frequently than LDCT. MRI with varied pulse sequences also enhanced the sensitivity of lung nodule detection. T1-weighted and T2-weighted MRIs were used to detect tiny lung nodules [20], [21]. In comparison to 3T 1.5 MRI, 1.5T MRI is substantially better at detecting ground glass opacities. Ground glass opacities were successfully detected in 75% of lung fibrosis patients who underwent 1.5T MRI with SSFP sequences [22]. In immunocompromised subjects, MRI with T2-weighted rapid spin echo gives comparable or even higher performance for ground glass infiltration detection [23]. Several research groups have recently investigated the viability of using Magnetic Induction Tomography (MIT) to identify lung disease [24]. Nevertheless, the widespread utilization of MIT technology as a commercial imaging tool is still a distant prospect, mostly due to a lack of measurement

devices, the high cost of computational electromagnetic models, constraints on image resolution, and various other challenges.

1.1.3 Lung Cancer Diagnosis Challenges

Numerous clinical trials have substantiated three challenges associated with current screening programs: elevated over-diagnosis rates, significant costs, and a rise in radiation exposure. The NLST study [25], For instance, it exhibited a 96.4% rate of false positives in the context of positive screening outcomes, resulting in expenditures amounting to $52,000 and $81,000 per individual for an extra year of life and one quality-adjusted life year (QALY), Correspondingly [25]. Furthermore, radiation caused 1–3 deaths from lung cancer for every 10,000 participants [25]. All these problems make it necessary to introduce solutions such as computer-aided detection systems [26], [27]. ML, DL, and statistical methods can increase the capability of CAD/CADe systems to accurately detect and diagnose lung cancer. Over the past twenty years, this topic has garnered a considerable interest of attention from researchers working in the field of medical image processing. In contrast to human reading performed by thoracic radiologists, the utilization of computer-aided detection/diagnosis (CAD/CADe) systems as an adjunct to radiologists during the reading process has promise for enhancing the positive predictive value and reducing the false positive rate in the context of lung cancer screening for tiny nodules [28]. This is because CAD/CADe systems are capable of potentially enhancing the positive predictive value and reducing the false positive rate. It has also been demonstrated that CAD/CADe systems make screening more efficient and cost-effective**.**

1.1.4 Lung Cancer Diagnosis Using Machine learning and Deep learning.

Artificial intelligence (AI) has fundamentally altered how people view computers since it has taken on and resolved several issues that were first thought to be extremely challenging for people to

address. However, dealing with tasks that are regarded as simple by humans but extremely difficult to formalize poses the actual problem for AI. To put it another way, issues that we commonly resolve intuitively, such as identifying faces, scribbled numbers, spoken sentences, etc. This problem argues that systems should learn from raw data. Machine Learning allows computers to use real-world knowledge to predict and make decisions. However, ML faces another challenge because model success depends on data quality. Thus, a doctor is still needed to be interfaced with the system and information.

While DL is now a technical trend that has generated a wide range of research due to its excellent outcomes in a variety of applications. It can be categorized as a sub-field of Machine ML approaches that employ layers of neural networks to learn complex concepts by building them out of smaller ones [29].

Deep Learning models are exemplified by artificial neural networks. Mathematical functions map inputs to outputs. These mappings approximate real functions. Layers in this model fulfill several responsibilities. The first layers use fundamental algorithms, largely from Computer Vision (CV), and feed the results into the next layers. Based on simple notions learned in earlier levels, the system can build complicated algorithms like object detectors. CNNs are state-of-the-art in image recognition. They use convolution, which generates an output by sliding a window along the input image. Deep Convolutional Neural Network (DCNN) models use several sliding windows with variable widths and values. DCNNs are now widely employed in industry jobs thanks to recent hardware and software advances that overcame their high processing costs. DL, which outperforms ML models, is seeing rapid growth and has numerous applications, particularly in the domain of medical imaging. It's also used to build CADs. Due to patient confidentiality issues, these systems often have data limitations, especially in the medical profession.

DL-based CAD systems may accurately diagnose lung illness in medical imaging [30]. Labeled and unlabeled data improve decision boundary accuracy, but training requires lots of labeled data, which increases cost, effort, and difficulty [31] [32], [33] [34]. A variety of supervised DL algorithms have been utilized in the analysis of CT images with the purpose of detecting anomalies and determining their anatomical localization. However, these methodologies do have several drawbacks. These include the need for a significant amount of labeled data, the assumption that network weights remain fixed after training, and the inability to evolve or adapt post-training.

1.2 Objectives

The primary objective is to improve the accuracy of lung cancer diagnosis using ML/DL techniques applied to two datasets of CT scans and clinical information. The accuracy has been measured the different evaluation metrics such as sensitivity, specificity, precision, and F1 score. In other words, the fundamental purpose of the study is to investigate the proper process for analyzing data, selecting features, and developing a more precise approach for assessing lung cancer. Early identification and correct diagnosis utilizing AI techniques contribute to the treatment plan as well as the avoidance of many serious diseases.

Specific objectives of the study

1 Applying data preprocessing and exploration techniques on the clinical and demographic datasets.

2 Applying a set of ML algorithms for lung cancer classification on a clinical and demographic dataset and choosing the most appropriate ML technique for the classification of lung cancer

3 Choosing the most effective variables that affect the dataset classification by using the Feature importance check.

4 Applying various medical image processing techniques including segmentation.

5 Applying Deep Learning (DL) models on CT Lung Images to classify the types of Lung cancer.

6 Performing a performance metrics comparison between the various ML / DL algorithms to find the most appropriate technique to implement a lung cancer CAD.

1.3 Contribution

This thesis contributes the lung cancer detection field by the following:

- Adopted a comprehensive two-model approach integrating machine learning (ML) and deep learning (DL) techniques for improved lung cancer detection.
- In ML Phase, it prioritized data efficiency, ensuring data quality and accuracy. It also addressed data imbalance issue using oversampling technique and optimized models through hyperparameter tuning. Beside identifying the crucial diagnostic features to enhance diagnostic accuracy.
- In DL Phase, it emphasized the refinement of medical imaging preprocessing and image segmentation techniques. As it overcame the limitations in existing methods related to inadequate preoperative methods and ineffective site of interest selection.
- Applied deep learning algorithms for feature extraction from segmented images, efficiently encoding them at a simple spatial representation level. Such that it leveraged the power of deep learning to detect complex patterns in data and enhance the linear separation process.
- Overall, these contributions aim to empower physicians, advance lung cancer detection, and highlight the significance of refining medical imaging techniques in the field of deep learning.

1.4 Overview

In this section of the chapter introduction, a synopsis will be presented for the upcoming chapters of this thesis, which aim to achieve the objectives and goals of the research. The thesis follows the following structure:

- Chapter 2 describes the global datasets used in the study, including lung cancer CT scan images, and the other demographic / risk factor dataset. It surveys the literature on the research topic and associated methods of tumor diagnosis in ML and DL.

- Chapter 3 sheds light on the methods used in this study, including data collection, and preprocessing techniques to improve data quality. It includes the use of the machine learning models used in the classification of the demographic/clinical dataset including DT, RF, SVM, XGB, LGB, and Adaboost. For medical images classification deep learning models were used such as CNN, VGG16, EfficientNetB0, MobileNet, VGG19, ResNet50, Xception, and DenseNet121, as well as custom CNN models, are presented. In addition to performance introducing a variety of measures to test deep learning models.
- Chapter 4 analyzes the research in detail and presents the results using different models for each data set. A comparative analysis is provided to identify the best model in each section. Besides making a comparison with a published paper as a benchmark.
- Chapter 5 concludes the topic and provides suggestions for further development.

Chapter 2
Background

2.1 Background

Lung cancer ranks as the second most prevalent kind of cancer within the United States, projected to yield approximately 238,340 novel instances by the year 2023. This encompasses 117,550 cases among males and 120,790 cases among females. The impact of this condition is primarily observed in the elderly population, with most diagnoses occurring in those aged 65 or older, and the average age of diagnosis is around 70 [35].

Significantly, lung cancer has the position of being the primary cause of cancer-related fatalities in the US, constituting nearly one-fifth of all cancer-related deaths. The 5-year survival rate, on average, stands at a mere 17%. The low survival rate of lung cancer can be attributed, in part, to the fact that merely 15% of cases are identified during the early stages, characterized by the absence of apparent cancer symptoms. On the other hand, if lung cancer is found at an early stage, the chance of survival increases by as much as 54%. Lung cancer early detection is a major problem, and research has focused on this where Computer-aided diagnosis (CAD) is an important topic of study in medical diagnosis [35].

The primary idea behind CAD is to use the numerical results generated by algorithmic computer software as a "second opinion" to aid doctors in illness detection interpretation tasks. CAD helps doctors by increasing diagnosis accuracy and consistency and reducing the images and symptoms interpretation time [36]. Several classification models, including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), Gradient Boosting (GBM), Light Gradient Boosting (LGB), Extreme Gradient Boost (XGB) and Adaptive Boosting (AdaBoost) were employed in this thesis. These models were applied to the global clinical/demographic/risk factors dataset, and the resulting outcomes were compared. In addition, to applying many deep learning models including

Convolutional Neural Network (CNN), Visual Geometry Group-16 (VGG16), EfficientNet B01, MobileNet, Visual Geometry Group-19 (VGG19), Residual Network 50 (ResNet50), Dense Convolutional Network 121(DenseNet121) and Extreme Inception (Xception) to categorize images of the CT Lung Images and split them into healthy, benign, and malignant inflammation.

2.2 Dataset Description

This section provides an overview of the datasets utilized in this study and their significance in addressing the research objectives. Two distinct datasets were utilized: CT images, and clinical, demographic and risk factor dataset. Each dataset was employed in a separate experiment, allowing for focused analysis of specific aspects related to lung cancer classification. By utilizing these two datasets together, this study strives to enhance the accuracy and reliability of lung cancer classification models.

2.2.1 Clinical, Demographic and Risk Factor Dataset

The “Lung Cancer” dataset is a public dataset that was published in Kaggle [37]. This dataset consists of 15 independent features which include a set of demographic information, risk factors, and symptoms that are related to lung cancer and one target feature which aims to diagnose lung cancer for 284 patients. Table 2.2.1 below shows the attributes in the used dataset.

The target variable in the dataset is "Lung Cancer" which indicates whether the patient was diagnosed with lung cancer or not. The dataset contains 309 records for 39 patients that did not have lung cancer (12.60% of the dataset) and 270 patients that were diagnosed with lung cancer, (87.40% of the dataset). As shown in the Figure 2.2.1 below.

Table 2.2.1 : Kaggle Dataset Description.

Feature Name	Description	Data Type
Gender	M-Male, F-Female	Categorical
Age	-	Numerical
Smoking	yes-2, no-1	Numerical
Yellow Fingers	yes-2, no-1	Numerical
Anxiety	yes-2, no-1	Numerical

Peer Pressure	yes-2, no-1	Numerical
Chronic Disease	yes-2, no-1	Numerical
Fatigue	yes-2, no-1	Numerical
Allergy	yes-2, no-1	Numerical
Alcohol Consuming	yes-2, no-1	Numerical
Wheezing	yes-2, no-1	Numerical
Coughing	yes-2, no-1	Numerical
Shortness of Breath	yes-2, no-1	Numerical
Swallowing Difficulty	yes-2, no-1	Numerical
Chest Pain	yes-2, no-1	Numerical
Lung Cancer	Yes / No	Categorical

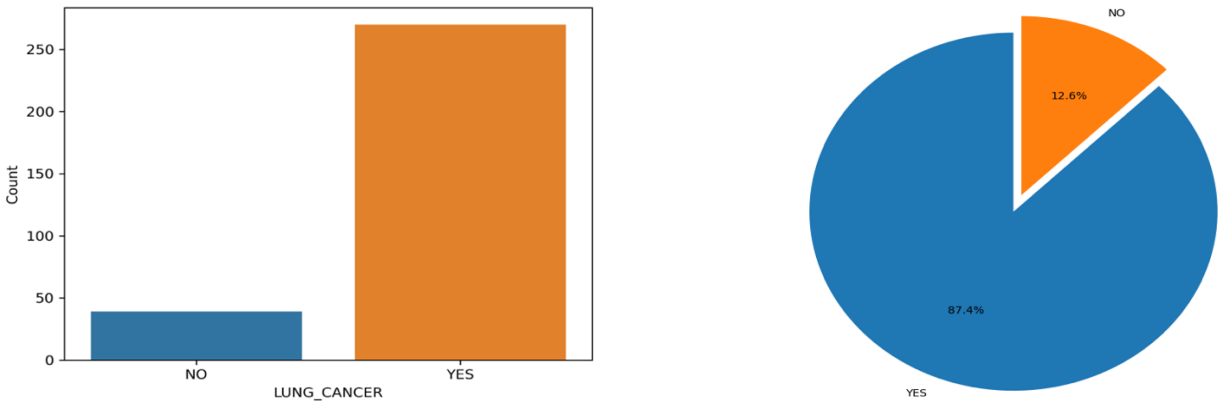


Figure 2.2.1: CT Images Dataset Distribution Per Class.

Further exploration was made to explore the distribution of lung cancer per the other features Figure 2.2.2). The output shows that anxiety does not have a clear relationship, while those who has a history of the alcohol consumption has higher percentage of being infected by the lung cancer. And the same applies on the smoking. As well the plots below show the distribution per gender and age, which confirm that males have a higher number of being infected. With regards the age, the age of the infected people within the dataset are ranging from 50-80s.

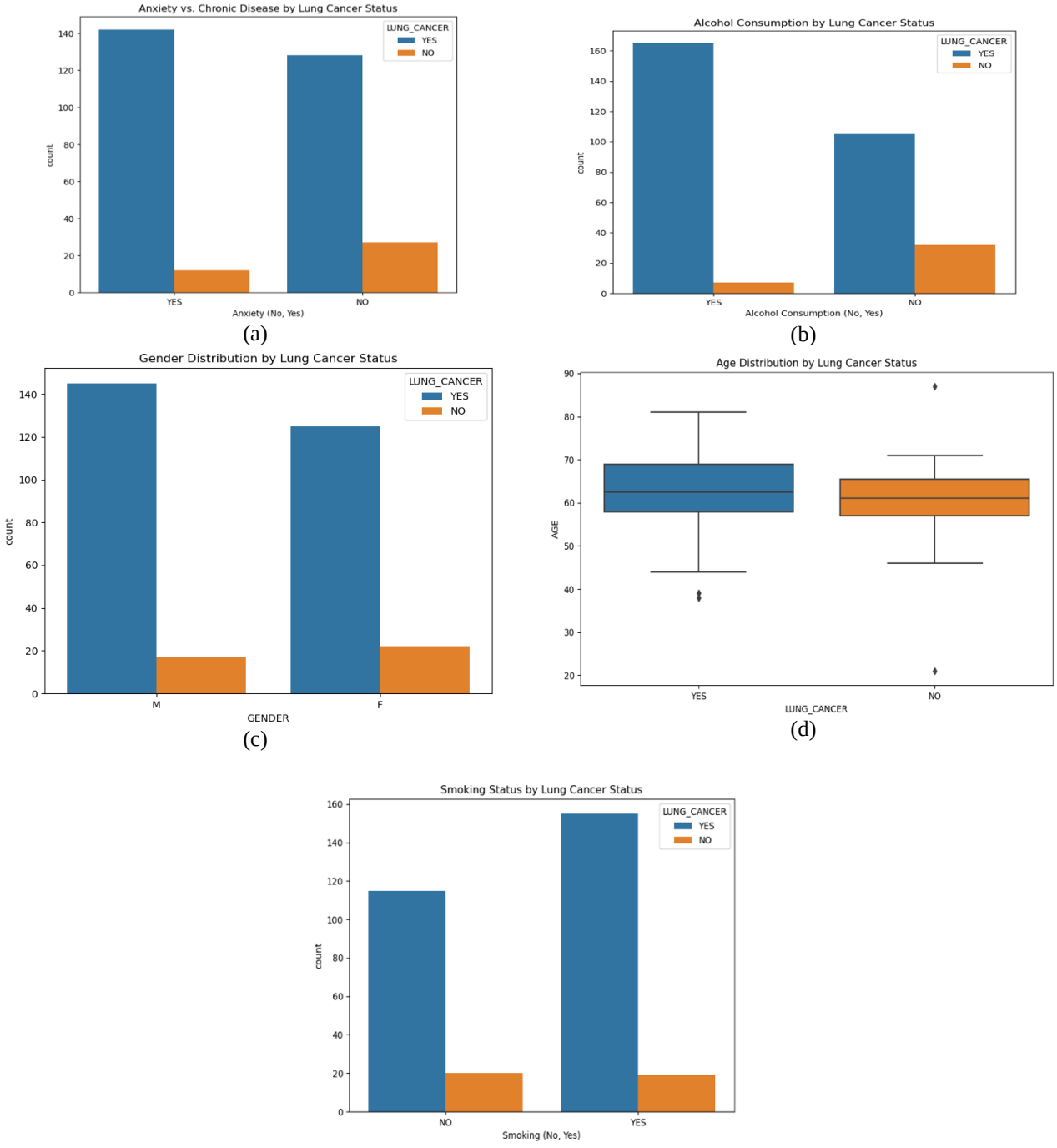


Figure 2.2.2 : (a) Anxiety vs lung cancer (b) alcohol consumption vs lung cancer (c) gender vs lung cancer (d) age vs lung cancer (e) smoking vs lung cancer

2.2.2 CT Images Dataset

The lung cancer dataset employed in this study was sourced from the Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases (IQ-OTH/NCCD) [38] over a three-month period in the fall of 2019. This dataset includes CT scans from patients diagnosed with lung cancer at various

stages and a control group of healthy individuals. Proficient professionals, including oncologists and radiologists from the hospitals, annotated and marked the slides provided by IQ-OTH/NCCD. The dataset comprises 1190 images, representing CT scan slices from 110 cases categorized into normal, benign, and malignant classes. Specifically, 40 cases were identified as malignant, 15 as benign, and 55 as normal. The original collection of CT scans was in DICOM format and acquired using Siemens' SOMATOM scanner, with a CT protocol involving 120 kV, a 1 mm slice thickness, and a window width ranging from 350 to 1200 HU, along with a window center from 50 to 600. All scans were taken during full inspiration with maintained breath hold. The study obtained approval from the institutional review board of participating medical centers, and written consent was waived.

Each CT scan consists of multiple slices, typically ranging from 80 to 200 slices, depicting various angles and views of the human chest. The 110 cases include diverse demographics, encompassing variations in gender, age, educational background, area of residence, and occupation. Some individuals work in the Iraqi ministries of Transport and Oil, while others are farmers and gainers. Most of the participants originate from the central region of Iraq. Figures 2.2.3, 2.2.4 and 2.2.5 present samples from the set of medical CT images for the three classes: Normal, Malignant, and Benign, followed by a pie chart in Figure 2.2.6 illustrating the distribution of these three classes.

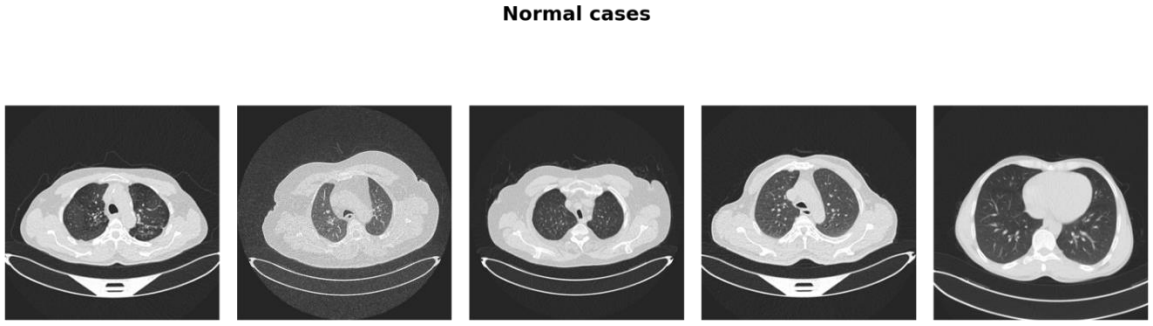


Figure 2.2.3: Sample of CT Images for Normal Cases

Figure 2.2.4: Sample of CT Images for Malignant Cases of Lung Cancer

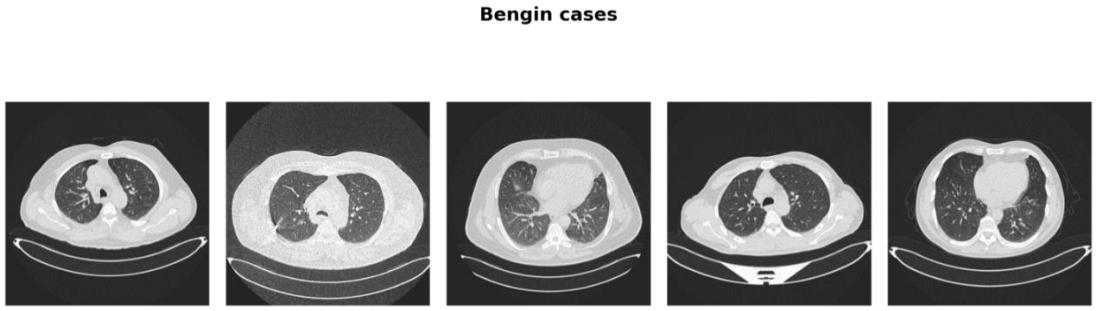


Figure 2.2.5: Sample of CT Images for Non-cancerous (Benign) Lung Conditions

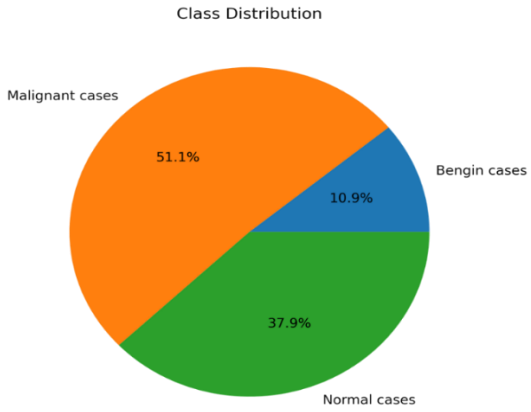


Figure 2.2.6: Dataset Distribution Per Class (Normal, Benign, Malignant)

2.3 Related Works

To keep track of more accurate and effective methods for lung cancer diagnosis, machine learning, and deep learning fields are emerging as powerful tools in the fight against this devastating disease. In this section, the literature review is analyzed into two main sections: one devoted to machine learning methods used to diagnosis lung cancer based on clinical, demographic, and symptom data, and the other is the deep learning applied to the medical images. This series of publications is a

comprehensive review of how advanced computational techniques can be used to improve lung cancer early detection, diagnosis, and prognosis using machine learning and deep learning by delving into the challenges of depth.

2.3.1 Machine Learning-Related Works.

The researchers have implemented a model to detect lung cancer based on a group of factors rather than an image that includes symptoms, risk factors, clinical and demographic attributes. In [39] the model was trained and evaluated using two data set, the first data was labeled into 3 categories including malignant tumor, benign tumor, and healthy person with no tumor. The second dataset classified into low, normal, and high possibility of having lung cancer. The two datasets were fed separately into DT, LR, SVM, and NB the outcomes shows that the best accuracy on the first dataset was for LR at 96.9% and the second was 99.2% for the SVM. Another system proposed in [40] detects the lung cancer-based onset of symptoms using ANN. Such as yellow fingers, peer pressure, chronic disease, chest pain, and shortness of breath. The system demonstrated an accuracy of 96.76%. Similarly, the researchers in [41] have used a set of medical records features to detect Lung cancer including the same features. Using a set of ensemble methods such as cross-validation tree, random forest, and random tree. Where it achieved an accuracy of 47% and 69%. While the system in [42] achieved 99.1% accuracy through RF using the same dataset. Another set of factors were utilized in [43] to detect lung cancer. These factors include a familial predisposition of lung cancer, smoking status represented by the number of years and number of packs per year, and alcohol. This data was used to train many algorithms including NB, LR, RF, and SVM. Where the LR and RF achieve a perfect accuracy of 100% which may be considered overfitting. Noting that this study basic goal is to study the relation between lung cancer and smoking rather than symptoms and risk factors.

2.3.2 Deep Learning Related Works

Numerous research work has been done to detect Lung cancer based on medical images in its early stages. This kind of studies and systems primarily consist of primary stages, the image preprocessing stage, using various techniques such as resizing, normalization, standardization, augmentation, and Filtering. Segmentation of preprocessed images is essential to identify potential nodules. After segmentation, features are extracted from the segmented images for input into the subsequent phase of the DL model. This phase accommodates both traditional machine learning models and deep learning models. Notably, researchers have increasingly investigated the application of deep learning algorithms for lung cancer detection. The efficacy of deep learning approaches lies in their ability to autonomously extract valuable features from images and proficiently classify instances of lung cancer. These systems were implemented to handle different tasks that would help on Lung cancer detection these tasks will be illustrated on the following sections.

- **Classification of Lung Images: Normal vs Abnormal**

In [44] the authors used Computed Tomography (CT) to diagnose lung cancer. The proposed methodology starts with applying image preprocessing techniques including image smoothing to remove the impulse noise without causing distortion on the edges using median filtering, followed by image enhancement using contrast adjustment by mapping the original pixel values to new values in such a way that the lowest and highest 1% of the input image data get saturated at the low and high intensity ends, respectively. The next phase was applying Image segmentation for the Lung region of interest. First, preprocessed grayscale photo was transformed to binary. Morphological opening with disk structuring element removed unnecessary elements from the binary image. The opening image was supplemented and bordered. Filling lung holes and gaps produced lung masks. Finally, exclusive OR surgery separated the tumor location from lung mask

output and clean border output. Once the images were ready the needed features were extracted including the area of the region, Perimeter, and Eccentricity. Where the area is the Cancerous lung nodule's total pixels, the Perimeter is the lung tumor's perimeter pixels, and Eccentricity or the irregularity index, circularity, or roundness. The SVM classifier was utilized to categorize the images into two groups, namely normal and up normal, based on the features and labels associated with each image.

In [45] they devised an (Adaptive Hierarchical Heuristic Mathematical Model) for the DL technique. The DL technique planned by the HT (historical therapy) method was analyzed scheme during the progress of NSCLC. For this, automated radiation protocols were constructed to optimize the local tumor regulation. In addition, the devised technique was executed in different phases such as to attain the image, preprocess, and segmentation of the image, extract the attributes, and detect the lung cancer using DNN. The lung CT image was segmented to extract the expressive features of a segmented image. The simulation outcomes demonstrated that the accuracy of the devised technique to decide whether the patient have cancer or not was calculated at 96.67 % for detecting the occurrence of lung cancer.

The implementation of the system proposed in [46] starts with applying a geometric mean filter as a preprocessing step to improve the CT image quality. Afterward, the images were segmented using the K-Mean algorithm to be used later in the feature extraction (LDA) to be fed finally to various DL / ML models including ANN, KNN, and RF for classification purposes. Where the ANN outperforms the other models.

- **Malignancy Assessment in Lung Cancer Detection**

The study [47], devise a lung cancer detection system utilizing computed tomography (CT) scans. This system comprises four key stages. First, there's the pre-processing stage, where CT-scan images undergo enhancement to improve their quality through grayscale conversion and

thresholding. Following that is the segmentation phase, where the system identifies and separates the desired cancer objects from the background using contour-based methods. Subsequently, the system extracts essential features, including area, contrast, energy, entropy, and homogeneity, utilizing the Gray Level Co-Occurrence Matrix (GLCM). Finally, it classifies lung cancer into benign and malignant categories using Support Vector Machines (SVM). During system testing, the accuracy level in Assessing if a lung cancer diagnosis indicates benign or malignant characteristics reached 83.33%.

In [48] a CAD system was proposed to detect lung nodules and classify the malignancy level of lung cancer based thoracic CT images in DICOM format labeled based on nodule size into the nodule and non-nodule as well classified the severity level into unknown, Benign, malignant primary (originate from the lung itself) and malignant metastatic (originate from other organs). The images were initially preprocessed using image preprocessing techniques like thresholding, morphology closing, and morphology opening. The system used two architectures the first is a 3D multipath VGG-like architecture and for the segmentation U-Net was used. The output of both approaches was then combined and gave an accuracy of 95.66%. The 3DCNN was adopted in this work as 2D images may overlap together causing a high false positive rate while the 3D object gives a clearer view of the objects.

In [49] the researchers suggested a lung cancer diagnosis system for image segmentation based on 3D-input CT scan data with two main parts the image segmentation using a transformers-based segmentation network called UNETR network, and the classification to classify the segmentation output into either benign or malignant using self-supervised network. The system was able to achieve a segmentation accuracy of 97.83%, and 98.77% as classification accuracy.

A. Masood, et.al [50] recommended an enhanced multidimensional region-based fully convolutional network (mRFCN) based automated decision support system (DSS) to detect and

classify the lung nodule into malignant and bengin. This algorithm adopted an image classifier backbone to extract the attributes and a new multilayer fusion region proposal network (mLRPN) was suggested. A median intensity projection was considered for deriving 3D (three-dimensional) information from CT scans. The suggested algorithm was implemented based on a deconvolutional layer for automatically selecting the ROI. The outcomes of trials proved that the proposed algorithm performed well to detect lung cancer and its sensitivity was counted at 98.1% and accuracy was 97.91% for classifying lung cancer.

CT images in DICOM format were used in [51]. The CT images were initially processed using The Gabor filter for image enhancement and modified color-based histogram equalization for contrast adjustment. Then, the image was segmented by using the Guaranteed Convergence Particle Swarm Optimization (GCPSO) algorithm. Later the features were extracted from the segmented images including the ABC parameter which describes the border structure, lesions diameter, and asymmetry, the FOS parameter, which is a set of statistical measures like entropy, standard deviation, etc., and GLCM features which estimates pixel pair frequency and texture characterization time including energy, correlation, contrast. The Bag of Visual Words (BoVW) and a Convolutional Recurrent Neural Network (CRNN) were applied for image classification into malignant and benign and feature extraction processes. In context of results, an average precision of 96.5% and, an accuracy of 99.35% were achieved.

- **Lung Cancer Subtypes Classification**

[52] Presented an automated, intelligent system to identify NSCLC and classify its subtypes Adenocarcinoma, LCC, and SCC using two DL algorithms. This was done by utilizing both CT and X-ray images where a set of features are extracted from them like area, diameter, texture, and radius. The process begins by reading the image and performing preprocessing, which involves removing any noise, resizing the input to align with the input size for VGG-19 and LSTMs,

applying filters to enhance the pixels, and transforming the image into grayscale. Next, in the deep learning phase, a zero matrix is created, and a filtration process is applied using Gabor Filter and Discrete wavelet transformation (DWT) to calculate the magnitude and wavelength for all pixels. Then for each images a set of features such as area, shape, diameter, and correlation, and a dynamic threshold are extracted. The image is inverted to separate the foreground and background and a binary image is created to diagnose the disease with a size of 1024 x 1024. Potential cancer areas are identified, and circles are drawn around them. The number of detected areas and their drawn circles are determined, the percentage of pixels that are white is computed and compared to the threshold. If the calculated value is greater than the threshold, then it's cancer, and its Adenocarcinoma, LCC, or SCC. Overall, this algorithm provides a comprehensive approach to detect and classify this disease from medical images with high accuracy and reliability. The model achieved the best accuracy of 99.42% when the VGG 19 and LSTM were combined.

- **Lung Cancer Nodule Detection**

The author [53] investigated a novel CAD system based on a Faster R-CNN (Region-based convolution neural network) model within an adaptive anchor box to detect the lung nodule where initially the feature extracted from CT images then Region Proposal Network (RPN) which is a subnetwork of R-CNN used for nodule detection and finally a classification was done for the nodule into nodule or background. This system was trained on real nodule sizes for generating adaptive anchor box sizes of Faster R-CNN. The efficacy of detecting lung cancer was enhanced using learned anchors as hyper-parameters. An RCNN (residual convolutional neural network) was put forward for mitigating the FPs (false positives) from the investigated system on the LUNA16 dataset. The experiment results confirmed that the investigated system yielded a sensitivity of 95.64% at 1.72 FPs (false positives) per scan.

A new CADSS [54] was constructed to detect the lung nodule based on 3DDCNN (three-dimensional-deep Convolutional Neural Network). Furthermore, an alternative viewpoint was provided to the radiologists to facilitate decision-making in the diagnosis of lung cancer. The 3D information was extracted from CT scans using MIP. The mRPN (multi-Region Proposal Network) was adopted to select the RoI (region of interest) automatically. The experimental results indicated that the constructed system offered a sensitivity of 98.4%, specificity of 92%, AUROC of 96%, and accuracy of 98.51%. This system performed well along with the CC (cloud computing) while detecting the lung nodule accurately.

A new method based on image fusion for lung segmentation for nodule detection was proposed in this study [55]. Laplacian Pyramid (LP) decomposition and Adaptive Sparse Representation (ASR) were used to construct the proposed image fusion technique, which divides medical images into various sizes and combines them using the LP decomposition. The suggested model scored a 0.9929 of The Dice Similarity Coefficient (DSC) index, which exceeded earlier released findings, according to the data. The algorithm additionally got competitive results for additional evaluation criteria like sensitivity (89%), specificity (98%), and accuracy (99%). The medical images can be represented on several scales thanks to the LP decomposition, which can enhance the efficacy of image fusion. Additionally, using the ASR approach improves segmentation and fusion results by reducing noise and artifacts. To shorten computation time and increase accuracy in complex circumstances, the proposed approach additionally integrates morphological operations and masking, negating the necessity for post-processing chores. In conclusion, the suggested algorithm offers a possibly more effective approach for lung segmentation and image fusion in the detection of lung cancer. In comparison to recently published results, the suggested algorithm is effective and competitive as evidenced by the high values obtained for the evaluation parameters.

Through the study conducted in [56] authors projected a TSCNN (two-stage convolutional neural networks) to detect the lung nodule. The primary phase focused on implementing the enhanced U-Net for detecting the lung nodule based on ResDense. Moreover, a two-fold predictive technique was also put forward. The next phase aimed to generate a dual pooling structure into three 3D-CNN algorithms to diminish the FPs based on SeResNet, DenseNet, and InceptionNet. A random mask was implemented as the technique for augmenting the data. In addition, ensemble learning was deployed to enhance the generalization capacity of the projected technique. The experimental results attained on the LUNA dataset confirmed that the projected technique was performed well for detecting lung cancer nodules.

During the development of a lung cancer detection system, various stages should be undertaken, including the preprocessing of medical images, image segmentation, and the utilization of machine learning techniques. However, it became evident that certain limitations existed within this approach, primarily inadequate preprocessing procedures. Addressing this issue necessitated the adoption of an effective segmentation technique capable of selecting the region of interest.

Moreover, the prevailing methodologies resulted in a reduced number of features available for the classification process, potentially leading to erroneous cancer diagnoses. To overcome this limitation, the proposed solution involves implementing a deep learning model. In this model, the initial stage extracts crucial features from the segmented images in the form of a sparse code. These features are then passed to the subsequent classification layer, which distinguishes between tumor and non-tumor features. This multifaceted approach focuses on comprehensive image preprocessing, accurate image segmentation, and feature extraction. By incorporating various deep learning models, which possess the capability to discern intricate patterns within data and enhance linear separation, the system aims to optimize each phase. It aims to identify the most effective combination of strategies for achieving accurate lung cancer detection.

Chapter 3
Methodology

1.1. Proposed Method

This chapter clarifies the methods used, covering both ML and DL components. The first stage involves the use of ML models on a dataset comprising clinical and demographic information. Subsequently, a performance evaluation will be conducted to choose the most effective model for enhancing the accuracy of lung cancer diagnosis. The initial phase of preprocessing is elucidated. Subsequently, the next section will describe the models employed in the experiment, namely Decision Tree (DT), Random Forest (RF), Gradient Boost (GBM), Light Gradient Boost (LGB), Extreme Gradient Boost (XGB), and Adaptive Boosting (AdaBoost). Figure 3.1.1 depicts the block diagram illustrating the ML model, whereas.

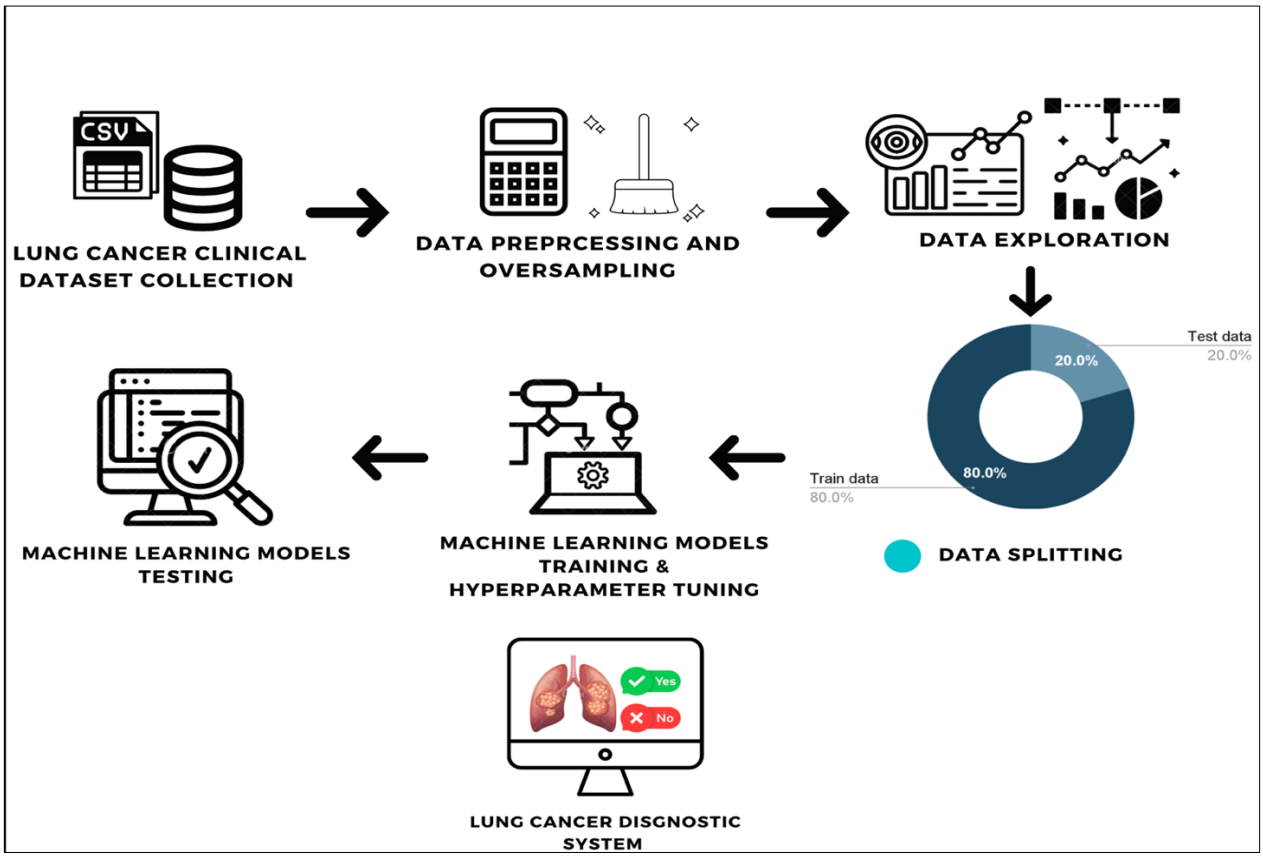


Figure 1.1.1 : The workflow of a machine learning-based model for diagnosing Lung Cancer based on clinical data.

The subsequent section showcases the DL steps, encompassing image preprocessing, augmentation, and the utilization of DL models including MobileNet, EfficientNet, DenseNet, CNN, Xception, ResNet, VGG16, and VGG19. Following this, a comparative analysis of the classification accuracy of the outcomes obtained. Figure 3.1.2 presents the block diagram representing the DL model.

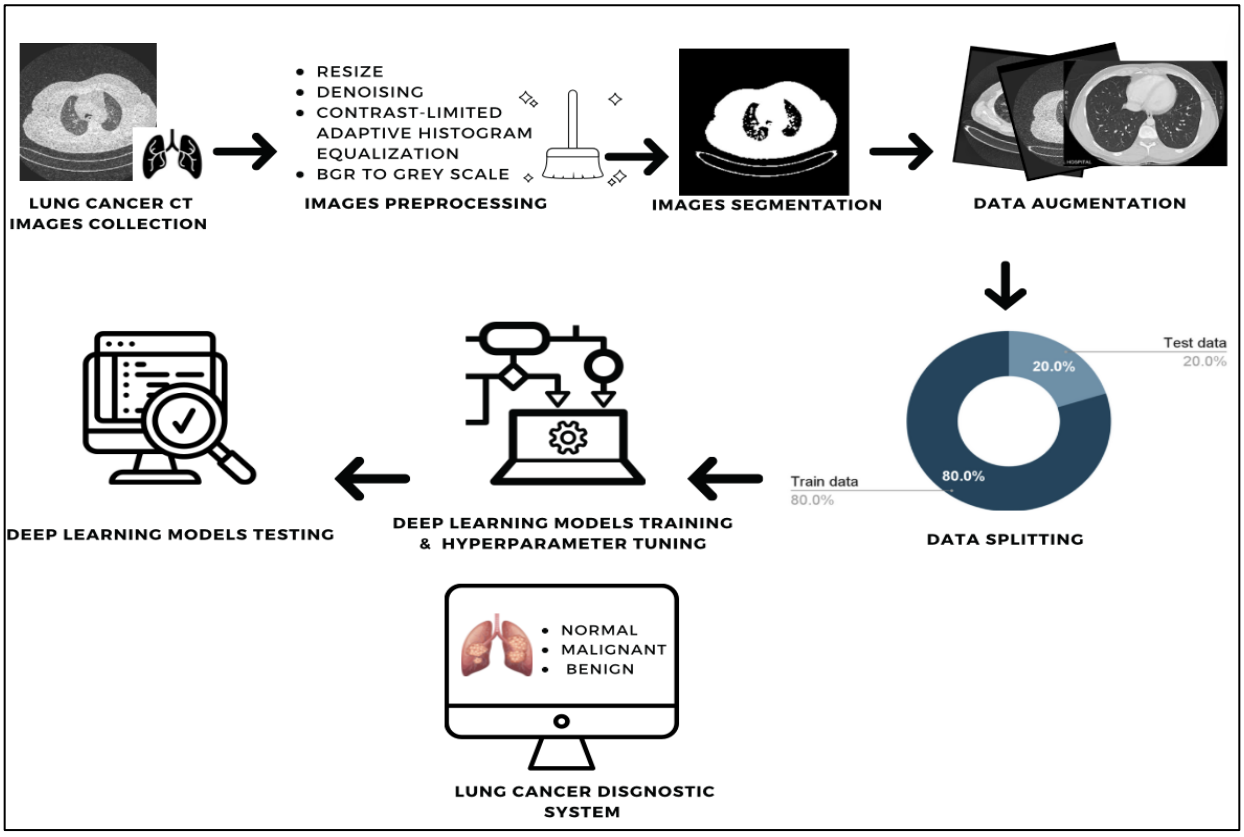


Figure 1.1.2 : The workflow of a deep learning-based model for diagnosing Lung Cancer based on CT Images.

1.2. Data Preprocessing Phase

Data preprocessing is considered a critical stage before being used in model training as it guarantees that the data is in a format that matches the ML/DL models as well as increases the data quality which subsequently improves the effectiveness of the trained models [57]. Many preprocessing techniques have been used on both datasets will be represented in the following section.

1.2.1. Numeric Data Preprocessing and Exploration

- **Label Encoder**

A Label Encoder was utilized to convert the labels of category features into numerical representations. This facilitates the application of ML/DL techniques in following phases. Table 3.2.1 below provides a thorough analysis of the labeling mapping procedure.

Table 1.2.1: Label Encoding Conversion.

Feature Name	Feature Labels Map
Gender	M (0), F (1)
Smoking	yes-2 (1), no-1(0)
Yellow Fingers	yes-2 (1), no-1(0)
Anxiety	yes-2 (1), no-1(0)
Peer Pressure	yes-2 (1), no-1(0)
Chronic Disease	yes-2 (1), no-1(0)
Fatigue	yes-2 (1), no-1(0)
Allergy	yes-2 (1), no-1(0)
Alcohol Consuming	yes-2 (1), no-1(0)
Wheezing	yes-2 (1), no-1(0)
Coughing	yes-2 (1), no-1(0)
Shortness of Breath	yes-2 (1), no-1(0)
Swallowing Difficulty	yes-2 (1), no-1(0)
Chest Pain	yes-2 (1), no-1(0)
Lung Cancer	Yes (1), No (0)

- **Data Normalization Using Standard Scaler**

Data normalization using the Standard Scaler was applied to preprocess the dataset before conducting further analysis. The dataset consists of binary data except for the 'age' column, which contains numerical values. The normalization process was designed to ensure that the 'age' column is appropriately scaled while preserving the binary nature of the rest of the data. The Standard Scaler formula in Equation 3.1 was followed to transform the 'age' column, scaling it to have a 0 mean and 1 standard deviation as shown in Figure 3.2.1 [58].

$$Z = \frac{X-\mu}{\sigma} \qquad (3.1)$$

Where X presents the value, μ is the mean, and σ is the standard deviation. This normalization step helps in addressing any variations in the 'age' feature, bringing all values

to a consistent and comparable scale. Meanwhile, the binary data was left unchanged, as it is inherently normalized within the range of 0 and 1. This will improve the accuracy and precision of subsequent analyses, ensuring fair treatment of each feature.

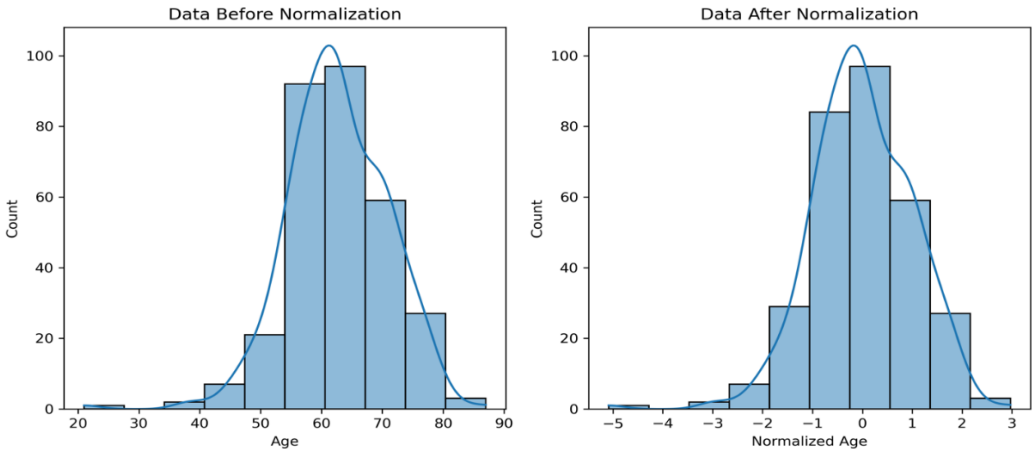


Figure 1.2.1: Age Field Before and After Standard Scaler Normalization.

- **Data Balancing Using Synthetic Minority Oversampling Technique (SMOTE)**

Through the exploration stage, it was clear that the data is unbalanced for the target class value as shown in Figure 3.2.3a. This could affect the testing accuracy, recall, and precision, leading to a bias in favor of the dominant class. As known in binary classification, unbalanced data results in findings that are skewed in favor of the majority class. Both under- and over-sampling are common solutions to this issue to provide balanced class data. It is common to use the (SMOTE) to enhance random oversampling. Observations are resampled around the dominant class to create balanced class data while avoiding the biasing problem using SMOTE [59]. The SMOTE algorithm is an over-sampling method used to address the class imbalance problem in machine learning. It aims to generate synthetic examples for the minority class to balance the class distribution. For each pattern in the minority class, SMOTE selects one of its K nearest neighbors from the minority class and creates a new synthetic pattern by interpolating between the selected pattern and the original pattern. The degree of interpolation is regulated by a random variable, leading to the creation of a fresh pattern along the line segment that connects the two existing patterns. The goal of this

algorithm is to move the classification decision boundary towards the region of the minority class by producing synthetic examples resembling the original minority class samples [60]. Figure 3.3.2 below contains a sample of how SMOTE is working.

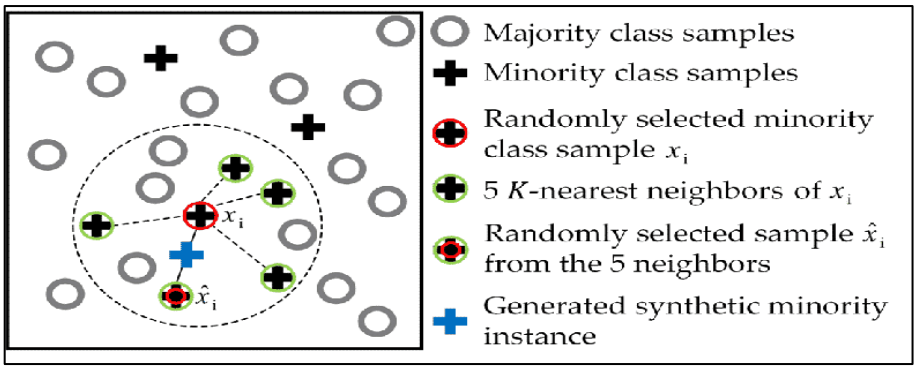
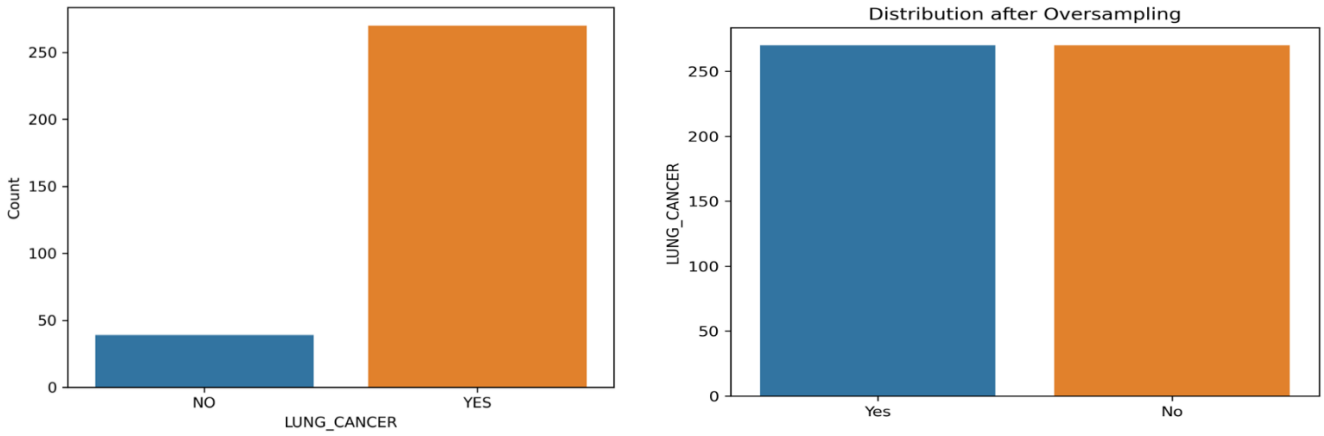


Figure 1.2.2 : SMOTE working procedure [61]

Figure 3.2.3(a) below shows how uneven the initial data samples are, and 3.2.3(b) the result after SMOTE.



(a) (b)

Figure 1.2.3: (a) Data Distribution Before Oversampling (b) Data Distribution After Oversampling.

1.2.2. Images Preprocessing and Enhancement

- **Resizing images**

In the realm of neural networks (NN), all images must be resized to a consistent size before they are input into the DL models. This standardization is imperative because NN exclusively accept inputs of the same dimensions. The degree of resizing required diminishes as the fixed size grows

larger. Minimizing the resizing process helps preserve the internal characteristics and patterns of the image, resulting in fewer distortions. Such an approach can effectively mitigate the reduction in classification accuracy attributed to the effects of resizing. However, it's crucial to note that larger images come with trade-offs. They demand a larger neural network and occupy more memory space, thereby amplifying complexity both in terms of space and time. Hence, the choice of this fixed size for images involves a trade-off between precision and processing efficiency [62]. To adapt images larger than the fixed size, two common methods can be employed: cropping the boundary pixels or downscaling using interpolation. Both methods entail some degree of information loss. Downscaling carries the risk of distorting image features or patterns, while cropping may eliminate relevant details located at the image's borders. In this context, downscaling is considered a favorable option for resizing larger images to the designated fixed size, as it poses a lesser risk of losing essential patterns. As part of this study, the images were uniformly resized to a specific dimension of 224 * 224 pixels, as shown in Figure 3.2.4.

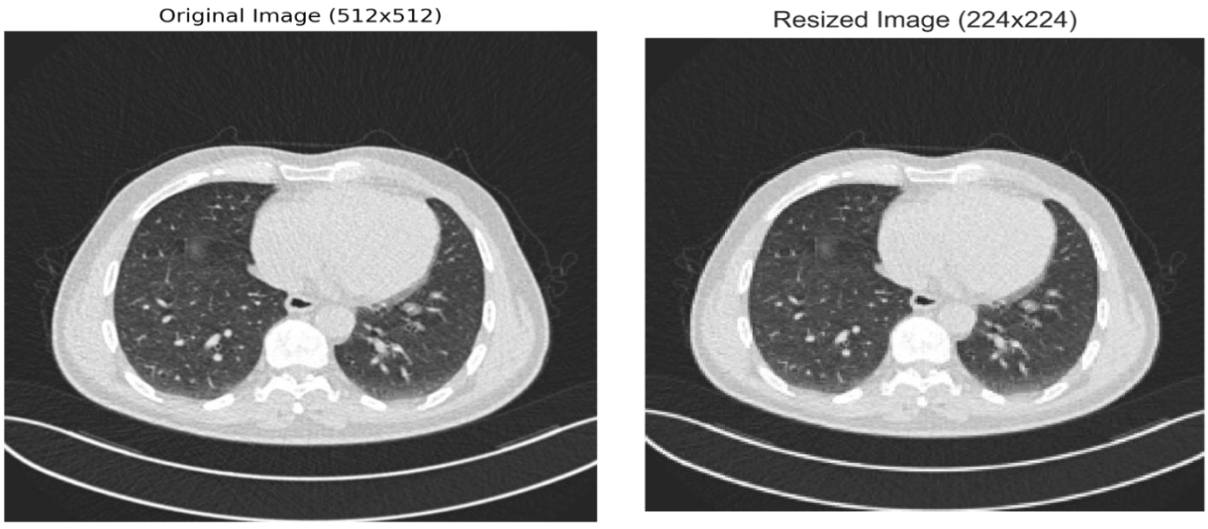


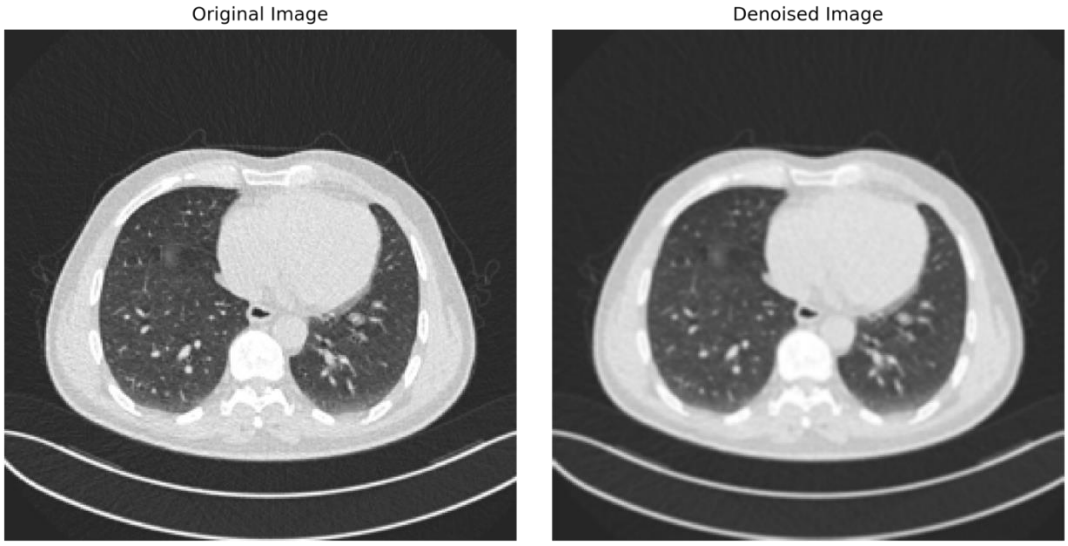
*Figure 1.2.4: (a) Original Image of Size 512*512 (b) Resized Image of Size 224*224*

- **Denoise the Image Using Gaussian Blur**

The Gaussian Blur technique is used to lessen the noise of an image by applying the Gaussian function. By deleting some of the unnecessary information and maintain the low spatial frequency of the image. The following Equation (3.2) provides the formula, where X is the direction and σ denotes the Gaussian distribution's standard deviation. The preferred implementation method is typically the two-pass Gaussian filtering approach, chosen for its requirement of fewer calculations [63], [64].

$$g(X) = \frac{e^{-\frac{x^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \tag{3.2}$$

Figure 3.2.5 shows an example of an image before and after applying the Gaussian Blurring.



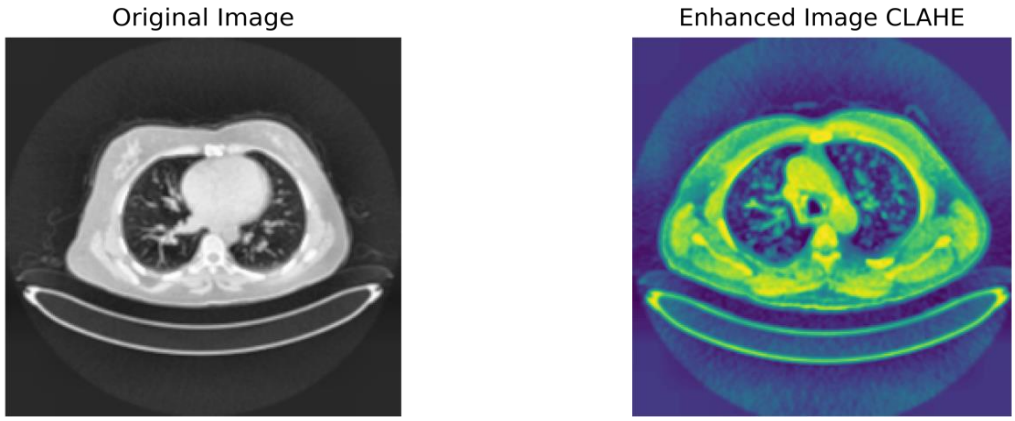
(a) (b)

Figure 1.2.5: (a) Original Image (b) Denoised Image After Applying Gaussian Blurring.

- **Contrast Limited Adaptive Histogram Equalization (CLAHE)**

CLAHE demonstrated the most significant improvement in enhancing prediction accuracy. To avoid excessive contrast enhancement that could lead to peculiar visual appearance and undesired artifacts, CLAHE employed a clipping value to limit the histogram. The utilization of inappropriate

contrast enhancement methods often resulted in poor representation of disappearing regions, such as tiny veins. However, this challenge was addressed by the enhanced CLAHE, incorporates a fixed contrast point in the filter by introducing a global threshold value. As a result, the histogram of the target image was enhanced in an adaptive manner using a predetermined global threshold value. The superiority of the enhanced CLAHE over the traditional CLAHE with fixed clipping where the proposed model significantly enhances image distinctiveness, particularly regarding the tiny veins. Furthermore, normalization was performed to widen the intensity distribution within an appropriate range, enabling more informative features to be inputted into the CNN model [65], [66]. Below Figure 3.2.6 is a sample image before and after applying CLAHE.



(a) (b)

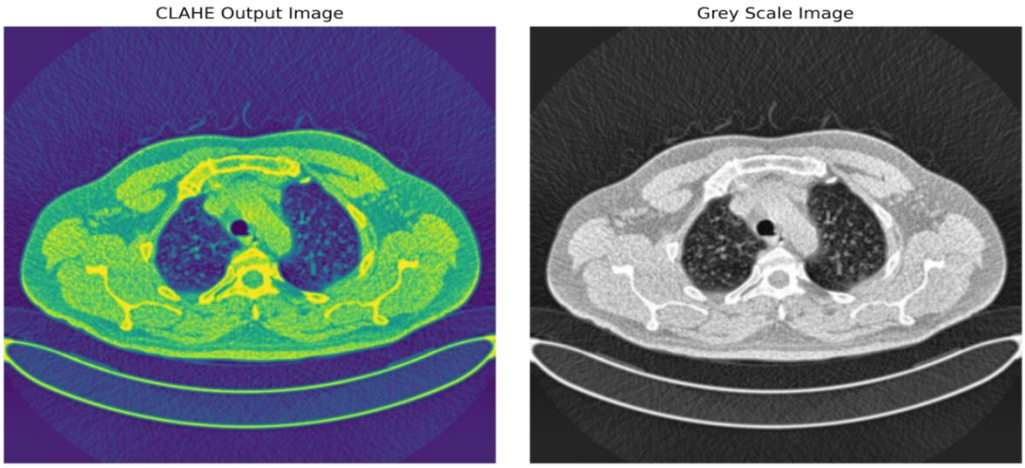
Figure 1.2.6: (a) Original Denoised Image (b) Equalized Image After Applying CLAHE Filtering

- **Convert Image from RGB (Red-Green-Blue) to Grey Scale**

It is anticipated that the process of converting RGB photos into grayscale images will reduce or mask image noise and make it simpler to distinguish between the intensity of colors in the image. An image can be converted to grayscale by selecting all its pixels and giving each one information about the three primary colors Blue, Green, and Red. To obtain an average value, the three primary colors will be put together and then divided by three. The three primary colors of a pixel will be set to an average value to add color to the image pixels, converting the color to grayscale [67]. As

the output of the previous preprocessing step of applying CLAHE filtering was an RGB image, it was necessary to convert it back to a grey-scale image again as this will be more efficient in deep learning training concerning time and cost. This conversion can be represented by the following equation:

$$Gray = 0.299 \times Red + 0.587 \times Green + 0.114 \times Blue \quad (3.3)$$



(a) (b)

Figure 1.2.7: (a) CLAHE Equalized Image (b) Grey Scale Image After BGR to Grey Conversion.

- **Image Segmentation Using Otsu's Thresholding**

A basic step in preprocessing medical images in computer-aided applications is segmentation. Segmentation is a process that helps divide a digital image into various pieces such that the pixels in each area have comparable visual characteristics. The goal is to make an image's representation simpler or different to better examine it [68]. Thresholding is a simple segmentation technique that extracts the area of interest. Using a threshold, you may distinguish between foreground and background pixels. The intensity of the processed image is changed halfway between the minimum and maximum levels of the native intensity after the gray threshold is applied using the Ostu method [69] [70]. This method developed by Otsu splits the image into foreground and background using a threshold value. The threshold value continuously rises until it reaches the highest variance between pixels belonging to two groups [71]. The main objective of Otsu's strategy is to offer the

ideal threshold value. By using bimodal histograms to group the pixels into the C1 and C2 categories, the threshold value is calculated. Additionally, it lowers the intraclass variance by selecting the proper threshold value. For the implementation, we used Python's threshold function. The equation for Otsu's thresholding is as follows:

$$\sigma_w{}^2(t) = w_1(t) \cdot \sigma_1{}^2(t) + w_2(t) \cdot \sigma_2{}^2(t) \quad (3.4)$$

Where: t is the threshold value being considered, $\sigma_w{}^2(t)$ is the weighted variance between the classes, $w_1(t)$ is the weight of class 1 (pixels with intensity values below the threshold), $w_2(t)$ is the weight of class 2 (pixels with intensity values above the threshold), $\sigma_1{}^2(t)$ is the variance of class 1, and $\sigma_2{}^2(t)$is the variance of class 2.

The results of Otsu's approach being applied to a CT scan of lung cancer are shown in Figure 3.2.8 below:

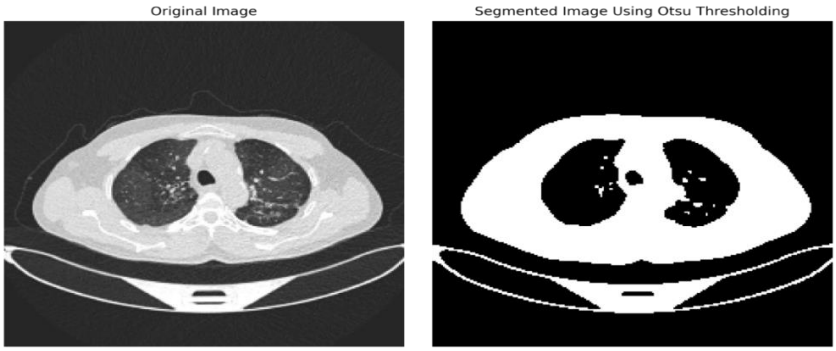


Figure 1.2.8: Samples To illustrate The Results of Applying Otsu's Thresholding for Segmentation.

- **Images Augmentation**

DL has made enormous advances in computer vision over the last several years due to effective computing tools, potent algorithms, and large amounts of image data. The creation of a powerful model with many trainable parameters has been a major effort. For such a model to perform competitively, a high number of photos must be used during training. However, gathering images is typically a costly and difficult procedure. In practical applications like medical and agricultural pictures, and it's particularly difficult to achieve adequate performance with a small dataset. To

tackle that issue, it has been established that image augmentation is a viable and proficient approach [72]. This can be accomplished by the utilization of diverse methodologies, such as horizontal flipping, color space augmentations, random cropping, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, neural style transfer, and meta-learning schemes [73]. In this context, horizontal flipping and histogram equalization are employed for the purpose of image augmentation and at the same time achieve balance between the three different categories, as shown in Figure 3.2.9.

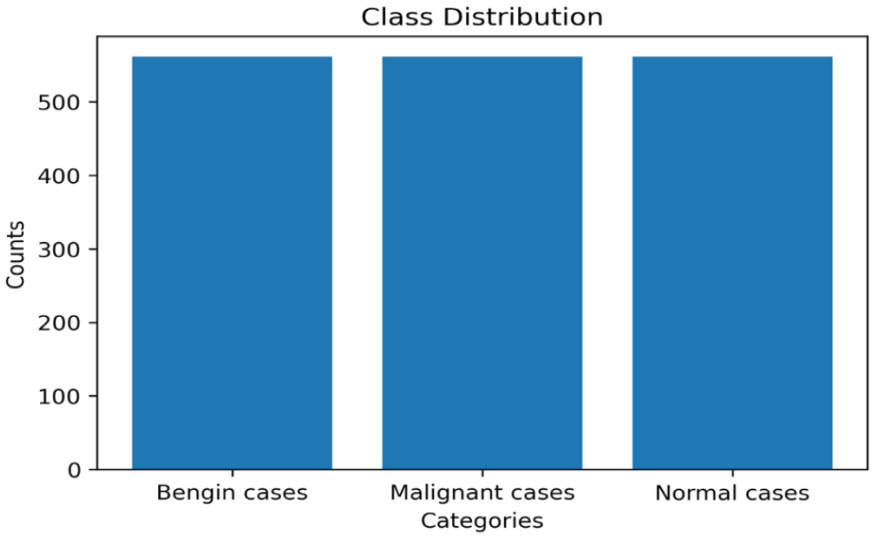


Figure 1.2.9: Data distribution dataset after data augmentation

- **Horizontal Flip Augmentation**

Flipping an image refers to the rotation of the image along the horizontal or vertical axis, such as the rows, and columns of pixels in an image are reversed horizontally. This means that the pixels on the image's left side are exchanged with those on the right side., resulting in a mirror image of the original along the vertical axis [73]. Figure 3.2.10 shows an example of Horizontal Flipping.

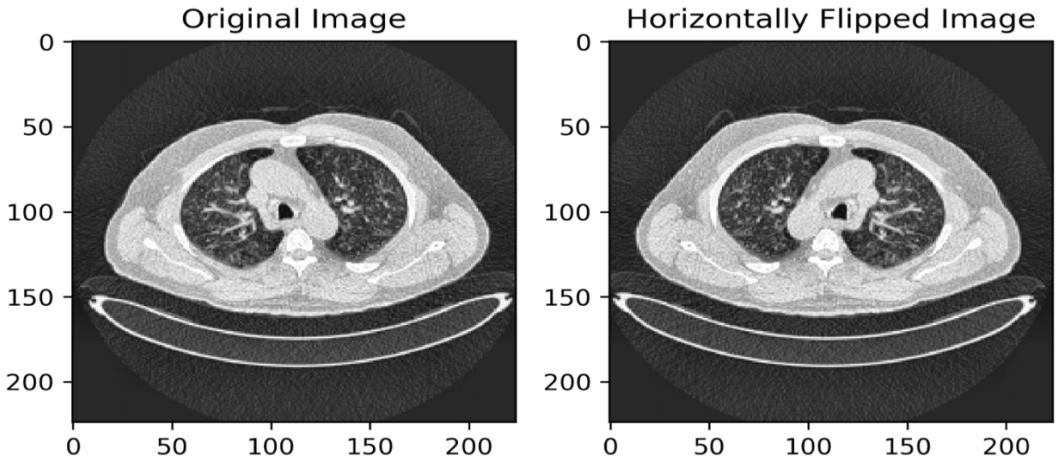


Figure 1.2.10 : Original Image and Horizontally Flipped Image

- **Histogram Equalization Augmentation**

Equalization of the histogram demonstrated in Figure 3.2.11 is the distribution of a specific kind of data which allows enhancing an image's contrast. By using histogram equalization, the entire range of pixels (0–255) will be widened. A decent histogram is the type employed by grayscale potential values. Applying this can be done using various methods such as histogram expansion, odd sectioning, and cumulative histogram equalization which will be used in this study [74]. The proposed approach encompasses a series of sequential procedures. Initially, the histogram of the original image is generated. Subsequently, the cumulative distribution function histogram is computed. Following this, the generic histogram equalization formula is employed to calculate new values. Finally, new values are assigned to each gray value inside the image. This is applied based on the following algorithm:

Let X be an image that consists of L discrete grey levels. Where the probability density for the image C can be calculated as:

$$P(X_k) = \frac{n^k}{n} \tag{3.5}$$

For each level k, where k ranges from 0 to L-1, n_k denotes the frequency of occurrences of level X_k in the input image X, and n represents the total number of samples in the input image.

p(X_k) is linked to the histogram of the input image, indicating the count of pixels with a particular intensity X_k.

Using the density function, the cumulative density is calculated as:

$$C(x) = \sum_{j=0}^{k} P(X_j) \tag{3.6}$$

Where X_k as X for k = 0, 1, ..., L-1 and c(X) is equal to 1.

Histogram Equalization (HE) is a method that remaps an input image to span the entire dynamic range (X_0, X_(L-1)) using a cumulative density function as a transformation function. The transformation function, denoted as f(x), is based on the cumulative density function and can be defined as:

$$f(x) = X_0 + (X_{(L-1)} - X_0) * c(x) \tag{3.7}$$

The output image, denoted as Y = {Y (i, j)}, can be expressed as Y = f(x), which is further defined as:

$$f(Y) = f(X(i, j)) / \forall X(i, j) \in X \tag{3.8}$$

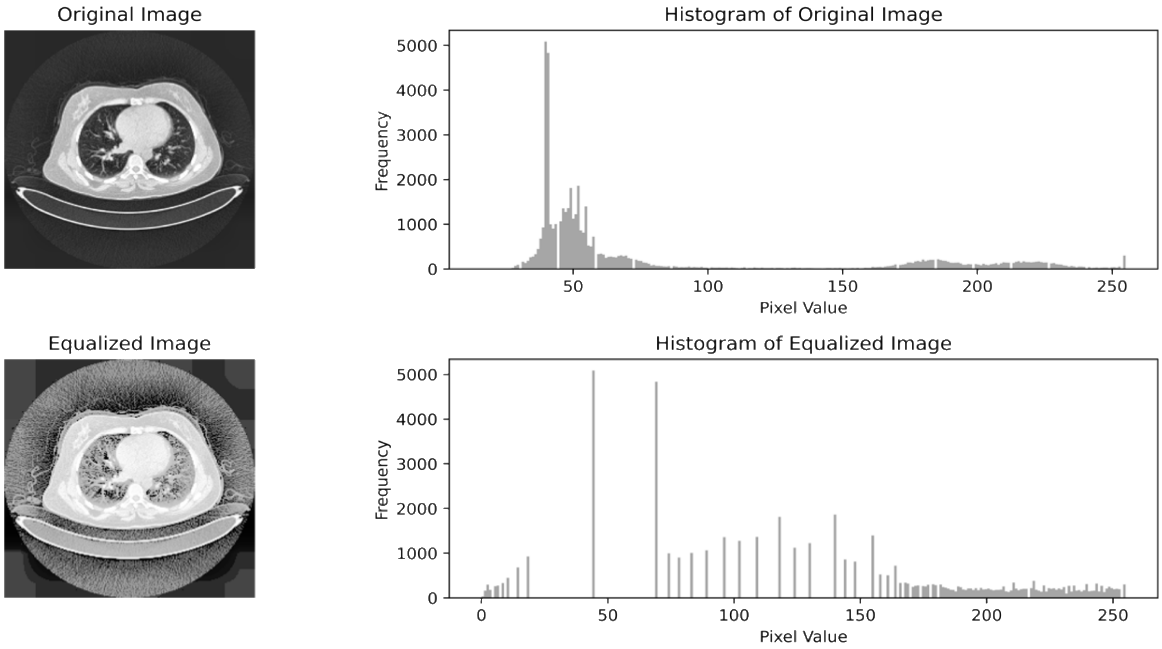


Figure 1.2.11 : Original and Equalized Images with Their Correspondence Histograms.

1.3. Building And Training Models Phase

This section of methodology will present the different models that have been applied to the used datasets including both ML models and DL models.

1.3.1. Decision Tree

The decision tree algorithm is categorized as a supervised learning technique. It has applications in solving both regression and classification problems. The fundamental idea [75] behind the decision tree (DT) algorithm involves representing a problem using a tree structure. In the given depiction, every internal node within the tree corresponds to an attribute, and each leaf node denotes a class label [76]. This method begins by observing a specific element and draws conclusions about its value through the leaves of the tree, with its various attributes represented by the branches. DT algorithms produce a set of decision rules that lead to determining the class or test result. The approach is based on a process of recursive node division, starting from the root node containing all records, and progressively dividing the dataset into smaller subgroups until reaching the target class nodes [77]. This classification process follows a top-to-bottom approach, where the root node initially encompasses all records and splits based on the class label column, which forms the foundation of the classification [78]. Figure 3.3.1 below represents the overall structure of a Decision Tree.

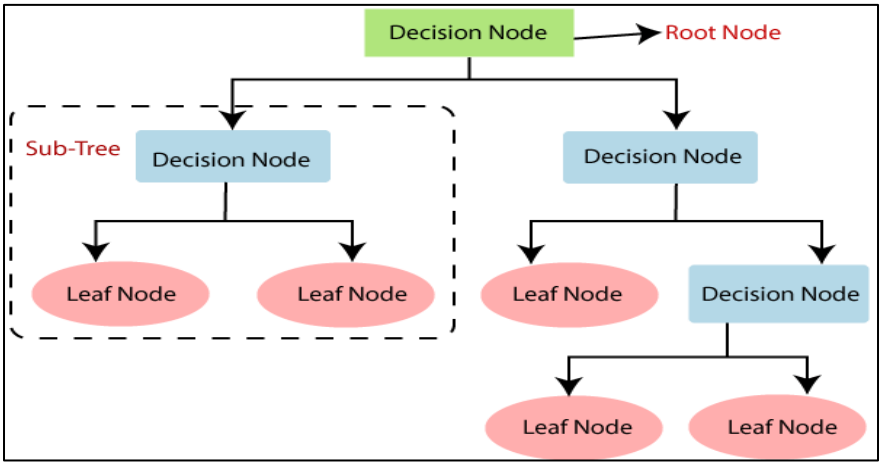


Figure 1.3.1 : General Decision Tree Structure [79]

The establishment of a Decision Tree comprises the sequential execution of the following steps:

1. Splitting: This stage entails dividing the dataset at each node into two or more sets based on attribute descriptors. The division is determined by measures of information gain.

- **Entropy:** This concept, introduced by information theory pioneer Claude Shannon [80], gauges the disorder in a system. Entropy quantifies the probability of variance within a node, indicating whether the node contains more than one class. It ranges between 0 and 1, with 0 denoting purity (only one class) and values closer to 1 representing higher randomness. Its equation is:

$$E = -\sum_{i=1}^{N} pi \log_2 pi \tag{3.9}$$

- **Information Gain:** Calculated to identify the best attribute for splitting, it aims to maximize attribute purity in a branch. Information Gain is computed as 1 minus the entropy.

$$Information\ Gain\ =\ 1 - Entropy \tag{3.10}$$

2. Stopping: To prevent overcomplicated models, stopping rules are essential. These rules minimize various factors including the count of records in a leaf, the depth (number of levels), and the count of records before splitting.

3. Pruning: DT can be prone to overfitting, where the model shows significant adaptation to the training data and lower results on testing data due to the presence of noise. To mitigate this, pruning is employed to simplify the tree while maintaining high accuracy. The ideal tree balances precision with simplicity and minimal depth [81].

1.3.2. Random Forest (RF)

Random Forest, as its name suggest, is a ML technique that leverages the power of multiple decision trees – often numbering in the hundreds or thousands – to collectively formulate predictions. This algorithm operates within the realm of supervised learning, proving valuable for

both classification and regression tasks [82]. The base of RF involves a process wherein a subset of features is randomly chosen from the dataset, subsequently forming a decision tree based on this selected subset. This iterative process is repeated multiple times, generating a multitude of decision trees whose outcomes are then amalgamated to yield an overarching prediction. What truly makes RF a robust method lies in its adeptness at handling voluminous datasets and accommodating both numerical and categorical features. Moreover, its capacity to manage missing data and outliers enhances its suitability for various applications. Fundamentally, the underlying principle of the RF algorithm is elegantly simple yet remarkably potent. In the domain of data science, the collective strength of numerous relatively uncorrelated models (trees) working in harmony surpasses the performance of any individual constituent model. To enhance the accuracy of RF, one can fine-tune its parameters through optimization, such as adjusting the node size and the count of estimators [83]. Figure 3.3.2 demonstrated the general concept of RF.

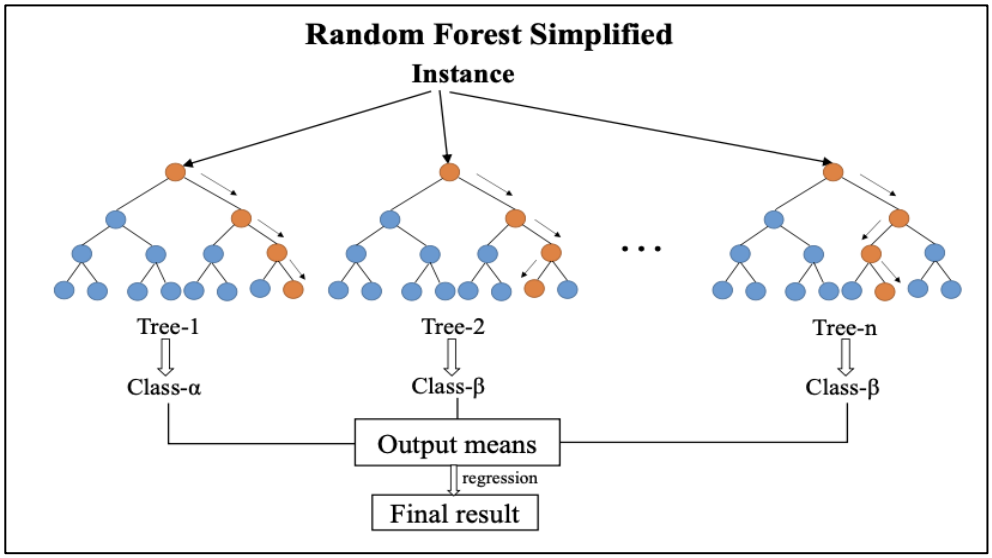


Figure 1.3.2 : Random Forest General Concept [83]

RF operates as a form of ensemble learning, regulating multiple decision tree models to generate predictions. The mathematical expression encapsulating the essence of Random Forests is elucidated below:

$$(x) = 1/N \sum i\ (fi(x)) \tag{3.11}$$

Where:

F(x) signifies the combined output of the Random Forest algorithm, N represents the total number of decision trees within the forest, fi(x) denotes the outcome of the ith decision tree within the forest and x corresponds to the input data.

1.3.3. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are employed for performing classification as well as regression tasks. The fundamental concept underlying this approach involves the identification of a hyperplane that effectively partitions a given dataset into two distinct classes [84]. This process, depicted in Figure 3.3.3, entails the selection of a hyperplane that optimally separates the data points, with the two closest points to the hyperplane serving as the supporting vectors.

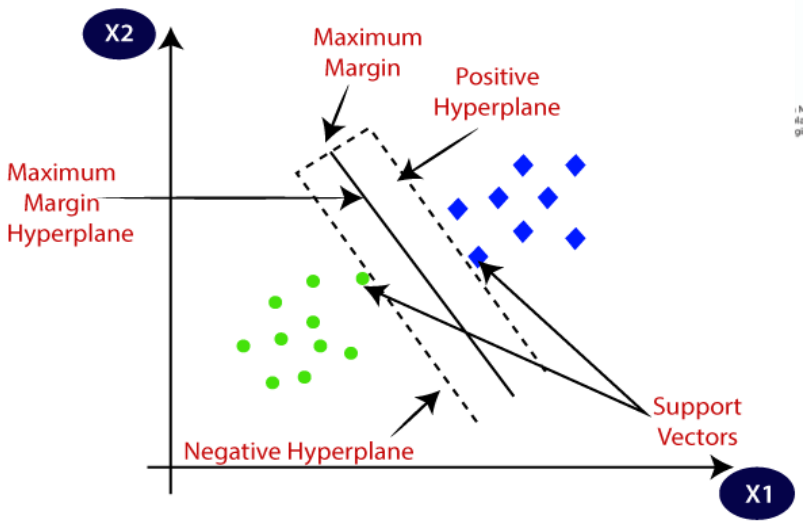


Figure 1.3.3 : SVM Hyper Plane Concept [85]

Support vectors refer to the data points near the hyperplane, which is the dividing boundary inside the dataset. These points possess the characteristic of significantly influencing the position of the hyperplane if they were to be eliminated from the dataset. These points can be regarded as significant components within the dataset. While the hyperplane is defined as a linear boundary that effectively separates and categorizes the given dataset. From an intuitive perspective, it may be inferred that the greater the distance between points and the hyper-level, the higher the level of

confidence in the right ranking of the points. The objective is to maximize the distance between the points and the hyperplane while ensuring that they remain on the positive side of the decision boundary. When incorporating additional data for testing purposes, it will be classified according to its position relative to the hyperplane. The margin refers to the distance between the hyperplane and the nearest point within a specified dataset. The goal is to select a hyperplane that exhibits the maximum margin from any point within the training dataset, hence enhancing the probability of accurately classifying any future data [86]. The hyper-plane equations for a given training dataset:

$$(x_1,y_1), \ldots, (x_n,y_n),\ x_i \epsilon\ R^d \text{ and } y_i\ \epsilon\ (-1,1) \tag{3.12}$$

Let Xi represent the feature vector symbol and Yi represent the class label (either 1 or 0) of training compound i. Therefore, to denote the equation of the optimal hyperplane:

$$wx^T + b = 0 \tag{3.13}$$

In the given context, the weight vector is denoted as W, the feature vector as X, and the bias as b. Both variable b and variable w need to meet the variance requirements for the components inside the training dataset.

$$wx_i^T + b \geq +1 \text{ if } y_i = 1 \tag{3.14}$$

$$wx_i^T + b \leq -1 \text{ if } y_i = -1 \tag{3.15}$$

The primary goal of the training dataset is to identify the appropriate values for B and W that can effectively separate the data using an optimal hyperplane while maximizing the margin distance of $1/\| w \|^2$ [86]. The concept of vectors is a fundamental topic in mathematics and physics. Vectors are mathematical entities The support vectors are defined as xi for which the absolute value of yi is multiplied by the dot product of wxiT and b equals 1. Ultimately, The SVM model offers several advantages, including its effectiveness on small and refined data sets, as well as its efficiency through the utilization of a portion of the training data. On the contrary, it is essential to know that this approach is not well-suited for handling huge datasets due to the significant amount of time

necessary for training the data. Additionally, its effectiveness diminishes when dealing with noisy and overlapping data that exist between different classes.

1.3.4. Gradient Boosting

Gradient boosting is an ensemble approach that produces a collective forecast by merging multiple individual guesses, referred to as "weak" predictions. It is used as Supervised machine learning, particularly in classification and regression tasks. The objective of the procedure is to progressively refine a model through iterative training, wherein faults from the previous model are identified and corrected to improve the predictive abilities of the model progressively. The GB method is a decision tree-based approach that is widely recognized and commonly used in several domains. It possesses the capability to effectively handle missing values and outliers, making it a valuable tool in data analysis and modeling. The concept underlying GB is that the performance of a weak classifier can be enhanced by iteratively fitting additional weak classifiers while maintaining the knowledge gained from earlier iterations. The procedure is executed in an iterative manner, wherein each subsequent classifier acquires knowledge from the shortcomings of its predecessors. This iterative process leads to a decrease in the overall error and an enhancement in the accuracy of the classifiers [87]. The mathematical equation for Gradient Boosting can be expressed as follows:

$$F(x) = \sum_{i=0}^{m} f_i(x) \tag{3.16}$$

where F(x) represents the output prediction, m is the number of estimators, and fi(x) represents the prediction of the ith estimator. The function F(x) can be represented as the sum of m individual functions fi(x), where i ranges from 1 to m.

1.3.5. Extreme Gradient Boosting (XGB)

Extreme Gradient Boosting (XGBoost) is a ML algorithm that leverages the collective power of multiple weak learners to generate a highly effective predictive model [88]. Per Chen and Guestrin [15], Xgboost involves an ensemble of K Classification and Regression Trees (CART), represented as {T1(xi, yi), . . ., TK (xi, yi)}. In this context, xi refers to the feature associated with a recored in the training dataset, while yi signifies the class label to be predicted. Each CART assigns a real score to its leaves, which represent outcomes or targets. The final score is obtained by summing the prediction scores from each CART and evaluated using K additive functions, as depicted in Equation (3.17).

$$\widehat{y_i} = \sum_{k=0}^{K} f_k(x_i), f_k \in F \tag{3.17}$$

In this context, fk represents an independent tree structure with leaf scores, and F represents the space of all Classification and Regression Trees (CART). The regularized objective to be optimized is defined by Equation (13.18).

$$Obj(\ominus) = \sum_{i}^{n} l(y, \hat{y}_i) + \sum_{k}^{K} \Omega(f_k) \tag{3.18}$$

1.3.6. Light Gradient Boosting (LGB)

LightGBM is a ML technique with its roots in decision trees and boosting. This framework was first introduced by [89]. It sets itself apart from the XGBoost model by utilizing histogram-based algorithms, which bring about several advantages. In this approach, continuous floating-point values are discretized into k bins, creating a histogram with a width of k. Unlike other methods, this histogram algorithm doesn't require extra storage for pre-sorted results. Moreover, it can efficiently store feature values post-discretization using only 8-bit integers, leading to a substantial reduction in memory consumption, down to just 1/8 of the original amount. Interestingly, this

coarse partitioning doesn't sacrifice model accuracy; in fact, it acts as a form of regularization, effectively preventing overfitting. One of the key ideas to grasp here is that decision trees, in the context of LightGBM, are considered weak learners. As such, the precision of segmentation points isn't as critical. Even coarser segmentation points introduce a regularization effect, which further helps guard against overfitting. LightGBM adopts a leaf-wise growth strategy, a more efficient alternative to the level-wise approach. With the leaf-wise strategy, the algorithm identifies leaves with the highest branching gain at each step and expands them. This results in more effective error reduction and higher precision using same segmentation steps compared to the level-wise strategy. However, it's crucial to note that leaf-wise growth can lead to deeper decision trees and an increased risk of overfitting. To mitigate this, LightGBM imposes a maximum depth limit on the topmost leaves, ensuring both efficiency and overfitting prevention [90].

1.3.7. Adaptive Boosting (AdaBoost)

The AdaBoost method constructs an ensemble of weak learners by managing the weights assigned to the training data, then iteratively modifies these weights Following each weak learning cycle, the weights assigned to training samples misclassified by the current weak learner will be increased, whereas the weights assigned to correctly classified samples will be decreased. [91]. One of the primary concepts behind the AdaBoost algorithm is the preservation of an ensemble of weights over the training set. The weight assigned to training example i in round t is represented as Dt (i). At the outset, all weights are uniformly initialized. However, in each cycle, the weights of misclassified instances are augmented, compelling the weak learner to prioritize challenging examples within the training set. The primary task of the weak learner is to identify a weak hypothesis that is suitable for the distribution of Dt [92]. The Algorithm of Adaptive Boosting (AdaBoost) can be summarized in the following steps:

1. Initialize the observation weights wi = 1/N, i=1, 2, ..., N. Where N = Number of observations.
2. For m = 1 to M:

- Fit a classifier Gm(x) to the training data using weights wi
- Compute:

$$\mathrm{err}_m = \frac{\sum_{i=0}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=o}^{N} w_i} \quad (3.19)$$

- Compute:

$$\alpha_m = \log\left(\frac{1 - \mathrm{err}_m}{\mathrm{err}_m}\right) \quad (3.20)$$

- Set:

$$w_i < -w_i * \exp\left[\alpha_m * I\left(y_i \neq G_m(x_i)\right)\right], i = 1,2,..,N \quad (3.21)$$

3. Output:

$$G(x) = \mathrm{sign}\left[\sum_{m=1}^{M} \alpha_m \; G_m(x)\right] \quad (3.22)$$

1.3.8. Visual Geometry Group-16 (VGG16) Model

The VGG16 architecture represents a foundational convolutional neural network (CNN) model frequently utilized for image analysis tasks. As shown in Figure 3.3.4, it consists of 16 layers and is commonly employed as a baseline example of CNN design. This architecture was initially pre-trained on the ImageNet dataset, a comprehensive collection of labeled images spanning diverse object categories, serving as a prominent benchmark for image classification. The VGG16 network takes an RGB image input with dimensions 224 × 224 pixels. It commences with two convolutional layers, each equipped with 64 filters, followed by a subsequent max pooling layer. This sequence is reiterated in three more blocks, resulting in a total of four sets of convolutional and pooling layers. Subsequent blocks involve trios of convolutional layers featuring 128, 256, and 512 filters respectively, followed by another max pooling layer. The final block culminates in three convolutional layers, each comprising 512 filters. Beyond the convolutional layers, the architecture integrates fully connected (dense) layers. It incorporates an initial dense layer with 4096 units, followed by a dropout layer aimed at mitigating overfitting. Subsequently, another dense layer with 4096 units follows. This layer often functions as a feature extractor during transfer

learning scenarios, facilitating the extraction of high-level representations from input images for classification purposes. Notably, the 1000th layer in the VGG16 architecture yields a vector encompassing 1000 elements, signifying the model's prediction confidence regarding objects or concepts within an image [93]. In the context of the DL network, the 4096 features derived from the VGG16 model serve as pivotal input layers, underscoring their significance for subsequent analyses and tasks.

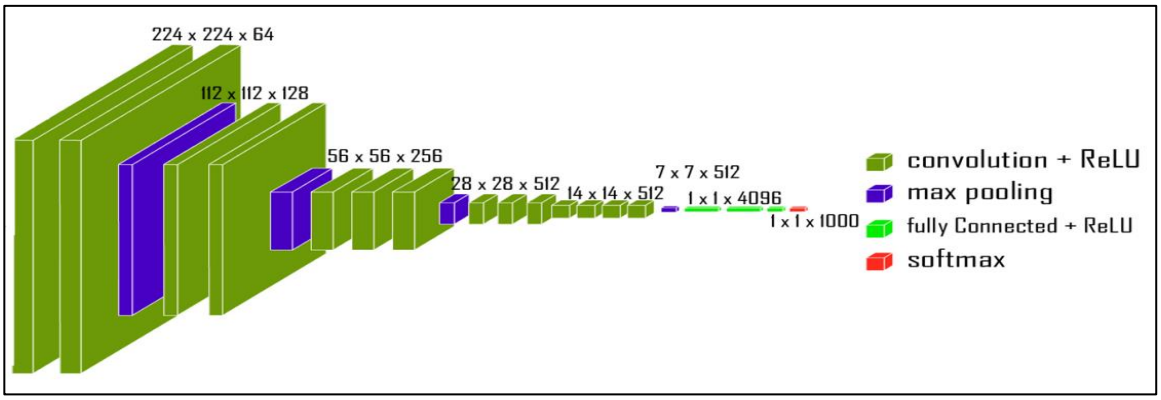


Figure 1.3.4: VGG16 pretrained CNN model architecture [94].

1.3.9. Visual Geometry Group-19 (VGG19) Model

Deep learning networks have found applications in various domains for image classification tasks, leveraging extensive datasets with approximately 60 million parameters and around 650,000 neurons. In practical scenarios, the network's structure typically encompasses five convolutional layers and three fully connected layers, each assigned distinct roles. The architectural layout consists of dual initial convolutional layers (a standard layer followed by a max-pooling layer), the subsequent 3rd and 4th convolution layers interconnected directly, succeeded by a final convolution layer (max-pooling layer). This culminates in the output layer (Softmax layer). It's worth noting that some networks boast specialized architectures tailored for specific applications. Take GoogleNet, for instance, which incorporates approximately 7 million parameters, 9 inception modules, 4 convolutional layers, 4 max-pooling layers, 3 average pooling layers, 5 fully connected layers, and 3 Softmax layers [95]. Throughout, Relu activation functions are employed in

convolutional and dropout layers, with a 70% parameter reduction ratio universally applied to fully connected layers [96].

VGG-19, a deep-learning neural network, boasts 19 connection layers, comprising 16 convolutional layers and 3 fully connected layers. These convolution layers serve to extract intricate features from the input images, while the fully connected layers undertake the classification of leaf images based on these derived features. To mitigate overfitting and streamline feature extraction, max-pooling layers are strategically incorporated, as illustrated in Figure 3.3.5 [97].

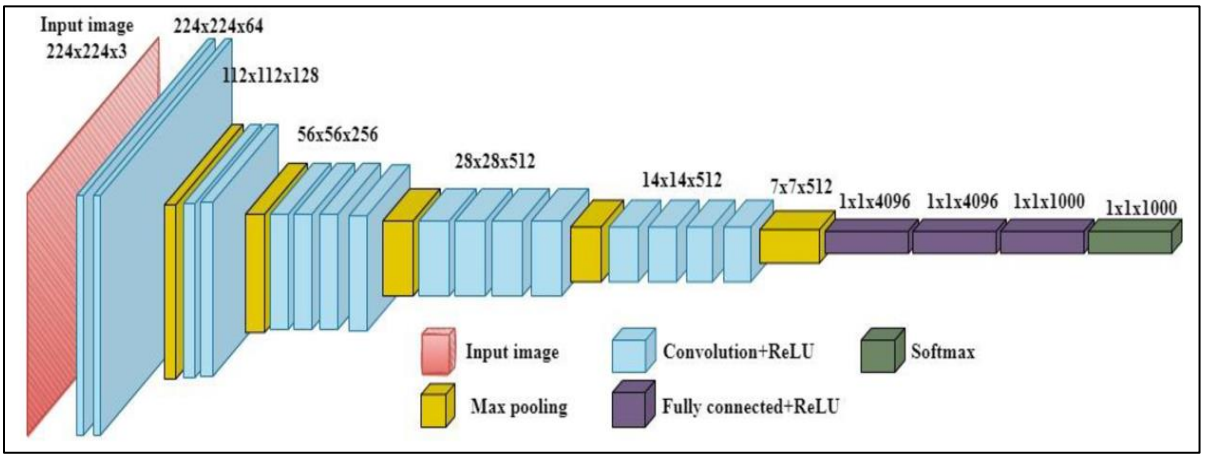


Figure 1.3.5: VGG19 pre-trained CNN model architecture [97]

1.3.10. Densely Connected CNN (DenseNet121) Model

Deep Convolutional Networks (DCNNs) have emerged as a highly effective architecture for image recognition, leveraging specific convolutional and pooling layers. However, as networks deepen, the challenge of vanishing gradients becomes apparent, where gradients diminish as they propagate through numerous layers toward the final layer of the network. To address this issue, DenseNets introduces a solution by directly connecting all layers with identical feature sizes. This strategy mitigates gradient vanishing and empowers DenseNets to capture both broad and deep network features effectively. The pre-trained Connectivity-Densified Convolutional Neural Network, comprising 121 layers, is employed for feature extraction [98].

The architecture of DenseNet-121 in Figure 3.3.6 encompasses 120 Convolutional layers, 4 Average Pooling layers, and 1 Fully Connected Layer. It commences with a basic convolution layer and pooling layer, followed by 3 transition layers and 4 dense blocks. The classification layer concludes the architecture. The initial convolutional layer employs a 7x7 convolution with a stride of 2, followed by 3x3 max pooling with a stride of 2. Subsequently, a dense block is integrated, succeeded by 3 sets, each containing a transition layer and a dense block. As per Huang et al.'s DenseNets concept [98], dense connectivity is achieved through direct connections between any layer and all previous layers, facilitating improved gradient flow across the network. This necessitates concatenating feature maps of prior layers, a process challenging when feature map sizes differ. To address this, transitional layers are inserted between dense blocks, consisting of batch normalization, a 1x1 convolutional layer, and a 2x2 average pooling with a stride of 2. The architecture comprises multiple dense blocks interconnected by these transitional layers. Specifically, the DenseNet169 architecture, pre-trained on ImageNet, employs 4 dense blocks with convolution layers of 1x1 and 3x3 dimensions, arranged in sequences of 6, 12, 24, and 16 blocks. The final layer performs global average pooling with a size of 7x7, followed by a fully connected layer activated by "Softmax" [99].

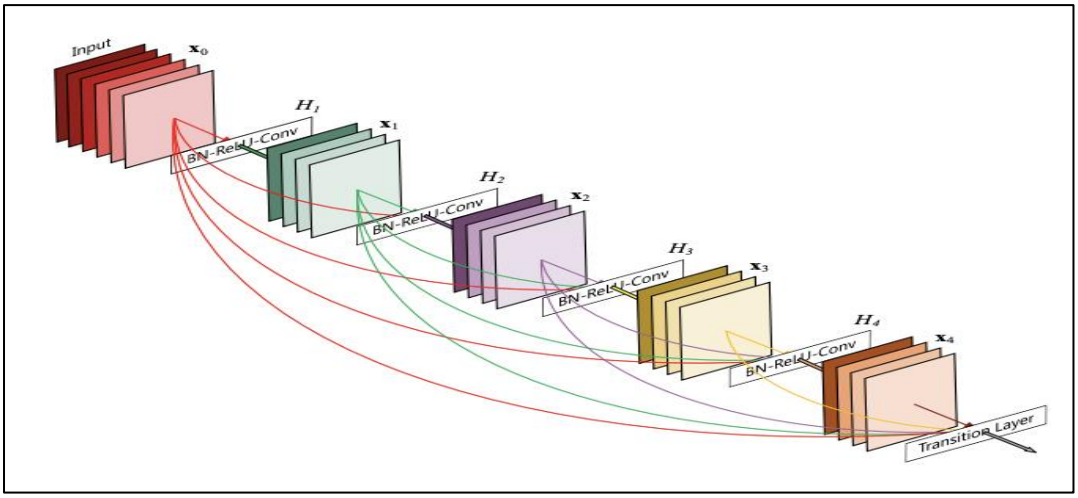


Figure 1.3.6: DenseNet121 Architecture [100]

1.3.11. Residual Networks (ResNet50) Model

ResNet's fundamental idea is to establish an identity shortcut connection that skips one or more layers [95]. It uses a method called bypass connections to connect skips on two to three layers that contain batch normalization and ReLU among the architectures. Instead of allowing the network to learn the underlying mapping, it enables the network to match the residual mapping. Consider H(x) as an example of an underlying mapping that can be fitted by several stacked layers, with "x" designating the inputs to the first layer. If it is assumed that many nonlinear layers can asymptotically approximate complex functions, this is equivalent to assuming that they can asymptotically approximate the residual function F(x): = H(x) x. Thus, the initial function is changed to F(x) + x. The learning curves for each form may differ, even though it should be possible for both to asymptotically resemble the required functions (as was initially intended). The advantage of employing the skip link is that regularization will skip any layer which reduces the design performance. As a result, unlike traditional CNN models, very deep neural networks can be trained without being hindered by vanishing gradients. Like how they are utilized in long short-term memory (LSTM) networks, parametric gates are used in these skip connections. The amount of data that travels via the skip connection is managed by these gates. When ResNet trains an excessive number of deep CNNs, it addresses the gradient vanishing and feature map vanishing problems. The ResNet works because identity linkages between non-adjacent layers do not affect the ideal mapping that the application task wishes to build. Backpropagation is more fluid since gradients have access to a second shortcut channel because of the identity relationship. In conclusion, ResNet50 is a well-liked deep residual network that successfully strikes a compromise between network depth and training effectiveness [101] [102]. Figure 3.3.7 displays a typical ResNet50 architecture that was initially trained using over a million photos gathered from the ImageNet database.

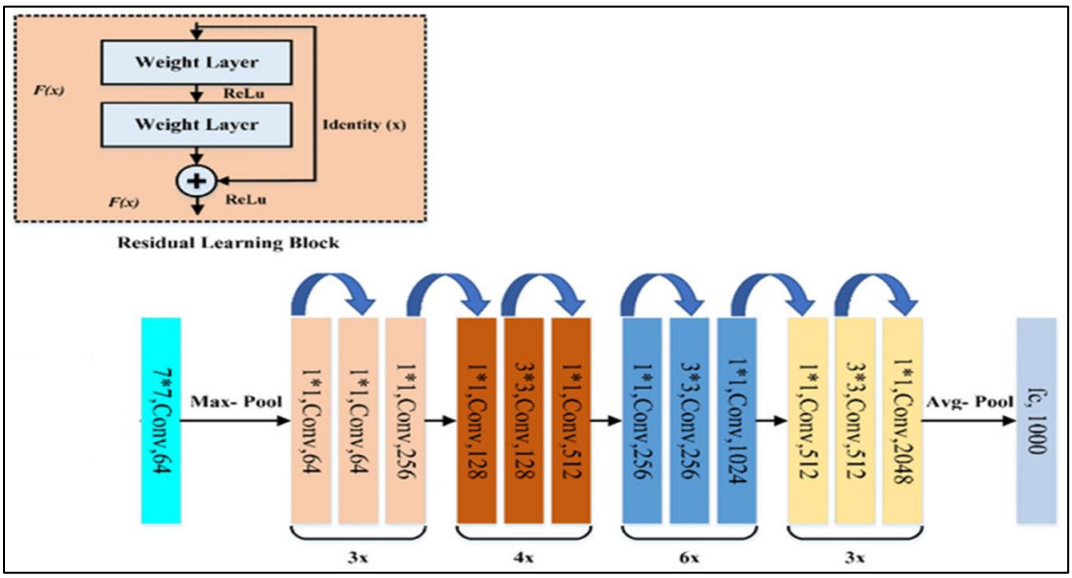


Figure 1.3.7: RESNet50 architecture [101]

1.3.12. EfficientNetB01 Model

EfficientNet is a member of the Convolutional Neural Network (CNN) family. Its fundamental architecture is built upon the mobile inverted bottleneck convolution (MBConv) module, which incorporates the concept of a squeeze and excitation network (SENet). The introduction of MBConv allows the model to possess adaptive depth, resulting in reduced training time and enhanced performance. In the case of EfficientNet-B1, it comprises 16 mobile flip bottleneck convolution modules, 2 convolution layers, 1 global average pooling layer, and 1 classification layer, as illustrated in Figure 3.3.8 [103].

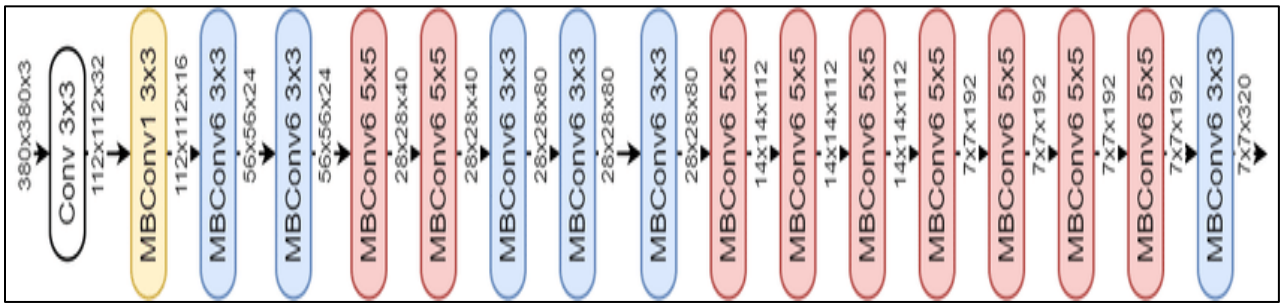


Figure 1.3.8: EfficientNetB01 Model Architecture [104]

EfficientNet emerged from the efforts of engineers at the Google Brain team [105]. Their central aim was not only too upscale DNNs but also to pioneer a fresh category termed EfficientNets. This

collection encompasses eight models, ranging from EfficientNet-B0 to EfficientNet-B7. Among them, EfficientNet-B0 stands as the baseline version, while the subsequent models, EfficientNet-B1 to EfficientNet-B7, represent scaled-up variants.

Historically, the conventional approach to improve the model accuracy involved constructing a base model with a fixed cost and then amplifying its capacity as more computational resources become available. Techniques for scaling up included altering the model's depth, width, or resolution. Another avenue was to boost image resolution, thereby enhancing accuracy, as depicted in Figure 3.3.9. However, rather than randomly adjusting width, depth, or resolution, a novel strategy emerged – uniform scaling of these dimensions through a set of scaling coefficients. This innovative strategy introduced a new lineage of models known as EfficientNets. This family exhibited improved accuracy and showcased a tenfold efficiency increase compared to its predecessors, all while being compact and faster. While independently scaling individual model dimensions yielded performance benefits, the key insight was that achieving a balanced distribution of network dimensions (width, depth, and image resolution) relative to available resources yielded the most substantial enhancement in model performance [105].

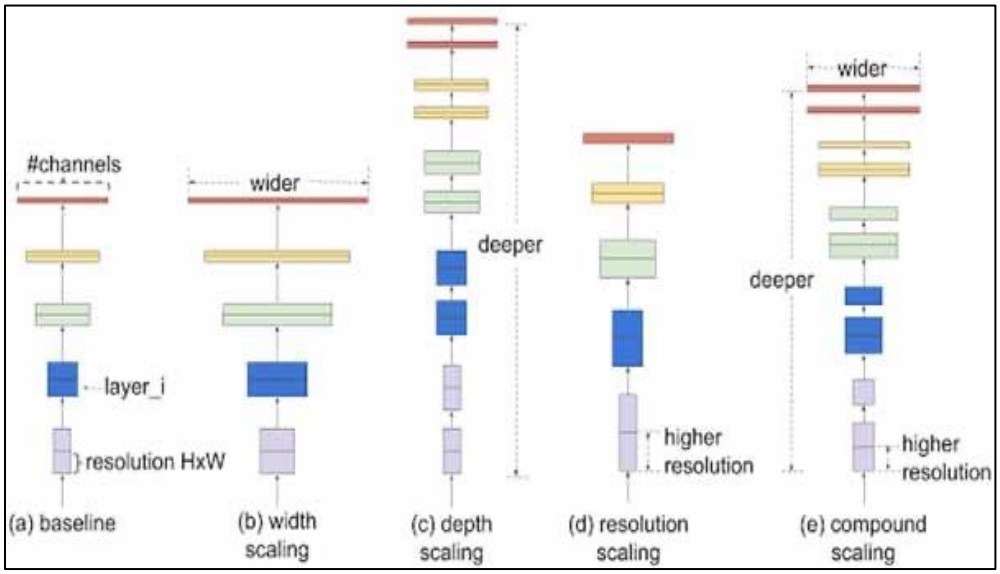


Figure 1.3.9 : . Different scaling methods. (b), (c), and (d) are the conventional methods with single-dimensional scaling [106]

The effectiveness of model scaling is intrinsically linked to the quality of the baseline network. Serving as the fundamental architecture for EfficientNets, EfficientNet-B0 was conceived through the application of neural architecture search within the AutoML MNAS framework. This approach yielded improvements in both efficiency and accuracy. The design is based on the concept of mobile inverted bottleneck convolution, which plays a pivotal role in shaping the resulting network structure. Table 3.3.1 provides a detailed breakdown of the architecture of EfficientNet-B0.

Table 1.3.1 : Architecture of EfficientNet-B0 [106].

Stage i	Operator F(i)	Resolution Hi x Wi	#Channels Ci	Layers Li
1	Conv3x3	224x224	32	1
2	MBConv1, k3x3	112x112	16	1
3	MBConv6, k3x3	112x112	24	2
4	MBConv6, k5x5	56x56	40	2
5	MBConv6, k3x3	28x28	80	3
6	MBConv6, k5x5	14x14	112	3
7	MBConv6, k5x5	14x14	192	4
8	MBConv6, k3x3	7x7	320	1
9	Conv 1x1 & pooling and FC	7x7	1280	1

1.3.13. Extreme Inception (Xception)

The term "depthwise separable convolutions" was introduced by Laurent Sifre in his PhD thesis [107]. In convolutional layers, network weights are shared, enabling simultaneous feature extraction and fusion. These layers employ an invariant function to spatially down sample the pooling layer, resulting in the generation of a large number of parameters [108]. In contrast, depthwise separable convolutions (also referred to as depthwise convolutions, distinct from spatially separable convolutions in image processing) consist of two consecutive and independent convolutional layers. The first layer is the depthwise convolution, which applies a single filter

across each channel (utilizing n different filters for an n-channel input) of the input data. The second layer is the pointwise convolution, involving a 1x1 convolution that projects the output of the depthwise convolution into a new channel space [109]. This process is depicted in Figure 3.3.11 below [110].

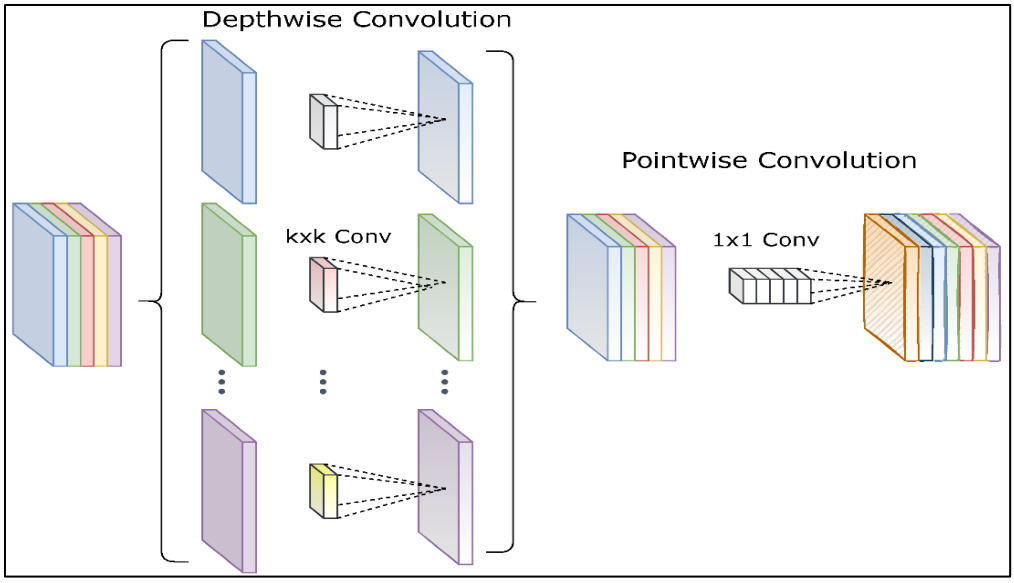


Figure 1.3.10 : Xception Model Architecture [111]

The process of factorization involved here leads to a significant reduction in both the size of the model and the computational expenses. The subsequent content provides the mathematical formulations for different convolutional processes: the convolution (Equation (3.23)), depthwise convolution (Equation (3.24) where ⊙ signifies the element-wise product), pointwise convolution (Equation (3.25)), and the operation of depthwise separable convolution (Equation (3.26)).

$$Conv(W,y)_{(i,j)} = \sum_{k,l,m}^{K,L,M} W_{(k,l,m)} \cdot y_{(k+i,l+j,\ m)} \tag{3.23}$$

$$\text{DepthwiseConv}(W,y)_{(i,j)} = \sum_{k,l}^{K,L} W_{(k,l)} \odot y_{(k+i,l+j)} \tag{3.24}$$

$$\text{PointwiseConv}(W,y)_{(i,j)} = \sum_{m}^{M} W_{(m)} \cdot y_{(i,j,m)} \tag{3.25}$$

$$DepthwiseSepConv(W_p, W_d, y)_{(i,j)} = PointwiseConv_{(i,j)}(W_p, DepthwiseConv_{(i,j)}(W_d, y)) \quad (3.26)$$

These mathematical expressions illustrate the operations of traditional convolution, depthwise convolution, pointwise convolution, and depthwise separable convolution, highlighting the factorization and reduction in computational complexity achieved through the depthwise separable convolution approach. Depthwise separable convolutions find practical applications in various deep learning models. Notably, the Xception model and the MobileNet family introduced in [112] showcase the effectiveness of this technique. The MobileNet architecture illustrates the advantages of depthwise separable convolutions in terms of model size reduction and computational efficiency.

1.3.14. MobileNet

The MobileNet architecture is founded on the concept of depthwise separable convolutions, which involves a factorization of the standard convolution process. This factorization splits a standard convolution into two separate operations: the depthwise convolution and the pointwise convolution. The depthwise convolution applies a single filter to each input channel, while the pointwise convolution employs a 1 × 1 convolution to combine the outputs of the previous depthwise convolution [113] Ordinarily, a standard convolution both filters and combines inputs simultaneously, but depthwise separable convolutions separate these actions into distinct layers. This separation yields a substantial reduction in both computation demands and model size.

The conventional convolutional layer processes an input feature map of dimensions DF × DF × M and generates an output feature map of size DF × DF × N. The convolution kernel K, which has dimensions DK × DK × M × N, controls this process. Here, DK represents the spatial width and height of a square input feature map, M represents the number of input channels (input depth), DG represents the spatial width and height of a square output feature map, and N represents the number

of output channels (output depth). Assuming stride one and padding, the output feature map for standard convolution is calculated as follows:

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \quad (3.27)$$

The computational cost of standard convolutions is:

$$DK \cdot DK \cdot M \cdot N \cdot DF \cdot DF \quad (3.28)$$

where the cost is multiplicatively dependent on the number of input channels M, the number of output channels N, the kernel size Dk, and the size of the feature map DF. MobileNet addresses computational efficiency by using depthwise separable convolutions. These convolutions consist of two layers: depthwise convolutions and pointwise convolutions. Depthwise convolutions filter input channels independently, and pointwise convolutions create linear combinations of their output. Depthwise convolution is highly efficient compared to standard convolution, but it doesn't combine channels to create new features. To accomplish this, an additional 1 × 1 convolution layer is introduced after depthwise convolution, generating a linear combination of its outputs. This combined process is known as depthwise separable convolution [114].

The computational cost of depthwise separable convolutions is the sum of the costs of depthwise and pointwise convolutions:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (3.29)$$

MobileNet employs 3 × 3 depthwise separable convolutions, resulting in 8 to 9 times lower computation than standard convolutions, with only a minor decrease in accuracy.

Further attempts at factorization, particularly in spatial dimensions, do not significantly reduce computation since the majority of computational load lies in depthwise convolutions.

1.3.15. Convolutional Neural Network (CNN)

The architectural design of convolutional neural networks is specifically tailored to effectively process picture data, which is characterized by its multidimensional array structure. The present concept possesses the capability to maintain the integrity of the initial data structure while also producing hierarchical representations. Figure 3.3.12 below shows a representative CNN that encompasses several sequential stages of processing layers, arranged in a left-to-right fashion. The CNN architecture typically comprises four distinct types of layers, namely a convolutional layer, a pooling layer, a fully connected layer, and a classification layer. The incorporation of convolutional and pooling layers is fundamental to the architectural framework, commonly observed in the initial stages. The convolutional layer leverages the inherent characteristics of natural signals, such as images, through the utilization of local connectivity, weight sharing, and pooling.

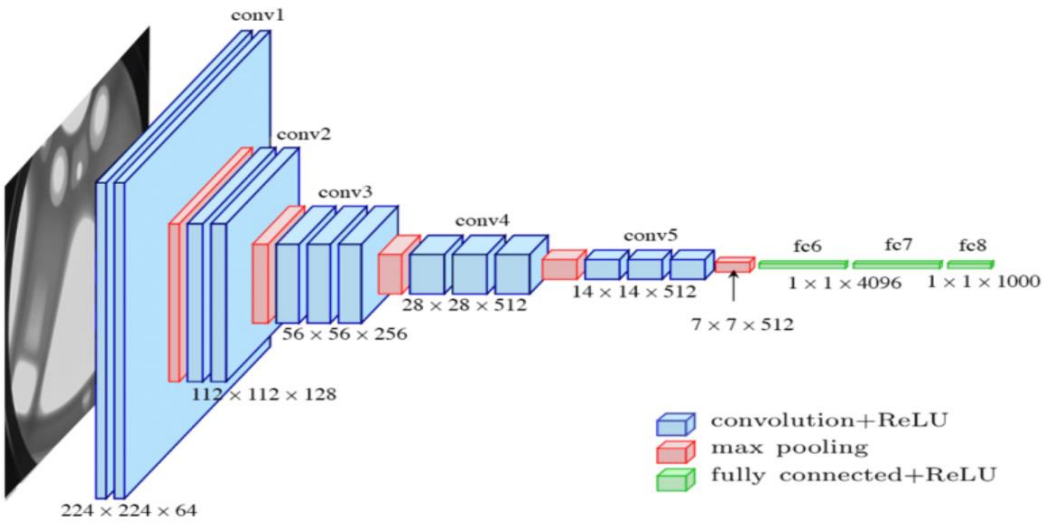


Figure 1.3.11 : An example of a convolution neural network architecture [115]

The following are the basic concepts of the CNN:

• Convolutional layer: is a fundamental component of convolutional neural networks (CNNs). It is structured into feature maps, which are created using the concept of local connectivity and weight sharing. The concept of local connectivity pertains to the arrangement where individual

neurons inside a feature map establish connections exclusively with nearby sections of the feature map in the preceding layer. These connections are facilitated by a collection of weights known as a filter bank [116], [117]. The weight-sharing paradigm is represented by the sharing of one filter bank among all units inside a feature map. In addition, distinct filter banks are employed for individual feature maps. The rationale for incorporating local connectivity and weight sharing in a model is to reduce the number of parameters required while making use of the strong correlation between neighboring pixels and the position invariance of local picture statistics [118]. The weighted total of each unit is subsequently inputted into a non-linear transformation function, commonly referred to as an activation function. The activation function facilitates the non-linear transformation of the data that is transmitted to the successive stages of processing.

• Pooling Layer: used for image recognition and computer vision tasks, where this layer executes a process of down-sampling to consolidate information with comparable semantic characteristics from the convolutional layer into a single representation. A pooling layer unit receives a local patch from a preceding feature map (convolutional layer) as its input and calculates the maximum or average value of the patch to produce its output. Consequently, the reduction in the dimensionality of the representation leads to a decrease in the number of parameters required in successive layers, enhancing the resilience of the representations through the establishment of invariance to minor shifts and distortions.

• Fully Connected Layer: is a type of neural network layer where each neuron is connected to every neuron in the previous layer. The units within this layer exhibit full connectivity with all units in the preceding layer, akin to the structure observed in a conventional neural network, specifically a multi-layer perceptron.

• Classification Layer: a component that is used to categorize or assign labels to data based on certain criteria or features. The function of this layer is to determine the method by which the prediction score is calculated, utilizing the outputs generated by preceding layers. The implementation of typical classification layers often involves the utilization of a softmax classifier or linear support vector machines [119]. The training algorithm for Convolutional Neural Networks (CNNs) follows the typical supervised training processes which involve the use of gradient descent and backpropagation.

1.4. Hyperparameters Tunning and Cross-Validation

1.4.1. Hyperparameters Tunning

Hyperparameters Tunning is crucial in building models where these hyperparameters control the configuration of the model [120]. By determining the optimal parameter value through experimentation with a predefined set of values, one can obtain the highest level of performance. The utilization of this strategy ensures significantly improved performance compared to models that are constructed utilizing randomly chosen parameter values [121]. The GridSearchCV is employed to fine-tune the parameters for the ML models while the DL models were optimized using the keras_tuner_cv package which is an extension for Keras Tuner to improve the performance and validation of DL models by employing cross-validation technique. Both are open-source tools designed for fine-tuning ML / DL models that facilitate the seamless and effective execution of hyperparameter tuning on models.

1.4.1.1. Machine Learning Models Tuning

Hyperparameter tuning involves adjusting the hyperparameters of ML models to optimize their performance and achieve the best level of accuracy using the GridSearch technique. Hyperparameters, which are parameters utilized to design the algorithm or determine the reduction of the loss function, were the focus of this optimization process. The method selects the

hyperparameters that yield the maximum performance by searching for predetermined values. This search ensures that the chosen hyperparameters result in a considerable performance improvement compared to randomly selected values [120].

The hyperparameters tunning of the DTC algorithm concerned optimizing the criterion, that is a feature used to degree the quality of splits, as well as the max_depth parameter, which determines the most intensity of the selection tree. Additionally, the min_samples_leaf parameter, which specifies the minimum quantity of samples required to form a leaf, and the min_samples_split parameter, which determines the minimal number of samples needed to perform a split, have been also optimized. [122]. The SVM employed the default RBF kernel to optimize the C parameter, which represents the regularization parameter, as well as the Gamma parameter, which represents the kernel coefficient [123]. In addition, the hyperparameters of the RF method were subjected to hyperparameter tuning. The criterion represents the function used to measure the quality of the split. The max_depth parameter determines the maximum depth allowed for the tree. The min_samples_leaf parameter specifies the minimum number of samples required to form a leaf node. The min_samples_split parameter determines the minimum number of samples needed to do a split. Lastly, the n_estimators parameter denotes the number of decision trees that will be constructed in the random forest [124].

While GBM model has the following parameters which include n_estimators that represent the number of boosting stages to perform, learning_rate that diminishes the impact of each tree proportionally by the specified rate, max_depth, and max_features [125]. Similar parameters were used for the light version of the model, XGB and Adaboost. The subsample and colsample_bytree determine the fraction of samples that will be utilized for training each base learner for the LGB [126].

1.4.1.2. Deep Learning Models Tuning

Deep learning models, a subset of artificial neural networks, excel in learning intricate patterns from data through a technique known as "deep learning." These models feature numerous hyperparameters, each demanding an adjustment to unlock their full potential. Key parameters such as the number of layers, number of neurons per layer, activation function, learning rate, and number of epochs are all significantly affected by DL The complexity of the model is affected by the number of layers and neurons. Meanwhile, the activation function plays a pivotal role in determining how the output of each neuron is computed. The learning rate, on the other hand, dictates the magnitude of steps taken during the gradient descent optimization process, while the number of epochs establishes the total iterations of training using the complete dataset. Although it presents a challenge, the task of identifying the optimal combination of hyperparameters for DL models is instrumental in achieving high accuracy for specific tasks.

1.4.2. Cross-Validation

The K-fold cross-validation approach is commonly employed in both classification and regression modeling. The K-fold cross-validation approach is commonly employed as an evaluation technique for classification modeling to minimize bias between the full dataset and the training set or test set. The methodology involves partitioning the original dataset into K distinct subsets. As demonstrated by Figure 3.4.1 a selection of one subset as the test set, while the remaining K - 1 subsets are designated as the training set. Next, the classification accuracy of each classifier is quantified as the mean value derived from the K models acquired during the training phase. Hence, the use of the K-fold cross-validation technique may successfully mitigate the inherent variability in dataset selection and provide a rigorous assessment of the classification model's dependability [127] [128]. In this thesis, 10-fold cross-validation is used for ML models while 5-fold is used for DL models. Both tuning techniques employed in this thesis including the GridSearchCV and keras_tuner_cv, introduce cross-validation methodologies that assess model performance across

different data subsets, enhancing robustness and generalization. Together, these techniques contribute to more efficient, reliable, and generalized deep learning models, furthering their applicability in real-world scenarios.

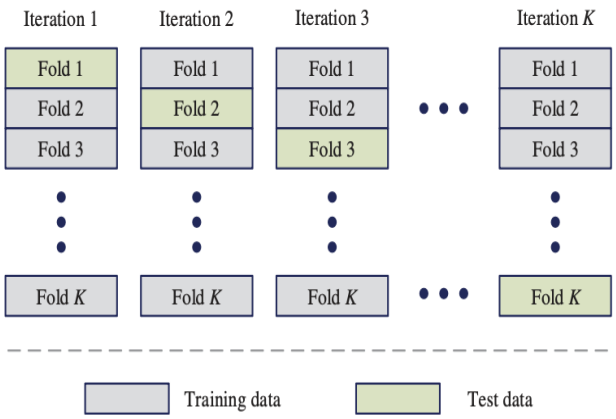


Figure 1.4.1 : K Cross-Validation Method [127]

1.5. Performance Metrics Selection

After the modeling stage where the ML/ DL model is created and trained, the next step is to evaluate the performance of that model using various performance metrics. These metrics reflect the effectiveness of the lung cancer classification system based on the predicted and actual values. Several methods can be used to evaluate the model's performance, including accuracy, precision, recall (sensitivity), confusion matrix, and area under the curve "AUC" [129].

- **Accuracy:** Accuracy is a widely used performance metric that measures the model's ability to correctly classify test samples into their respective classes. It calculates the ratio of correctly predicted instances to the total number of represented structures. Mathematically, accuracy can be expressed as the sum of true positives (TP) and true negatives (TN) divided by the sum of true positives, false positives (FP), true negatives, and false negatives (FN):

$$\text{Accuracy} = \frac{Correct\ Expectations}{Total\ Expectations} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \qquad (3.30)$$

However, it's important to note that accuracy has limitations, particularly when dealing with imbalanced datasets where the distribution of samples across categories is uneven. In such cases, accuracy may not provide a comprehensive evaluation of the model's performance. Precision, on the other hand, considers all correctly classified instances (TP + TN) and calculates the ratio of these instances to the entire dataset [2]. While accuracy describes the overall number of correct classifications across the entire dataset, it may overlook the balance between false positive and false negative rates. If the false positives and false negatives differ, two networks with different true positives and true negatives can yield identical accuracy scores, potentially misleading the assessment of class outcomes [2]. So other performance metrics can be used, such as the confusion matrix and the area under the curve (AUC) [2]. These metrics help gain a better understanding of the model performance by considering true positives, true negatives, false positives, and false negatives.

- **Confusion Matrix:** The confusion matrix is a valuable tool for evaluating the accuracy and performance of a model. It provides a tabular representation shown in Figure 3.5.1 that allows for a comprehensive analysis of the algorithm's predictions, also known as an error matrix. The matrix's rows represent the predicted class labels, while the columns represent the actual class labels, or vice versa. Its name stems from its ability to easily identify instances where the system confuses or misclassifies the two groups. The confusion matrix is particularly useful in the evaluation of multi-class classification problems. It consists of four essential parameters:

    - True Positive (TP): The model correctly predicts a positive example as positive.
    - True Negative (TN): The model correctly predicts a negative example as negative.

- False Positive (FP): The model mistakenly predicts a negative example as positive.
- False Negative (FN): The model mistakenly predicts a positive example as negative.

To be considered effective, a model must achieve accurate predictions, minimize errors, and reduce both false positives and false negatives. Ideally, the true positives (TP) and true negatives (TN) should be high, while the false positives (FP) and false negatives (FN) should be low. By analyzing the values within the confusion matrix, one can gain insights into the model's performance, identify areas of improvement, and determine its ability to correctly classify instances across different classes [130].

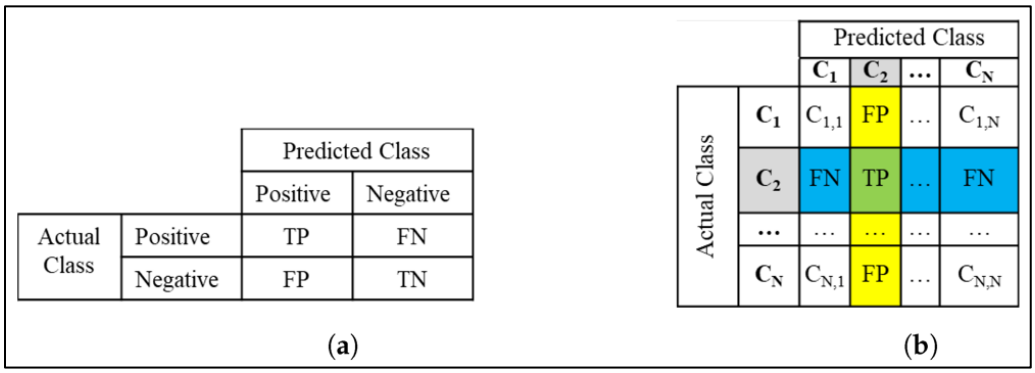


Figure 1.5.1: Confusion Matrix Illustration (a) Binary Classification CM (b) Multiclassification CM

[131]

- **Sensitivity or Recall:** Sensitivity is a performance metric that assesses the system's ability to accurately diagnose lung cancer. It specifically measures the proportion of positive instances that are correctly detected. The calculation of sensitivity involves determining the ratio of true positives to the sum of true positives and false negatives.

$$Sensitivity = \frac{TP}{TP+FN} \tag{3.31}$$

• **Specificity:** Specificity, also referred to as the True Negative Rate (TNR), is a performance metric that assesses the model's ability to correctly identify cases without lung cancer in the context of lung cancer diagnosis. It measures the proportion of true negatives, indicating the instances where the model accurately classifies individuals as not having lung cancer. The specificity of the model is computed by dividing the number of true negatives by the sum of true negatives and false

positives, providing insights into its effectiveness in correctly identifying negative cases of lung cancer.

$$Specificity = \frac{TN}{TN+FP} \tag{3.32}$$

- **Precision:** A model's precision is a performance metric that indicates how accurately it can identify tumor images in the context of tumor identification. To calculate this number, it divides the number of true positives (TP) by the sum of true positives and false positives (TP + FP), which includes both correctly identified and incorrectly identified tumor images. Higher precision values indicate a lower rate of false positives, meaning that the model is more precise in correctly identifying tumor images. When the accuracy rate is higher, the model becomes more reliable and useful in accurately identifying tumors. Therefore, maximizing precision is crucial in developing an effective model for tumor identification.

$$Precision = \frac{TP}{TP+FP} \tag{3.33}$$

- **F-Score** serves as a metric for assessing test accuracy by capturing the harmonic mean of Precision and Recall. It offers a measure of performance, with a value of 1 signifying perfect accuracy and recall (100%), and a value of 0 denoting the poorest possible performance. The F-Score is defined as follows:

$$\mathrm{F-Score} = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{3.34}$$

• **Receiver Operating Characteristic Curve (ROC) and Area under the Curve (AUC):** This graphical representation illustrates the relationship between True Positive Rate (TPR), also known as recall, and False Positive Rate (FPR). The ROC curve is constructed by plotting TPR and FPR at various classification thresholds to distinguish between positive and negative classes. The top-left corner of the ROC curve represents the optimal scenario, where 100% of positive values are correctly classified (TPR = 1) and 0% of positive values are incorrectly predicted as negative (FPR

= 0). The goal is to maximize TPR while minimizing FPR, making a larger Area under the ROC Curve (AUC) desirable. In Figure 3.5.2, "TPR" is on the y-axis, and "False Positive Rate (FPR)" is on the x-axis.

$$TPR = \frac{TP}{TP+FN} \tag{3.35}$$

$$FPR = \frac{FP}{FP+TN} \tag{3.36}$$

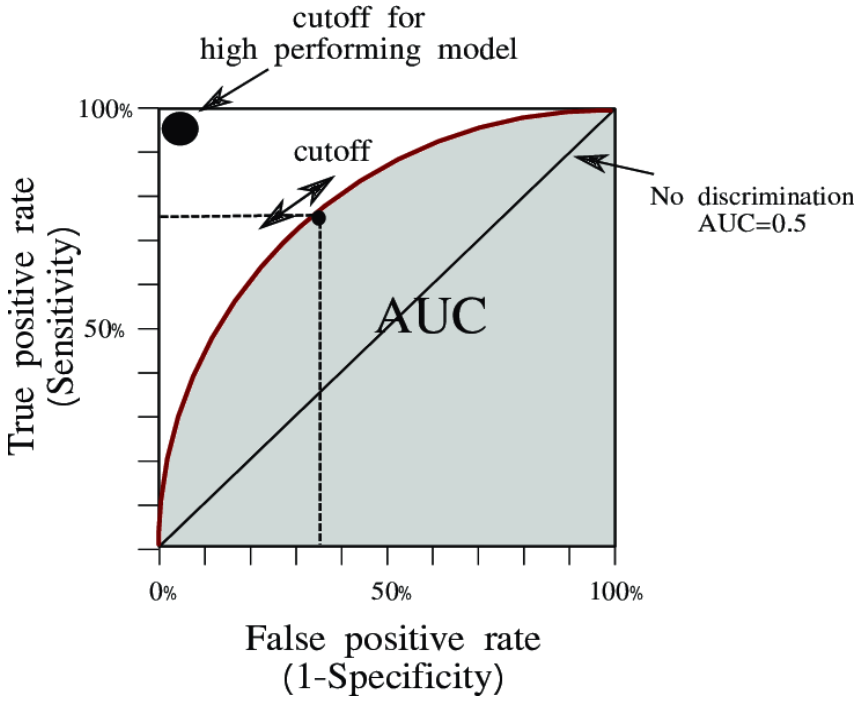


Figure 1.5.2: AUC and ROC illustration. [132]

Chapter 4
Experiments and Results

4.1 Experiments and Results

In this chapter, after presenting both the DL models and ML models in the previous chapter, those models will be applied to both the CT images dataset and the clinical/demographic dataset. The experiment is divided into two phases. The first stage includes applying the ML models to the clinical/demographic dataset. The next phase demonstrates applying DL models on the image's dataset. In both phases performance metrics will be applied including the accuracy, model's sensitivity, and the area under the curve (AUC) curves using receiver operating curves (ROC). Based on the results of the previous metrics, the best model will be determined.

4.2 Computing Environment

The experiments and calculations for this research were carried out on a MacBook Pro equipped with the Apple M1 Pro chip, featuring a 10-core CPU, 16-core GPU, 16GB Unified Memory, and 512GB SSD Storage. Apple M1 chip with a Pro GPU is good for training ML / DL models due to its significantly enhanced computational speed and superior processing capabilities in comparison to conventional CPUs. The utilization of a GPU allowed to train these models in a fraction of the time it would have taken with a CPU, facilitating the execution of more experiments, and enabling more efficient model refinement. In this thesis, the Python programming language, implemented within the Jupyter Notebook environment, forms the foundation of the computational framework. Several essential Python open-source libraries, including Pandas, Scikit-learn, Keras, TensorFlow, Matplotlib, GridSearchCV, and KerasTunerCV, have been employed to support various aspects of the research and analysis.

4.3 Machine Learning Practical Experiments

This section delves into the results and analysis of the applied ML models, by examining the outcomes and insights gleaned from ML-based approaches. This exploration showcases the effectiveness of various ML algorithms, their hyperparameter configurations, and the implications of their performance on this research objectives.

4.3.1 Support Vector Machine Results

Table 4.3.1 that follows provides a comparison of the results of the SVM model before and after tuning. Additionally, Table 4.3.2 displays the parameters that were tuned along with the values that were determined to be optimal. In addition, the final tuned model's Confusion Matrix and ROC curve shown in Figure 4.3.1 are displayed, and the model's accuracy was determined to be 96.29%.

Table 4.3.1 : SVM Metrics Before and After Tunning

	Accuracy	**Precision**	**Recall**	**F1-Score**
Before tunning	93.51%	96.00%	92.00%	94.00%
After tunning	96.29%	96.00%	96.00%	96.00%

Table 4.3.2 : SVM Optimal hyperparameter combination post hyper-tuning.

Hyperparameter Name	**Hyperparameter Values**	**Hyperparameters Optimal Value**
C	0.01,0.1, 1, 10	1
kernel	'linear', 'poly', 'rbf'	linear
degree	1, 2, 3, 4	1

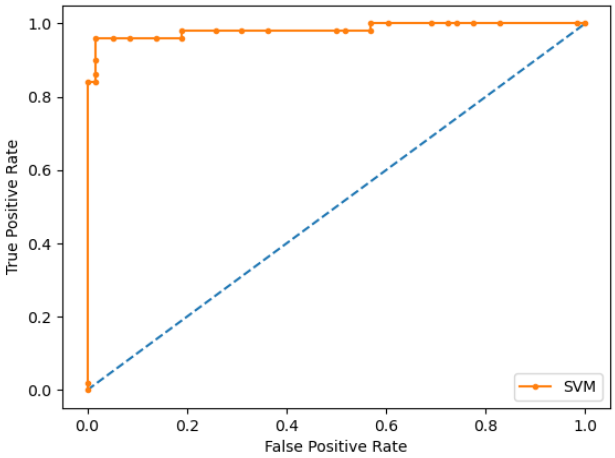


Figure 4.3.1 : SVM ROC Curve

4.3.2 Light Gradient Boosting Results

Table 4.3.3 provided below offers a comparative analysis of the LGB model's performance before and after tuning. Additionally, Table 4.3.4 presents the optimized hyperparameters with their corresponding optimal values. The final tuned model's evaluation includes a Confusion Matrix in Figure 4.3.2 and an ROC curve in Figure 4.3.3, revealing an impressive accuracy rate of 97.22%.

Table 4.3.3 : LGB Metrics Before and After Tunning

	Accuracy	**Precision**	**Recall**	**F1-Score**
Before tunning	97.22%	98.00%	96.00%	97.00%
After tunning	97.22%	98.00%	96.00%	97.00%

Table 4.3.4 : LGB Optimal hyperparameter combination post hyper-tuning

Hyperparameter Name	**Hyperparameter Values**	**Hyperparameters Optimal Value**
max_depth	[3,5,7]	7
learning_rate	0.001, 0.01, 0.1, 1.0	0.1
n_estimators	100, 200, 300	200
Subsample	0.8, 0.9, 1.0	0.8
colsample_bytree	0.8, 0.9, 1.0	0.9

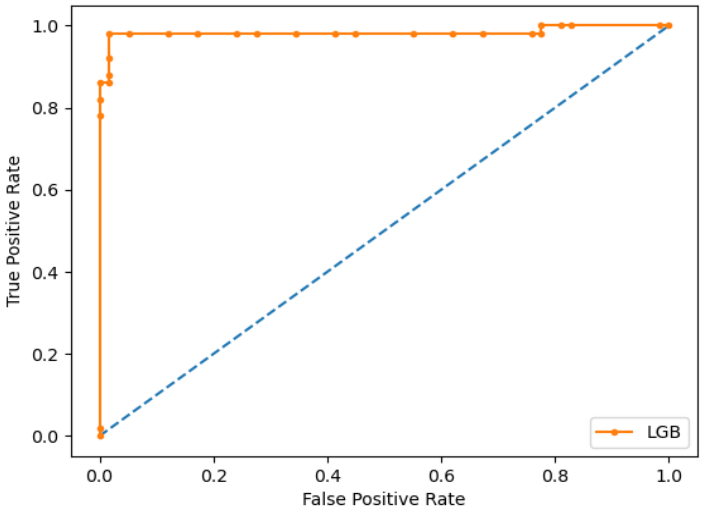


Figure 4.3.2 : LGB ROC Curve.

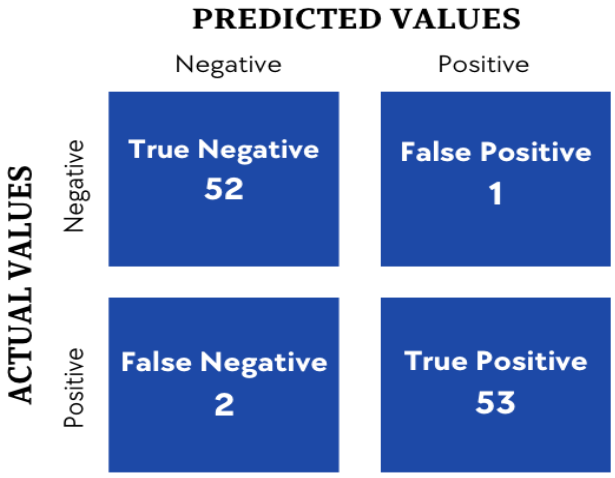


Figure 4.3.3 : LGB Confusion Matrix.

4.3.3 Extreme Gradient Boosting Results

Table 4.3.5 below compares the results of the XGB model before and after tuning, as well as displaying the tuned parameters with their optimal values in Table 4.3.6. The final tuned model's Confusion Matrix, as shown in Figure 4.3.4, and ROC curve, demonstrated in Figure 4.3.5, are also displayed., with a final accuracy of 97.22%.

Table 4.3.5 : XGB Metrics Before and After Tunning

	Accuracy	Precision	Recall	F1-Score
Before tunning	95.37%	98.00%	96.00%	97.00%
After tunning	97.22%	98.00%	96.00%	97.00%

Table 4.3.6 : XGB Optimal hyperparameter combination post hyper-tuning.:

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
max_depth	[3,5,7]	5
learning_rate	0.001, 0.01, 0.1, 1.0	0.1
n_estimators	100, 200, 300	200
subsample	0.8, 0.9, 1.0	0.9
colsample_bytree	0.8, 0.9, 1.0	1.0

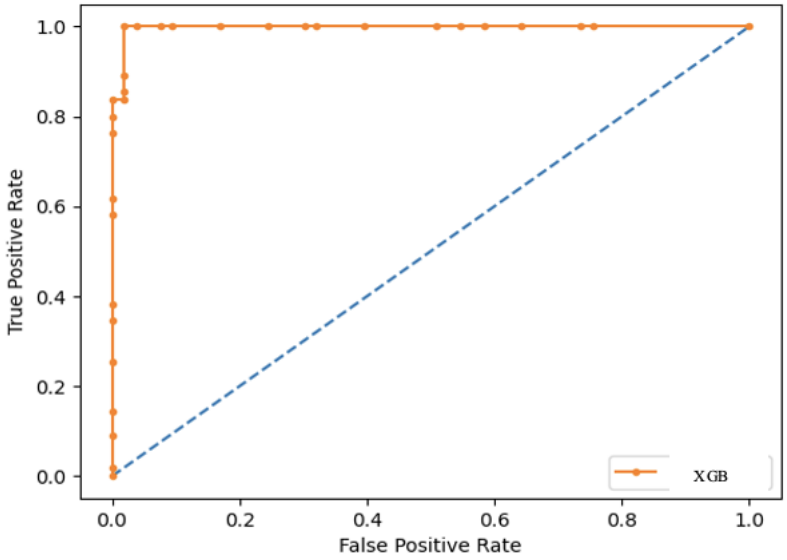


Figure 4.3.4 : XGB ROC Curve.

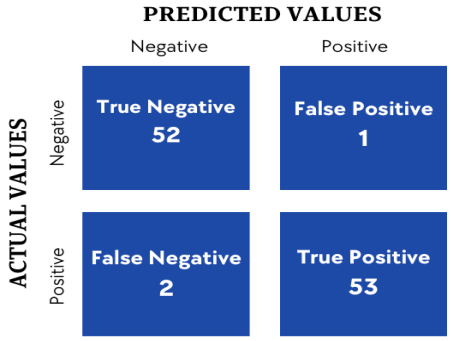


Figure 4.3.5 : XGB Confusion Matrix.

4.3.4 Decision Tree Results

The following Table 4.3.7 presents a comparative analysis of the outcomes obtained from the DT model before and after tuning. Additionally, Table 4.3.8 displays the tuned parameters together with their corresponding optimal values. The optimized final model is accompanied by the presentation of its confusion matrix in Figure 4.3.6 and ROC curve in Figure 4.3.7., demonstrating an accuracy of 98.14%.

Table 4.3.7: DT Metrics Before and After Tunning

	Accuracy	Precision	Recall	F1-Score
Before tunning	95.37%	98.00%	96.00%	97.00%
After tunning	98.14%	98.00%	93.00%	95.00%

Table 4.3.8: DT Optimal hyperparameter combination post hyper-tuning

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
max_depth	None, 100, 200, 300	20
Min_samples_split	2,5,10	2
Min_samples_leaf	1,2,4	4
criterion	'gini', 'entropy'	gini

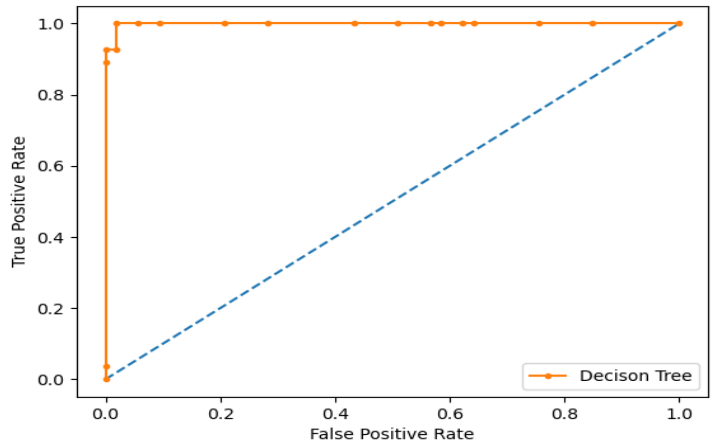


Figure 4.3.6 : DT ROC Curve.

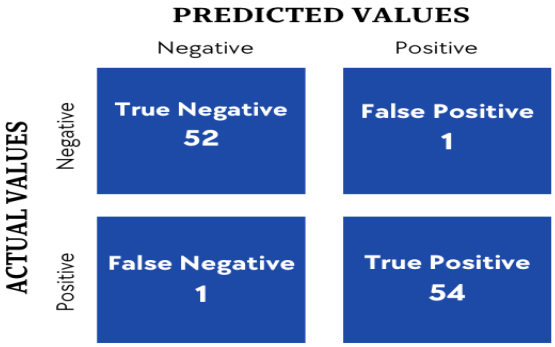


Figure 4.3.7 : DT Confusion Matrix.

4.3.5 Random Forest Results

Table 4.3.9 below shows a comparison between the results on the RF model before and after tunning, besides showing the tunned parameters with the optimal selected values in Table 4.3.10. The final tuned model is also presented with its Confusion Matrix in Figure 4.3.8 and ROC curve in Figure 4.3.9, achieving a final accuracy of 98.14%

Table 4.3.9: RF Metrics Before and After Tunning

	Accuracy	Precision	Recall	F1-Score
Before tunning	97.22%	98.00%	96.00%	97.00%
After tunning	98.14%	98.00%	98.00%	98.00%

Table 4.3.10: RF Optimal hyperparameter combination post hyper-tuning

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
n_estimators	[10, 50, 100, 200]	10
max_depth	None, 100, 200, 300	100
Min_samples_split	2,5,10	5
Min_samples_leaf	1,2,4	2
criterion	'gini', 'entropy'	entropy

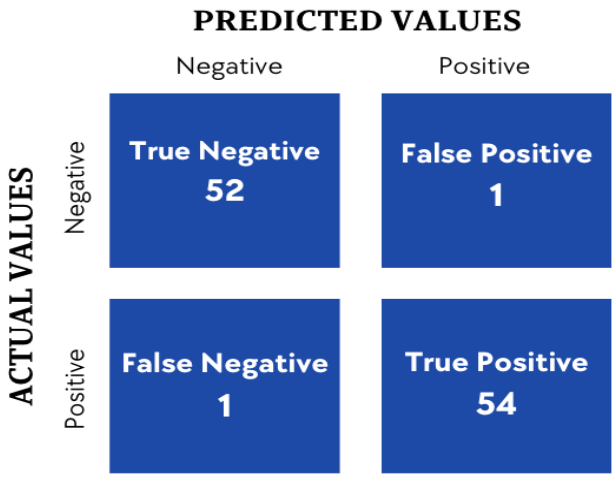


Figure 4.3.8 : RF Confusion Matrix.

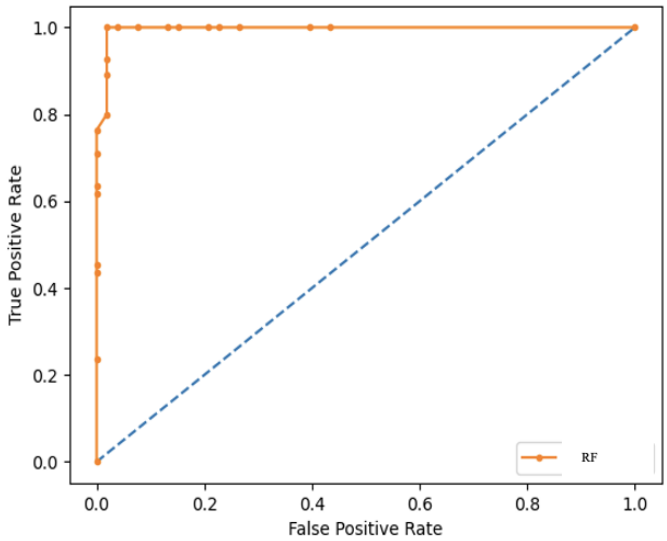


Figure 4.3.9 : RF ROC Curve.

4.3.6 Adaboost Results

The next Table 4.3.11 presents a comparative assessment of the findings obtained from the AdaBoost model before and after tuning. Additionally, the Table 4.3.12 displays the tuned parameters together with their corresponding optimal values. The confusion matrix in Figure 4.3.10 and ROC curve in Figure 4.3.11 are presented for the optimized final model, demonstrating an achieved accuracy of 99.07%, which outperforms both DT and RF.

Table 4.3.11 : Adaboost Metrics Before and After Tunning

	Accuracy	**Precision**	**Recall**	**F1-Score**
Before tunning	95.37%	98.00%	96.00%	97.00%
After tunning	99.07%	98.00%	100.00%	99.00%

Table 4.3.12 : Adaboost Optimal hyperparameter combination post hyper-tuning.

Hyperparameter Name	**Hyperparameter Values**	**Hyperparameters Optimal Value**
n_estimators	[50, 100]	100
learning_rate	0.001, 0.01, 0.1, 1.0	1.0

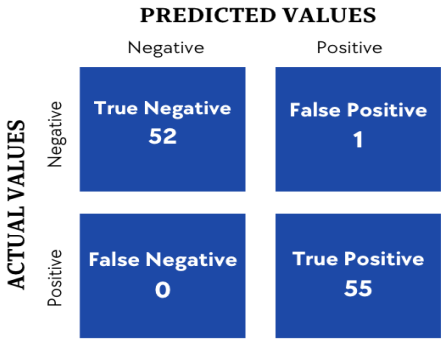


Figure 4.310. : Adaboost Confusion Matrix.

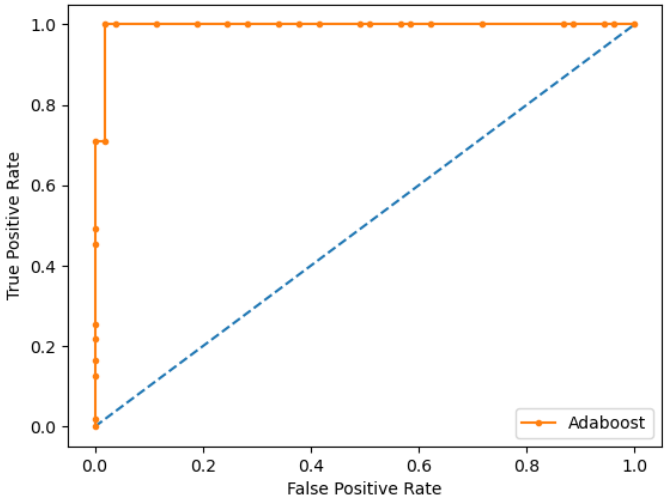


Figure 4.3.11 : Adaboost ROC Curve.

4.3.7 Gradient Boosting Results

In the context of a GB model, the following Table 4.3.13 illustrates a side-by-side comparison of its performance before and after tuning. It also showcases the finely tuned hyperparameters with their respective optimal values in Table 4.3.14. The evaluation of the ultimate tuned model incorporates both a Confusion Matrix shown in Figure 4.3.13 and a ROC curve shown in Figure 4.3.12, culminating in a remarkable accuracy score of 99.07%

Table 4.3.13 : GBM Metrics Before and After Tunning

	Accuracy	Precision	Recall	F1-Score
Before tunning	95.37%	98.00%	93.00%	95.00%
After tunning	99.07%	98.00%	100.00%	99.00%

Table 4.3.14 : GBM Optimal hyperparameter combination post hyper-tuning

Hyperparameter Name	**Hyperparameter Values**	**Hyperparameters Optimal Value**
n_estimators	[40,30,50]	30
learning_rate	0.001, 0.01, 0.1, 1.0	0.1
max_depth	3,4,5	3
max_features	Sqrt, log2	Sqrt

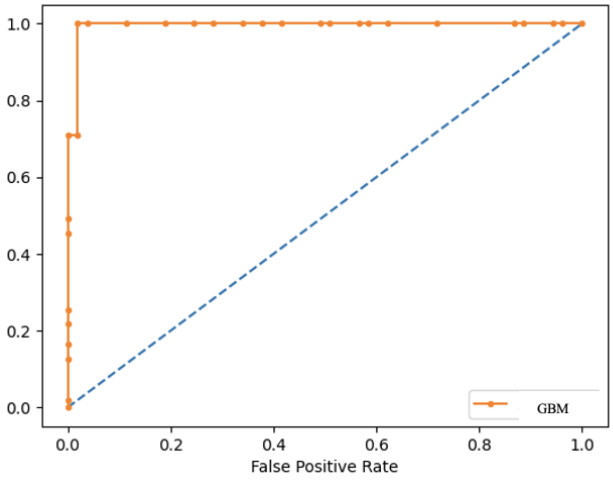


Figure 4.3.12 : GBM ROC Curve.

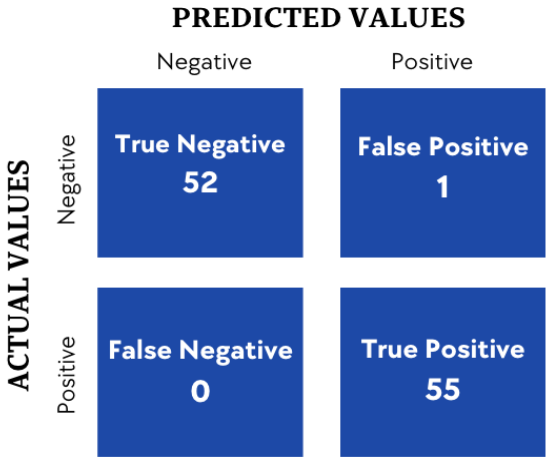


Figure 4.3.13 : GBM Confusion Matrix.

To summarize the findings, Table 4.3.15 and Figure 4.3.14 combined the results from all models, providing a concise overview of their performance metrics.

Table 4.3.15 : Classification results for different model on the same dataset

Model Type	Accuracy	Precision	Recall	F1-Score
Adaboost	***99.07%***	98.00%	100.00%	99.00%
GBM	***99.07%***	98.00%	100.00%	99.00%
RF	98.14%	98.00%	98.00%	98.00%
DT	98.14%	98.00%	98.00%	98.00%
XGB	97.22%	98.00%	96.00%	97.00%
LGB	97.22%	98.00%	96.00%	97.00%
SVM	96.29%	96.00%	96.00%	96.00%

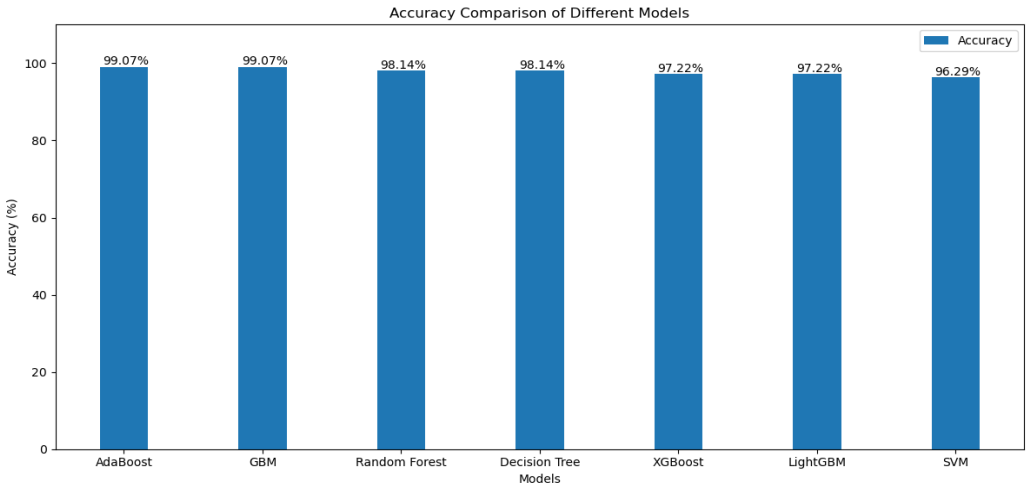


Figure 4.3.14 : Chart of Comparison of overall accuracy for all Classification Models.

The classification results represent the performance of various machine learning models on the dataset, showcasing their abilities in capturing patterns and making accurate predictions. AdaBoost and Gradient Boosting Machine (GBM) achieved the highest accuracy of 99.07%. These ensemble methods excel due to their sequential learning process. AdaBoost focuses on samples that previous models misclassified, adjusting their weights to improve accuracy. GBM builds decision trees sequentially, emphasizing the correction of errors made by prior trees. This iterative approach allows both AdaBoost and GBM to capture intricate relationships in the data, resulting in top accuracy. DT and RFalso performed remarkably well, each yielding an accuracy of 98.14% due to their ability to model complex feature interactions. RF, as an ensemble of Decision Trees, further enhances performance by averaging predictions and reducing overfitting, contributing to impressive accuracy.

XGBoost's strength lies in its efficiency and predictive power. It builds trees in a manner that minimizes loss functions, making it a versatile choice for a wide range of datasets. LGB, another gradient-boosting framework, mirrored XGBoost's performance with an accuracy of 97.22%.

SVM, though slightly behind with an accuracy of 96.29%. In summary, the ensemble methods (AdaBoost and GBM) and gradient boosting algorithms (XGBoost and LGB) demonstrated remarkable capabilities in capturing intricate data patterns.

The variable importance plot, which presents the most significant feature based on a mean decrease in Gini, was used to assess the importance of input features to the model. This finding suggests that the top variable has greater classification power and an impact on the model outcomes than the bottom variable. Figure 4.3.15 demonstrates how key factors in determining lung cancer include allergy, swallowing difficulty, alcohol consumption, age, coughing, chronic disease, etc. But at the same time trying to drop one of the least important features has affected the model performance negatively which implies the importance of these features even its low.

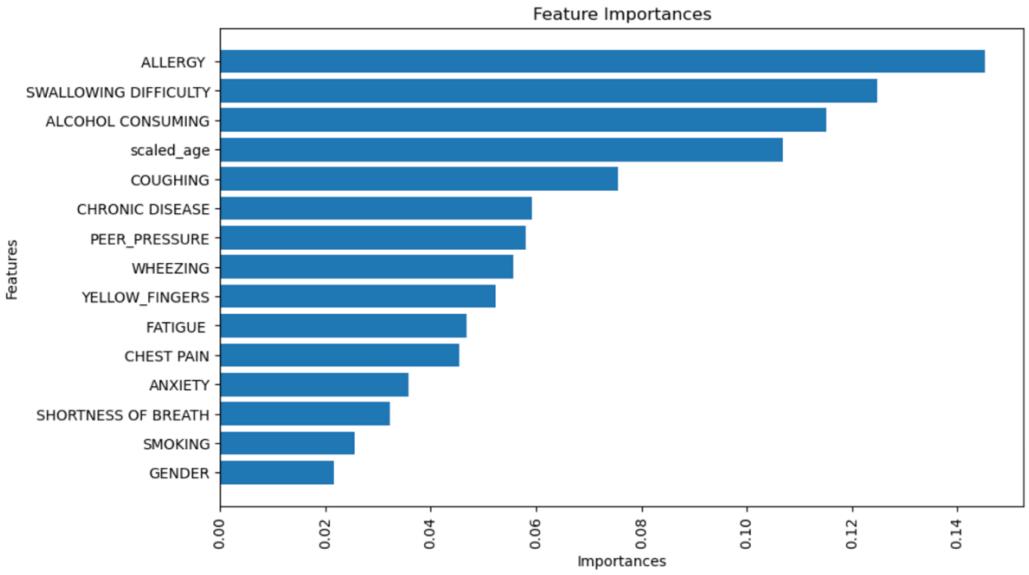


Figure 4.3.15 : Features Importance's Plot

The dataset used in this experiment is the same one used in [133] with a similar methodology. As part of the evaluation process for the machine learning models, a comparative analysis was performed, taking into consideration the findings presented in this research and the proposed model. Table 4.3.16 presents a summary of the accuracy achieved by each model in the context of the relevant studies.

Table 4.3.16 : Comparison of model performance between the proposed models and the models of the Published Paper [133]

Model Type	Accuracy	Accuracy on Published Paper [133]
Adaboost	***99.07%***	-
GBM	***99.07%***	98.76%
RF	98.14%	-
DT	98.14%	-
XGB	97.22%	96.27%
LGB	97.22%	96.89%
SVM	96.29%	-

Nevertheless, in the proposed model, systematically optimized ML techniques were employed, resulting in significant enhancements in the classification accuracy, as illustrated in Table 4.3.16 provided. In this context, image resizing has been performed using downscaling, which may introduce slight distortion. However, this distortion has been addressed through subsequent enhancement techniques such as denoising and the CLAHE filter. It's important to note that downscaling is preferred over the cropping approach, as cropping could result in the loss of essential elements and features. As well as replacing the Random Oversampling techniques with SMOTE techniques which helps in the improvement of performance as well. SMOTE appears to outperform random oversampling due to its ability to create synthetic minority-class samples that better reflect the underlying distribution. Unlike random oversampling, which simply duplicates existing samples, SMOTE generates diverse synthetic instances, aiding in model generalization and reducing overfitting. By strategically interpolating in sparse feature regions, SMOTE enhances class separability and data quality. Moreover, it mitigates data leakage risks and is particularly effective for small, imbalanced classes.

4.4 Deep Learning Practical Experiments

To provide a comprehensive analysis of the model's performance and its implications for this research objectives, this section, present and discusses the results of the DL experiments for various DL models applied to the CT images dataset. The experiment was carried out using a Jupyter Notebook, where Python code was implemented with the support of the Keras package. The dataset utilized in this research was divided into two separate subsets: a testing set (20%) and a training set (80%). The dimensions of the input for the model were configured to be 244 × 224 pixels images. The different CNN models, are employed as a convolutional base, provides a robust foundation for feature extraction from lung cancer image dataset. Furthermore, all layers in the model are set to be not trainable to prevent the retraining of the convolutional base layers and retain the pre-trained knowledge. A custom classifier is built on top of the CNN base as shown in Figure 4.4.1. This classifier comprises the following layers, flatten Layer, to transform the output of the convolutional base into a one-dimensional feature vector, Dense Layer which is a fully connected layer with certain number of units and ReLU activation to capture high-level features. The introduction of weight decay here helps in preventing overfitting, guiding the model to favor more parsimonious and robust representations, Dropout Layer to mitigate overfitting. Output Layer which is the final layer with a Softmax activation function for multi-class classification, with the number of units matching the number of classes in the used dataset (3). To explore the optimal hyperparameters, a hyperparameter tuning approach using the Keras Tuner library is employed including the number of units in the dense layer, learning rate, weight decay, dropout, batch size, and number of epochs. Each model with the best hyperparameters was compiled with the Adam optimizer and sparse categorical cross-entropy as the loss function to facilitate efficient training.

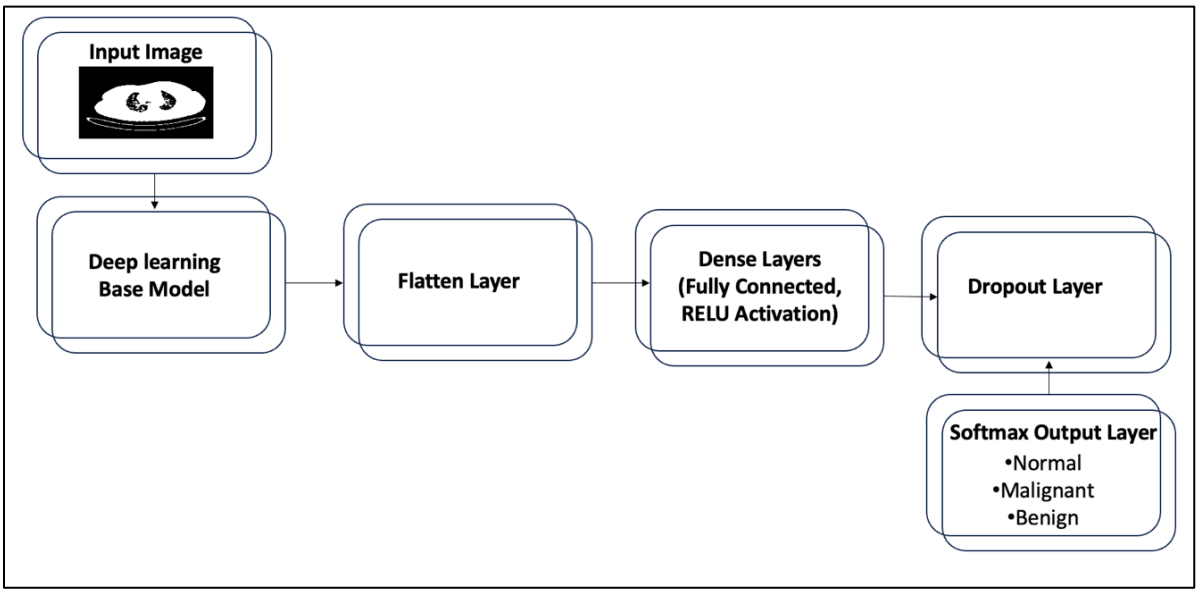


Figure 4.4.1 : DL Models Architecture.

4.4.1 DenseNet121 Results

In this part, the DenseNet121 architecture, a convolutional neural network characterized by its 120 convolutions, was employed. This model had undergone pre-training on an extensive database comprising over one million images. Leveraging DenseNet's capacity to work with fewer parameters and facilitate feature reuse. As is consistent with previous modeling approaches, the model was trained using the best-selected hyperparameters in Table 4.4.1 for 20 epochs as shown in Figure 4.4.2.

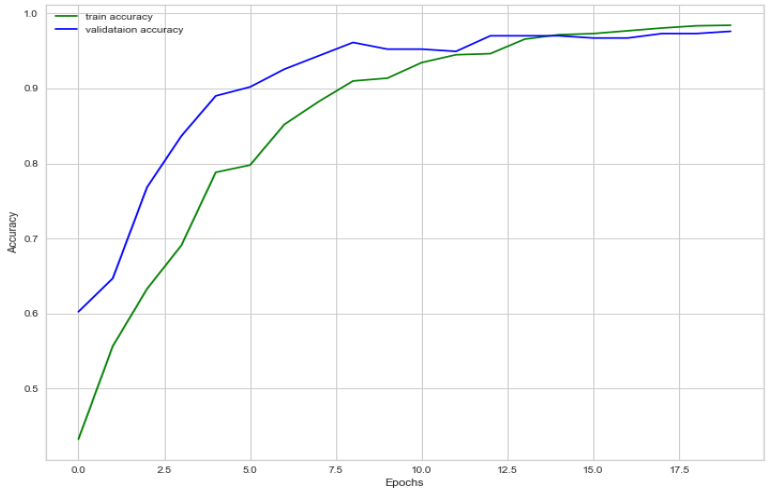


Figure 4.4.2 : Performance learning curves for training & validation accuracy DenseNet121

Table 4.4.1 : DenseNet121 tunned hyperparameters with optimal values

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0
units	[32, 64, ….512]	480
dropout	[0.2,0.3,0.4,0.5]	0.3
Batch_size	[100,110,120….150]	150
Num_epochs	[10,20]	20

The results obtained from the DenseNet model in Table 4.4.2 were notably consistent with those observed in the Xception model. Both models exhibited remarkable performance in the task of tumor classification. In particular, the DenseNet model, characterized by its 120 convolutional layers and pre-training on an extensive image database, demonstrated a high testing accuracy of 97.3%, which mirrored the exceptional accuracy seen in the Xception model. These parallel outcomes across different architectures underscore the robustness and reliability of the models in accurately classifying tumors, reaffirming their effectiveness in medical image analysis. This is confirmed by the confusion matrix in Figure 4.4.3 and ROC curve in Figure 4.4.4.

Table 4.4.2 : Classification Report for DenseNet121

	Accuracy	Class	Precision	Recall	F1-Score
DenseNet121	97.32%	**Benign (0)**	98.00%	98.00%	98.00%
		Malignant (1)	98.00%	98.00%	98.00%
		Normal (2)	96.00%	96.00%	96.00%

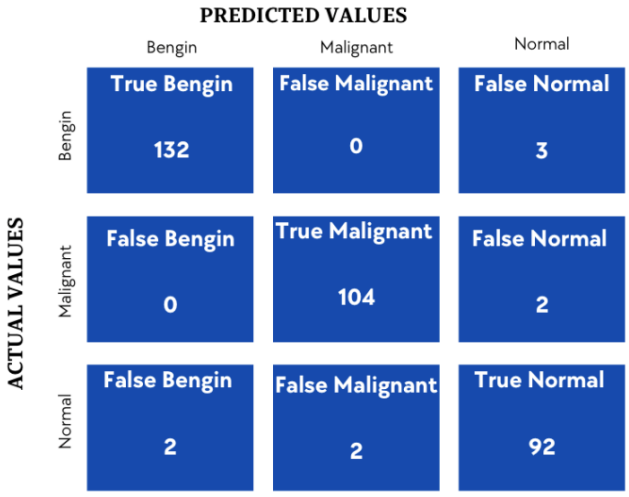


Figure 4.4.3 : Confusion Matrix for DenseNet121

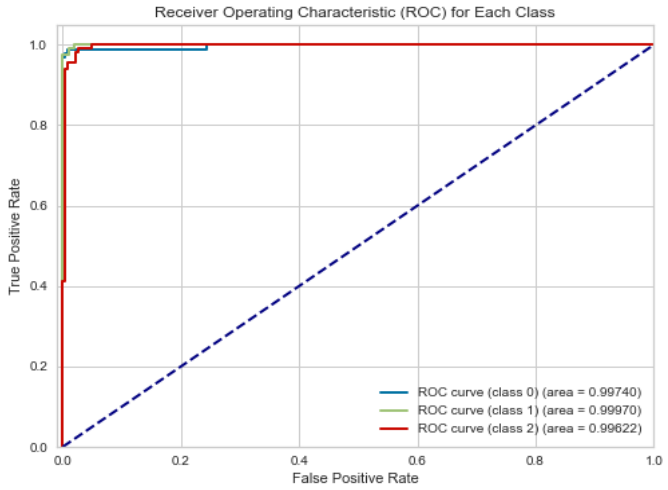


Figure 4.4.4 : Receiver operating characteristic (ROC) to Multi-Class Validation Set DenseNet121

4.4.2 MobileNet Results

In this experimental setup, MobileNetV2, a lightweight convolutional neural network architecture, was employed. MobileNetV2 is known for its efficiency and reduced computational requirements, making it a suitable choice for this task. It was pre-trained on a substantial image database, providing a strong foundation for tumor classification. The model was trained using the following hyperparameters selected through the tunning process in Table 4.4.3, through 20 epochs until it reaches the convergence as shown in Figure 4.4.5.

Table 4.4.3 : MobileNet tunned hyperparameters with optimal values.

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0.0001
units	[32, 64, ….512]	384
dropout	[0.2,0.3,0.4,0.5]	0.3
Batch_size	[100,110,120….150]	100
Num_epochs	[10,20]	20

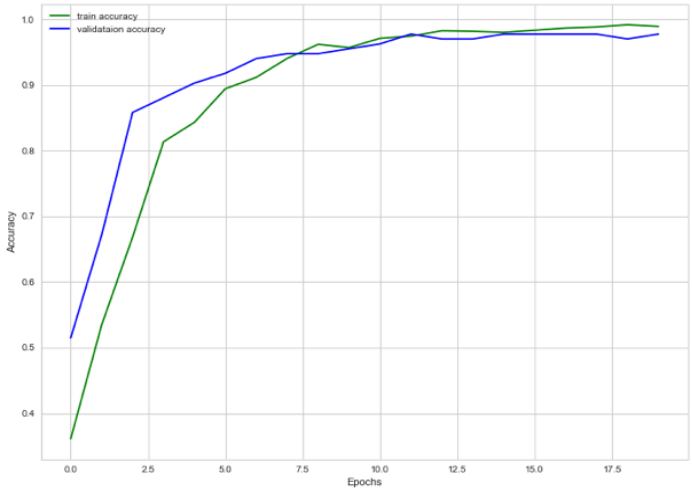


Figure 4.4.5 : Performance learning curves for training & validation accuracy MobileNet

As part of the model performance evaluation, different performance metrics were generated starting with the classification report and the confusion matrix shown in Table 4.4.4 and Figure 4.4.6.

Table 4.4.4 : Classification Report for MobileNet model

	Accuracy	Class	Precision	Recall	F1-Score
MobileNet	97.76%	**Benign (0)**	99.00%	99.00%	99.00%
		Malignant (1)	99.00%	100.00%	100.00%
		Normal (2)	98.00%	98.00%	98.00%

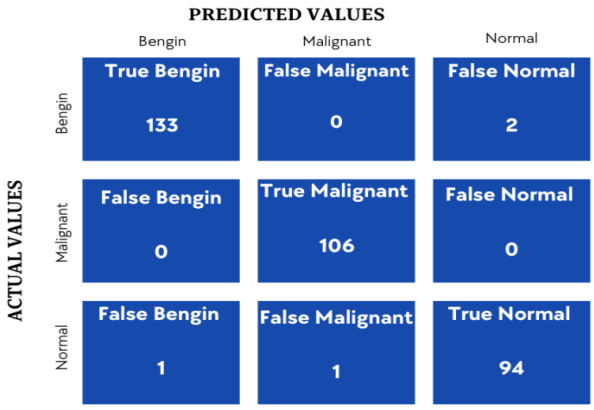


Figure 4.4.6 : Confusion matrix for MobileNet model

MobileNet shows an overall accuracy of 97.76% which is low when compared with other models' performance. The model achieved a precision of 99.00%, indicating a low rate of false positives,

and a perfect recall of 99.00%, capturing all actual benign cases. This results in an F1-Score of 99.00%, signifying strong performance in classifying benign cases Demonstrated a precision of 99.00%, showing its effectiveness in correctly identifying malignant cases with few false positives. The recall of 100.00% indicates it successfully captured all actual malignant cases, resulting in a robust F1-Score of 100.00%. In classifying normal cases, the model achieved a perfect precision of 100.00%, meaning it rarely misclassified normal cases, and a recall of 98.00%, capturing most actual normal cases. This leads to an F1-Score of 98.00%, highlighting its strong performance in identifying normal cases. ROC curve in Figure 4.4.7 is generated for the three classes to draw a clear image of the model performance.

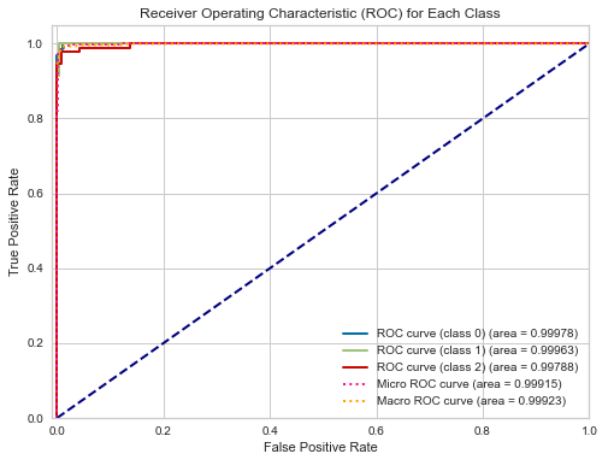


Figure 4.4.7 : Receiver operating characteristic (ROC) to Multi-Class Validation Set Mobile Net

4.4.3 Xception Results

In this experiment, the Xception model, a convolutional neural network that had been trained on a dataset containing over one million images, was employed to perform tumor classification. Like the approach used in previous models, the model was trained using the optimal hyperparameters in Table 4.4.5 for 20 epochs. Furthermore, as depicted in the Figure 4.4.8 , the graphical representation illustrates the model's training progress over time (in epochs), demonstrating a continuous improvement in the learning process.

Table 4.4.5 : Xception tunned hyperparameters with optimal values

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0
units	[32, 64, ….512]	480
dropout	[0.2,0.3,0.4,0.5]	0.3
Batch_size	[100,110,120….150]	150
Num_epochs	[10,20]	20

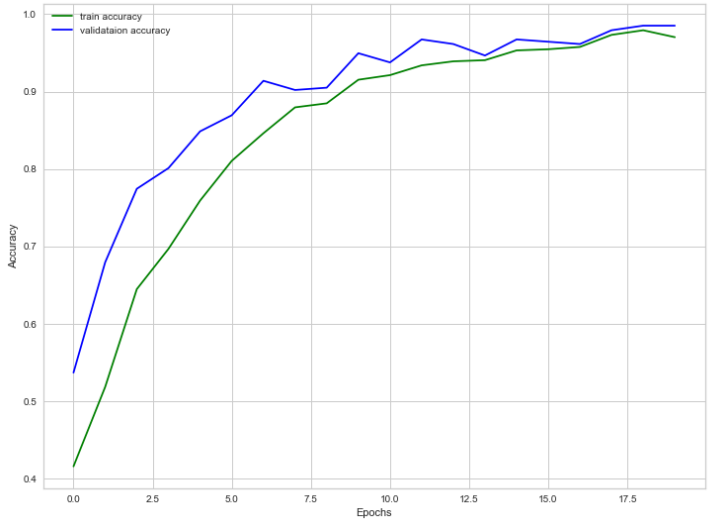


Figure 4.4.8 : Performance learning curves for training & validation accuracy Xception

In this evaluation, the Xception model in Table 4.4.6, Figure 4.4.9 and Figure 4.4.10, model achieved an overall accuracy of 97.32%. It exhibited excellent precision for all classes, between 98.00% - 99.00%, indicating a low rate of false positives in these classifications. Additionally, it displayed a high recall score for all classes, signifying its ability to effectively capture most actual cases in these categories. This remarkable balance between precision and recall for benign and malignant cases is reflected in their respective F1-Scores. The high precision, recall, and F1-Scores for all classes underscore the model's proficiency in accurately classifying benign, malignant, and normal tumor cases.

Table 4.4.6 : Classification Report for Xception

	Accuracy	Class	Precision	Recall	F1-Score
Xception	98.52%	**Benign (0)**	99.00%	100.00%	100.00%
		Malignant (1)	98.00%	99.00%	99.00%
		Normal (2)	99.00%	97.00%	98.00%

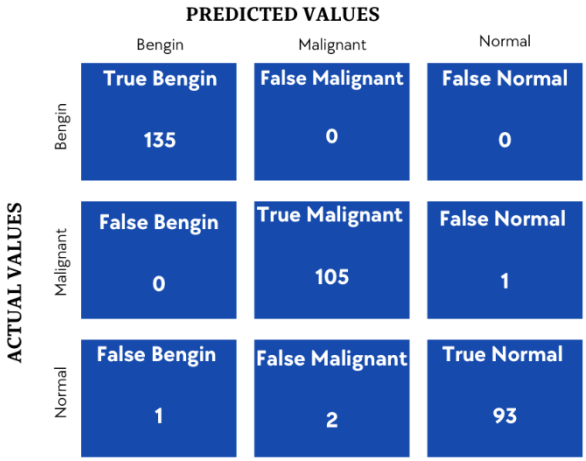


Figure 4.4.9 : Confusion Matrix for Xception

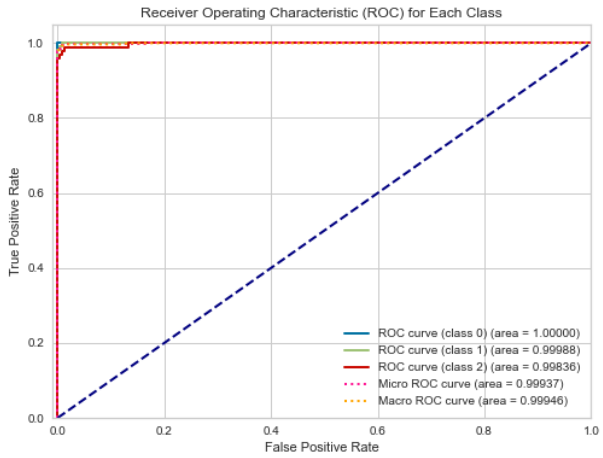


Figure 4.4.10 : Receiver operating characteristic (ROC) to Multi-Class Validation Set Xception

4.4.4 VGG19 Results

In this experiment, a pre-trained model made up of a sophisticated CNN network was called the VGG19 model. The VGG16 model gave rise to the deep form known as VGG19. The hyperparameter tuning process facilitated the selection of the best performing VGG19 model. Table 4.4.7 summarizes the tunned parameters with their optimal values. The model was trained using these parameters for 20 epochs, where the model as training progresses, the test's accuracy rises, but over time, training ends when the learning process is sufficiently established, as shown in the Figure 4.4.11 which shows the variation in the accuracy with the number of epochs.

Table 4.4.7 : VGG19 tunned hyperparameters with optimal values.

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0.0001
units	[32, 64, ….512]	384
dropout	[0.2,0.3,0.4,0.5]	0.3
Batch_size	[100,110,120….150]	100
Num_epochs	[10,20]	20

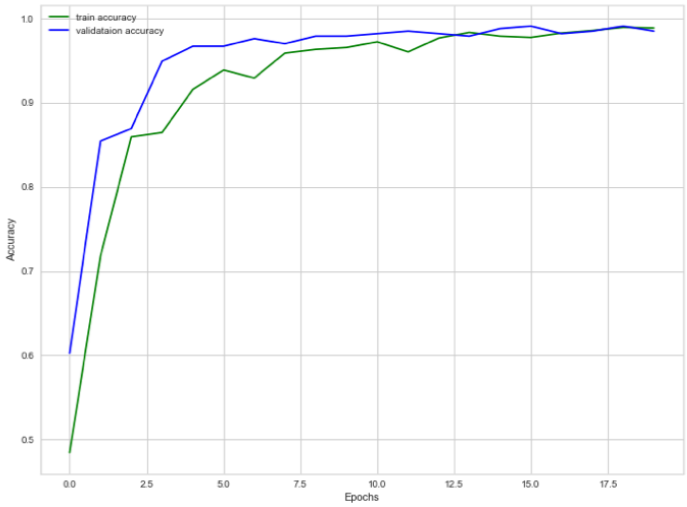


Figure 4.4.11 : Performance learning curves for training & validation accuracy MobileNet

As part of the model evaluation process classification report in Table 4.4.8, a confusion matrix in Figure 4.4.12 and ROC curve in Figure 4.4.13 were generated. In the case of VGG19, the model's performance in tumor classification is notable. For the benign cases (Class 0), the model showcases exceptional precision at 100.00%, meaning it rarely misclassifies benign cases, and a recall of 99.00%, signifying its effectiveness in capturing most actual benign cases, leading to a high F1-Score of 99.00%. In the context of malignant cases (Class 1), the model maintains strong precision at 97.00%, coupled with a recall of 100.00%, which results in a solid F1-Score of 99.00%. This showcases the model's proficiency in correctly identifying cancerous cases while minimizing false positives. Additionally, for normal cases (Class 2), the model maintains a precision of 99.00%, reflecting its accurate classification, and a recall of 96.00%. This balance between precision and recall results in an F1-Score of 07.00%, highlighting its efficacy in identifying normal cases. With an overall accuracy of 98.52%. The ROC curve areas of 0.997 for Class 0 (Benign), 0.100 for

Class 1 (Malignant), and 0.994 for Class 2 (Normal) indicate the model's strong discriminatory performance across all three classes.

Table 4.4.8 : Classification Report for VGG19

	Accuracy	Class	Precision	Recall	F1-Score
VGG19	98.52%	**Benign (0)**	99.00%	99.00%	99.00%
		Malignant (1)	97.00%	100.00%	99.00%
		Normal (2)	99.00%	96.00%	97.00%

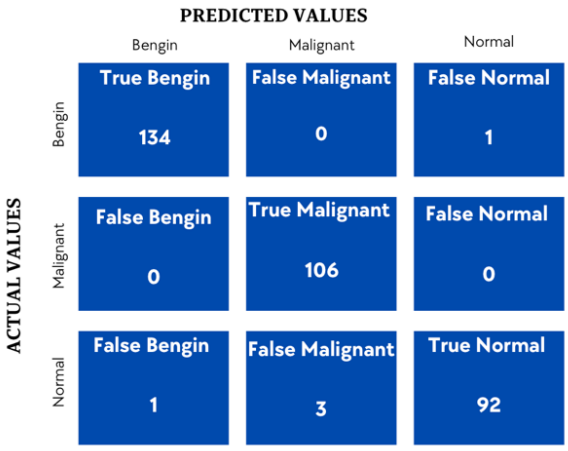


Figure 4.4.12 : Confusion Matrix for VGG19

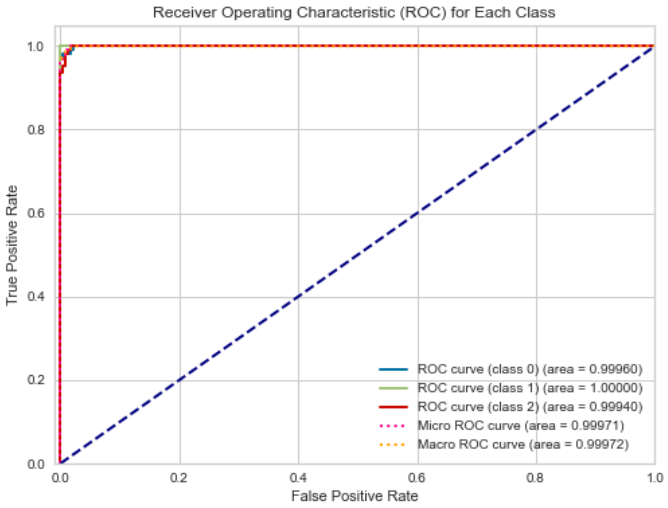


Figure 4.4.13 : Receiver operating characteristic (ROC) to Multi-Class Validation Set VGG19

4.4.5 VGG16 Results

In this experiment, the VGG-16 model is employed from the Visual Geometry Group (VGG) to perform Lung Cancer classification using a convolutional neural network (CNN). The VGG-16 model had been pre-trained on the ImageNet dataset. Figure 4.4.14 below shows a visual representation of the training process, which illustrates the learning and test validation curve as the number of epochs (time) progresses until it gets stable after 20 epochs. This implies that the model reaches the convergence point where it has learned as much as it can from the training data. Convergence means that the model parameters have settled into values that make it perform well on the training data. While this doesn't mean that it's only performing well in the training data it also becomes generalizable on unseen or validation data.

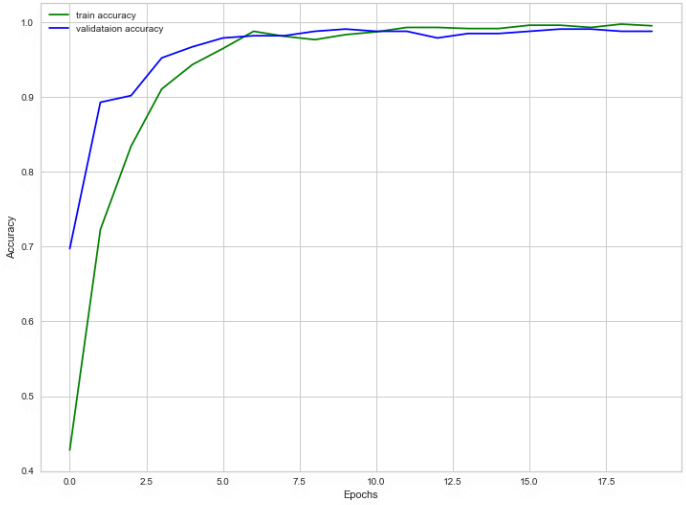


Figure 4.4.14 : Performance learning curves for training & validation accuracy VGG16

This model was trained using the optimal hyperparameters selected within the tunning process using the keras tuner library. Hyperparameters are listed in Table 4.4.9 below with the optimal selected value for each parameter:

Table 4.4.9 : VGG16 tunned hyperparameters with optimal values.

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0.0001
units	[32, 64, ….512]	384

dropout	[0.2,0.3,0.4,0.5]	0.2
Batch_size	[100,110,120….150]	140
Num_epochs	[10,20]	10

As part of the model performance evaluation, different performance metrics were generated starting with the classification report in Table 4.4.10 and the confusion matrix shown in Figure 4.4.15.

Table 4.4.10 : Classification Report for VGG16 model

	Accuracy	Class	Precision	Recall	F1-Score
VGG16	98.81%	**Benign (0)**	100.00%	98.00%	100.00%
		Malignant (1)	97.00%	100.00%	99.00%
		Normal (2)	99.00%	97.00%	98.00%

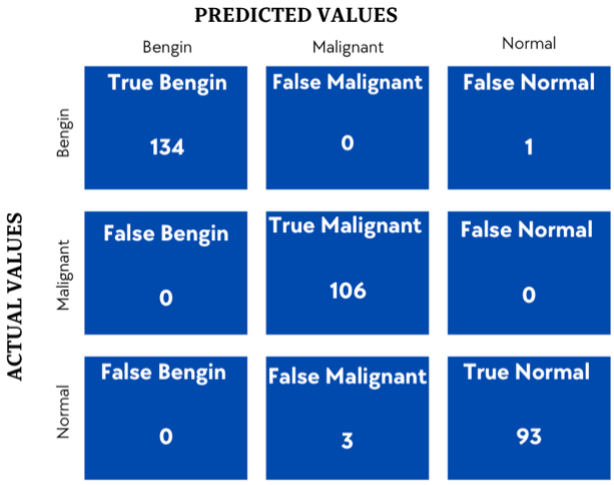


Figure 4.4.15 : Confusion matrix for VGG16 model

The model's test accuracy was 98.81% on the overall dataset and classes. The classification report and confusion matrix also show how the model performs on each class separately. For instance, on Benign**,** the model demonstrates a precision of 100.00% for this class, which means that 100.00% of the cases predicted as benign were indeed benign. With a recall of 98.00%, it successfully identifies 98.00% of actual benign cases. The F1-score of 100.00% reflects a balanced performance between precision and recall, making it effective in distinguishing non-cancerous conditions. Similarly, in the Malignant class, the model exhibits a precision, recall, and F1-score of 97.00% and 99.00% for malignant cases. This implies that every case identified as malignant is

genuinely cancerous, and it captures nearly all malignant cases. These metrics reflect the model's excellence in detecting cancerous conditions. While for the Normal cases, the model achieves a precision of 99.00%, signifying that 99.00%, with a recall of 98.00% and F1-score of 97.00% indicates that it performs well in classifying normal cases, striking a balance between precision and recall. These findings demonstrate the model's ability to classify the three classes without any bias toward any of them. This can be confirmed by drawing the ROC Curve in Figure 4.4.16. These findings were good when the points were in the upper-left corner; the current classifier equals 0.99, 0.99, and 0.99 for classes 0, 1, and 2, respectively, and the area under the curve occupies 99.99% of these curves for three classes.

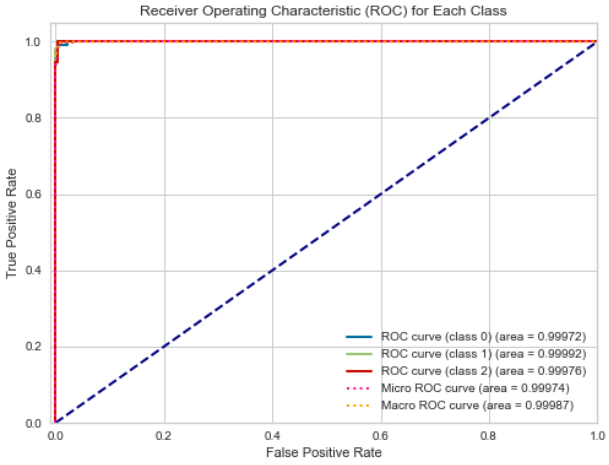


Figure 4.4.16 : Receiver operating characteristic (ROC) to Multi-Class Validation Set VGG16

4.4.6 EfficientNetB01 Results

The following research experiment utilizes the EfficientNetB01 model to conduct Lung Cancer classification through the implementation of a convolutional neural network (CNN). The diagram depicted in Figure 4.4.17 provides a graphical depiction of the training process, showcasing the learning and test validation curves as they evolve, measured in terms of the number of epochs until the model reach the convergence point.

The procedure of hyperparameter tuning assisted in the selection of the EfficientNetB01 model that exhibited the highest performance. Table 4.4.11 presented below provides a summary of the tuned parameters together with their corresponding optimal values.

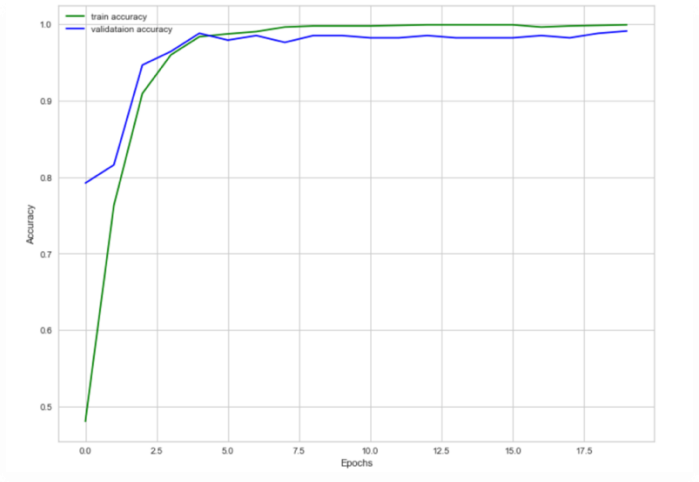


Figure 4.4.17 : Performance learning curves for training & validation accuracy EfficientNetB01

Table 4.4.11 : EfficientNetB01 tunned hyperparameters with optimal values.

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0
units	[32, 64, ….512]	256
dropout	[0.2,0.3,0.4,0.5]	0.4
Batch_size	[100,110,120….150]	100
Num_epochs	[10,20]	20

The confusion matrix in Figure 4.4.18 depicting the dataset of the model is presented in the subsequent chart, followed by a tabular representation of the classification report in Table 4.4.12.

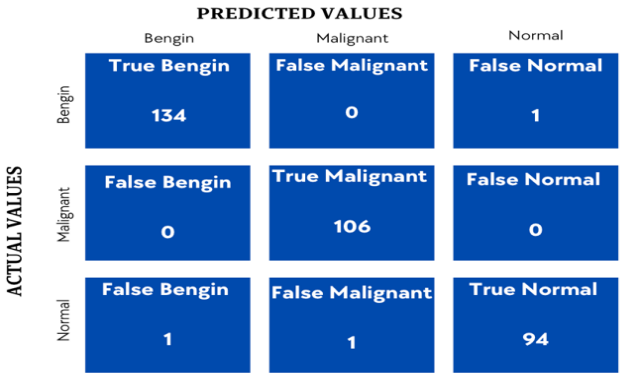


Figure 4.4.18 : Confusion matrix for EfficientNetB01 model

Table 4.4.12 : Classification Report for EfficientNetB01 model

	Accuracy	Class	Precision	Recall	F1-Score
EfficientNetB01	99.10%	**Benign (0)**	99.00%	99.00%	99.00%
		Malignant (1)	99.00%	100.00%	100.00%
		Normal (2)	99.00%	98.00%	99.00%

The overall testing accuracy of the model was 99.10 %. While it can be seen the model performs well in detecting each class separately. In the case of the Benign class, the model has a near-perfect precision of 99.00%, meaning it rarely misclassifies non-cancerous cases. It also captures most actual benign cases with a recall of 99.00%, resulting in a high F1-score of 99.00%, indicating a robust overall performance in identifying non-cancerous conditions. For cancerous cases (Malignant), the model's precision is excellent at 99.00%, indicating that it rarely misclassifies cancerous cases. With perfect recall at 100.00%, it captures all actual malignant cases. The F1-Score of 99.00% underscores its outstanding performance in identifying cancerous conditions. Besides robust precision at 99.00%, it implies its capacity to correctly classify normal cases. It exhibits a recall of 98.00%, highlighting its ability to capture most actual normal cases. The F1-Score of 99.00% further reinforces its effectiveness in distinguishing normal cases within the context of lung cancer classification. These metrics collectively portray a model that excels in accurately identifying lung cancer cases, both benign and malignant, as well as correctly

classifying normal cases. This can be seen on the ROC curve in Figure 4.4.19 as well with a very close area under the curve for all classes.

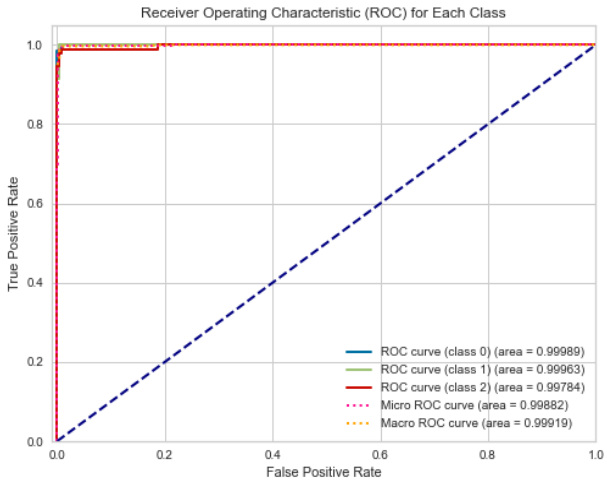


Figure 4.4.19 : Receiver operating characteristic (ROC) to Multi-Class Validation Set EfficientNetB01

4.4.7 ResNet50 Results

The 50-layer deep convolutional neural network ResNet50, which was pre-trained on a database with over a million photos, was employed in this experiment. The experimental findings about the model's performance indicate an accuracy rate of 99.4% during the testing phase. The curve below illustrates the progression of the model training procedure on the dataset, demonstrating the incremental improvement in the learning process with each epoch. Figure 4.4.20 illustrates that the learning process reaches a plateau when the model ceases to acquire new knowledge after 20 epochs, besides using the optimal parameters selected for this model in Table 4.4.13.

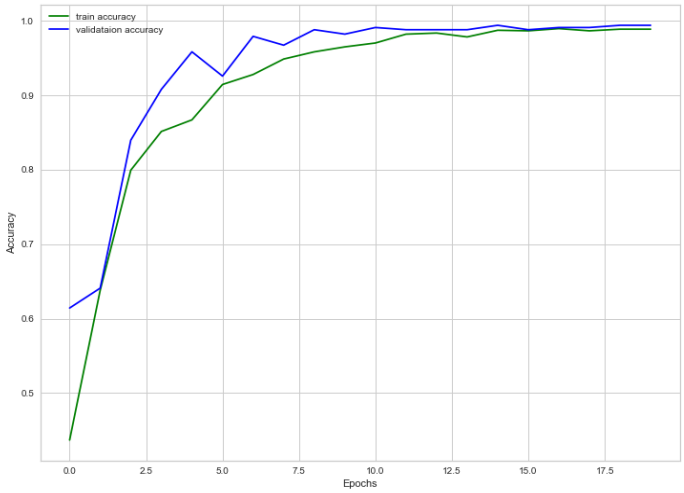


Figure 4.4.20 : Performance learning curves for training & validation accuracy ResNet50

Table 4.4.13 : ResNet50 tunned hyperparameters with optimal values

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
Weight_decay	[0, .0001, .001]	0.001
learning_rate	[0.001, 0.01]	0
units	[32, 64, ….512]	416
dropout	[0.2,0.3,0.4,0.5]	0.4
Batch_size	[100,110,120….150]	140
Num_epochs	[10,20]	20

MobileNet's achieves an exceptional performance as shown in Table 4.4.14 and the confusion matrix in Figure 4.4.21 with an accuracy of 99.40% besides the impressive precision and recall values. Precision represents the model's ability to make accurate positive predictions, and in this case, the high precision scores across all classes (100.00% for Benign, 100.00% for Malignant, and 98.00% for Normal) indicate that the model minimizes false positives, making it reliable in identifying each class. Recall, on the other hand, signifies the model's capacity to correctly capture all positive cases. The perfect recall of 100.00% for Malignant and Normal suggests that MobileNet successfully identifies all actual malignant and normal cases. The 99.00% recall for the Benign class demonstrates its effectiveness in capturing most actual cases in those categories, further validating its ability to minimize false negatives.

In addition to this excellent precision and recall scores, the high ROC curve areas in Figure 4.4.22 (.9995 for Benign, .100 for Malignant, and .9993 for Normal) reinforce Mobile Net’s strong discriminatory capabilities across all three classes. These areas signify the model's effectiveness

in distinguishing between the classes, with particularly outstanding performance for Malignant cases and very high performance for Benign and Normal cases.

Table 4.4.14 : Classification Report for ResNet50

	Accuracy	Class	Precision	Recall	F1-Score
ResNet50	99.40%	**Benign (0)**	100.00%	99.00%	99.00%
		Malignant (1)	100.00%	100.00%	100.00%
		Normal (2)	98.00%	100.00%	99.00%

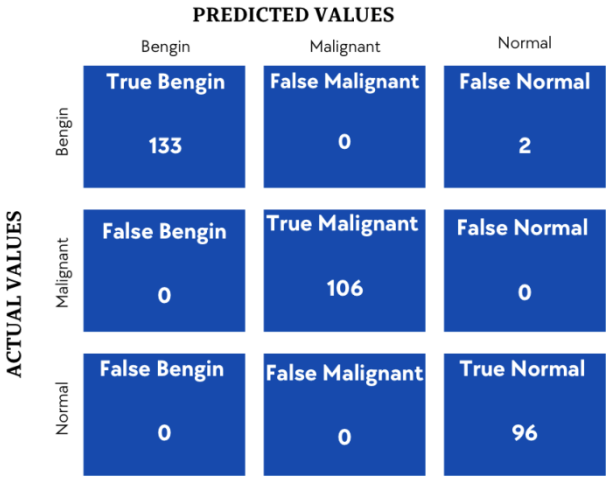


Figure 4.4.21 : Confusion Matrix for ResNet50

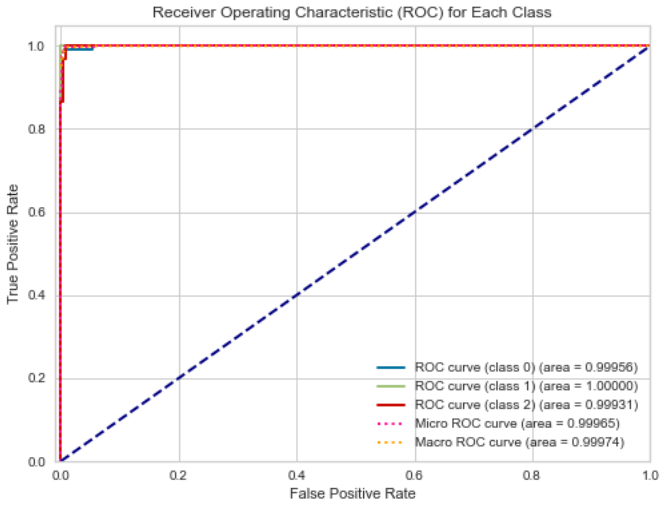


Figure 4.4.22 : Receiver operating characteristic (ROC) to Multi-Class Validation Set ResNet50

4.4.8 CNN Results

The CNN model's architecture was meticulously designed with careful consideration of key hyperparameters to optimize its performance. The hyperparameter tuning process revealed several

critical choices that contributed to the model's efficacy. Firstly, the conv1_units hyperparameter, controlling the number of units in the initial convolutional layer, was explored with values ranging from 32 to 256. After rigorous optimization, the optimal value for conv1_units were determined to be 128. Furthermore, the num_conv2d_layers hyperparameter, which dictates the number of convolutional layers, was fine-tuned within the range of 1 to 3. The optimal choice was to incorporate three convolutional layers. The architecture's convolutional layers, represented by conv2d_units_0, conv2d_units_1, and conv2d_units_2, were equally crucial components. conv2d_units_0 was tuned with values spanning from 32 to 256 and was determined to perform optimally at 96 units. conv2d_units_1, representing the second convolutional layer, exhibited the same optimal value of 96 units as conv2d_units_0. Lastly, conv2d_units_2, controlling the third convolutional layer, also settled at a value of 96 units as shown in Table 4.4.15. The structured model was trained using 20 epochs and a learning rate of .001 illustrated in Figure 4.4.23.

Table 4.4.15 : CNN tunned hyperparameters with optimal values.

Hyperparameter Name	Hyperparameter Values	Hyperparameters Optimal Value
conv1_units	[32, 64, ….256]	128
num_conv2d_layers	[1,2,3]	3
conv2d_units_0	[32, 64, ….256]	160
dense_units	[32, 64, ….256]	96
conv2d_units_1	[32, 64, ….256]	96
conv2d_units_2	[32, 64, ….256]	96

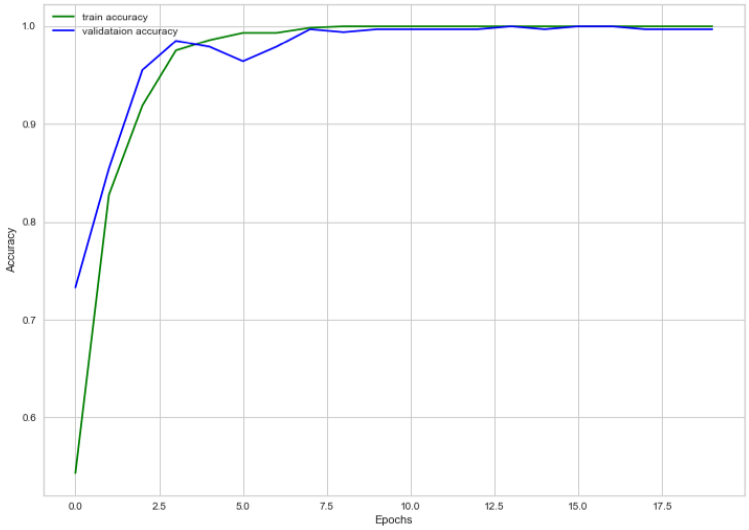


Figure 4.4.23 : Performance learning curves for training & validation accuracy CNN

The top-performing CNN model looking at the classification report in Table 4.4.16, confusion matrix Figure 4.4.24 and ROC Curve in Figure 4.4.25, it boasts an impressive overall accuracy of 99.70%, demonstrates outstanding performance across all three classes. For Class 1 (denoting the Malignant category), the model exhibits a precision of 100.00%, signifying its high accuracy in classifying cases within this category, and a perfect recall of 100.00%, meaning it successfully captures all actual Class 0 cases. This exceptional precision-recall balance results in a perfect F1-Score of 100.00%, underscoring the model's flawless ability to identify cases in this class. Similarly, for Class 0, the model excels with a precision of 100.00%, indicating that it rarely misclassifies cases in this category, and a recall of 99.00%, capturing almost all actual Class 1 cases. The F1-Score of 100.00% once again signifies its remarkable effectiveness in identifying cases within this class. Class 2 experiences equally strong performance, with a precision of 99.00%, signifying high accuracy in classifying cases within this category, and a recall of 100.00%, indicating that it effectively captures most actual Class 2 cases. The optimal trade-off between precision and recall yields an outstanding F1-Score of 99.00%, underscoring the model's high level of ability in correctly identifying instances belonging to Class 2. In conclusion, the CNN model with a 99.70% accuracy showcases exceptional precision, recall, and F1-Scores across all classes, making it an outstanding performer in accurately classifying cases within this specific problem domain.

Table 4.4.16 : Classification Report for CNN model.

	Accuracy	Class	Precision	Recall	F1-Score
CNN	99.70%	**Benign (0)**	100.00%	99.00%	100.00%
		Malignant (1)	100.00%	100.00%	100.00%
		Normal (2)	99.00%	100.00%	99.00%

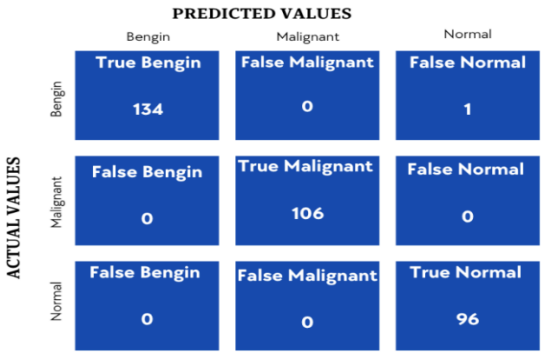


Figure 4.4.24 : Confusion matrix for CNN model

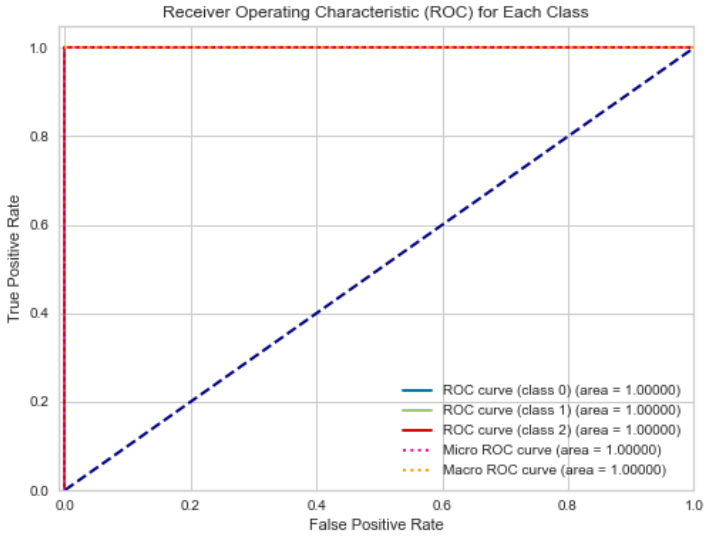


Figure 4.4.25 : Receiver operating characteristic (ROC) to Multi-Class Validation Set CNN

Summarizing the results of all DL experiments as shown in Table 4.4.17, in the context of a lung cancer classification problem, the performance of various deep learning models is evaluated across multiple metrics. The CNN model exhibits an outstanding overall accuracy of 99.70%, accompanied by perfect precision, recall, and F1 score, all at 100%. This exceptional accuracy highlights the effectiveness of the Convolutional Neural Network architecture in capturing and classifying complex patterns within the dataset. ResNet50 closely follows with a high accuracy of 99.40%, showcasing a well-balanced performance with 99% precision, 100% recall, and a 99% F1 score. EfficientNetB01, VGG16, Xception, and MobileNet demonstrate consistent accuracy levels above 97%, with precision, recall, and F1 scores consistently ranging from 98% to 100%. VGG19 and DenseNet121, while slightly trailing in accuracy, still maintain robust performance

with precision, recall, and F1 scores averaging around 98%. These results collectively underscore the efficacy of various deep learning architectures in the given task, with each model demonstrating a strong ability to generalize and accurately classify the data.

Table 4.4.17 : Table summarizing of the overall accuracy of all classification models for the CT Images dataset.

Model Type	Accuracy	Precision	Recall	F1-Score
CNN	***99.70%***	***100%***	***100%***	***100%***
ResNet50	99.40%	99%	100%	99%
EfficientNetB01	99.10%	99%	99%	99%
VGG16	98.81%	99%	99%	99%
VGG19	98.52%	98%	98%	98%
Xception	98.52%	99%	99%	99%
MobileNet	97.76%	99%	99%	99%
DenseNet121	97.32%	98%	98%	98%

In a comparative analysis of the proposed CNN model and a published model using the metrics shown in Table 4.4.18 below, it's evident that the proposed model outperforms the published one in several important aspects. Firstly, when considering the overall accuracy, the model achieves an accuracy rate of 99.70%, while the published model lags slightly behind at 99.45%. This means that this model makes more accurate predictions on the test data. Diving deeper into class-specific metrics, for example, the "Benign" class (Class 0), the proposed model demonstrates a precision of 100.00%. This signifies that out of the instances, this model predicted as "Benign," 100.00% of them were indeed "Benign." On the other hand, the published model had a precision of 98.00% in this class. In terms of recall, the proposed model excels with a rate of 100.00% for the "Benign" class. This indicates that it correctly identifies almost all actual "Benign" cases, leaving no room for false negatives. In contrast, the published model achieved a recall rate of 98.00% in this class,

suggesting that it may have missed a few actual "Benign" cases. When considering the F1-Score, the proposed model, at 100.00%, again outperforms the published model for the "Benign" class, which also achieved a perfect score.

Table 4.4.18 : Comparison between the model in [134] *and the proposed model in this study.*

	Accuracy	Class	Precision	Recall	F1-Score
CNN Proposed Model	99.70%	**Benign (0)**	100.00%	99.00%	100.00%
		Malignant (1)	100.00%	100.00%	100.00%
		Normal (2)	99.00%	100.00%	99.00%
CNN Published Model [134]	99.45%	**Benign (0)**	98%	98%	98%
		Malignant (1)	100%	99%	100%
		Normal (2)	98%	98%	99%

This improvement can be explained by using a different preprocessing technique on the proposed model that includes resizing, denoising, CLAHE, and ostus segmentation rather than just applying the resizing, denoising, and CLAHE filtering on the published model. More robust data preprocessing techniques can enhance the quality and consistency of the training data, leading to improved model performance. Besides adopting different augmentation techniques while at the same time making sure that the data are balanced which wasn't considered within the published model. The last contribution on the proposed model was related to the CNN model where the architecture is more complex and optimized. It includes a higher number of Conv2D layers (4 as compared to 3 in the published model) and has performed hyperparameter optimization to determine the optimal number of Conv2D layers and units. This increased complexity allows the proposed model to capture more intricate features and patterns in the data. Additionally, relu activation functions have been chosen for the Conv2D layers, which are known for their effectiveness in training deep neural networks. These architectural enhancements and optimization efforts contribute to the proposed model's superior performance compared to the simpler architecture of the published model.

4.5 Challenges and Limitation

Throughout this thesis, a multitude of challenges and limitations were faced that profoundly influenced the work. Starting with, the data collection phase, both the global datasets used were limited in size. Consequently, it was compelled to employ oversampling and image augmentation techniques to make up for the lack of diversity and volume in the dataset.

On the local front, data collection posed a different set of challenges. Local hospitals often lacked standardized and efficient management systems for patient data, complicating the process of data retrieval and integration. These challenges encompassed issues related to data organization, standardization, and accessibility, necessitating a careful approach to harness the full potential of locally sourced data. Through the local data collection process, we've communicated with the Director of the Scientific Unit for Health Education and Research Dr. Abdullah Qawasmi, and the Director of the General Administration for Information Technology in the Ministry of Health to get the needed approvals (See the appendices). Starting with Rafidia Hospital in Nablus unfortunately the data related to lung cancer was very limited seeing few cases in the past three years, so they've redirected the request to Al-Watani Hospital which is considered as the most important cancer Hospital in Nablus. In Al Watani Hospital I've worked with the Head of the Radiology Department Dr. Ali Salman and from the IT department Eng. Ahmed AbuSalha. Regretfully the data was limited as well specifically the control group (healthy people), for instance in 2023 there have been 500 cases of malignancy reported vs less than 10 cases reported as benign or healthy. These images are saved on an external server in DICOM format, which can be complex to manually convert to JPG, especially since each patient's image dataset typically contains hundreds of slices. Efforts to export these images directly via email have been made, but they've encountered a significant ethical concern. The exported images still contain patient-specific details, which raises privacy and confidentiality issues.

Furthermore, the quality of the collected images was another concern. Data often required extensive preprocessing to ensure they met the essential quality standards. This preprocessing involved tasks such as noise reduction, artifact removal, and the standardization of image formats, all of which were critical to guarantee that the input data was suitable for deployment in deep learning models.

Lastly, the hardware used for model training also presented its limitations. Although we conducted training on a computer with reasonably good specifications, it's important to note that more powerful hardware could have potentially led to better results and enabled a broader range of experiments to be conducted. These hardware constraints underscored the importance of resource availability in deep learning research.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

Lung cancer remains a significant global health concern, with early detection playing a pivotal role in improving patient outcomes and reducing mortality rates. Misdiagnosis can lead to delayed treatment and compromised prognosis, underscoring the importance of accurate and efficient diagnostic tools in the fight against this formidable disease.

The study and analysis of deep learning and Machine Learning techniques for the detection of lung cancer, focusing on the most recent DL algorithms can obtain results, even better, then those that had been obtained by experts. The tuning of the DL parameters improves the classification result, the result obtained is that the system generates an increase of the classification accuracy, with a good balance between precision and sensitivity.

In this thesis, the application of advanced techniques in medical image processing, machine learning, and deep learning have been explored to develop a Lung Cancer Detection System.

Many ML models (DT, SVM, XGB, LGB, Adaboost, and GBM) were applied to the Demographic/ risk factors dataset, the models that achieved the highest classification accuracy were the XGB and Adaboost with an accuracy of 99.07%. Also, well DL models (CNN, MobileNet, Xception, DenseNet, VGG16, VGG19, and EfficientNet) were applied to the CT scans dataset to classify the lung cancer along with various medical images preprocessing and augmentation where the CNN achieved the optimal performance of 99.70%.

The finding of this study has demonstrated the potential of these innovative approaches in enhancing the early identification and classification of lung cancer, addressing a crucial gap in current medical practice.

This research contributes to the ongoing efforts to advance the field of medical imaging and cancer diagnosis. The Lung Cancer Detection System we have developed can serve as a valuable tool to aid medical professionals in making more informed decisions, enabling early intervention, personalized treatment strategies, and improved patient care. Moreover, this work highlights the significance of interdisciplinary collaboration between the fields of medicine, computer science, and artificial intelligence. This synergy is crucial for driving innovation in healthcare and holds the promise of more accurate and timely lung cancer diagnoses.

5.2 Future Work and Recommendations

Several avenues for further research and development emerge from the findings and limitations of this study. Firstly, an expansion of the used dataset mainly a local dataset through a collaborative effort with multiple healthcare institutions could facilitate the acquisition of larger and more diverse local datasets for lung cancer detection systems. This would address the challenge of data scarcity and enhance the system's ability to generalize across different patient profiles.

Moreover, advancing image preprocessing techniques by investigating state-of-the-art methods can further elevate the quality of input data. Furthermore, the investigation of other deep learning approaches and hybrid models should be on the future agenda. Exploring alternatives to current models may provide more accurate and efficient solutions for lung cancer detection.

Efforts to achieve real-time processing in clinical settings, with the integration of a user-friendly graphical interface (GUI), are also worth exploring. This would not only enhance the efficiency and timeliness of diagnosis but also improve the accessibility of the lung cancer detection system for healthcare professionals. Finally, collaborating with experts from various disciplines, including computer science, medicine, and healthcare management, presents a promising approach to effectively address the multifaceted challenges in lung cancer detection. This collective effort will

undoubtedly pave the way for a more comprehensive and impactful future in the realm of medical image analysis and cancer detection.

Collecting and developing a nationally available dataset of Lung cancer images made up of public or private institutions that consider the types of Lung cancer, its stage, number of patients, number of images, and type of medical images will be a utile application using DL and ML.

Bibliography

[1] C. Woodman, G. Vundu, A. George, and C. M. Wilson, “Applications and strategies in nanodiagnosis and nanotherapy in lung cancer,” *Seminars in Cancer Biology*, vol. 69. Academic Press, pp. 349–364, Feb. 01, 2021. doi: 10.1016/j.semcancer.2020.02.009.

[2] “Cancer research UK. Annual report. 2017-2018.” Accessed: Jul. 11, 2023. [Online]. Available: https://www.cancerresearchuk.org/sites/default/files/cruk_annual_report_2017_18_final.pdf

[3] H. Sung *et al.*, “Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries,” *CA Cancer J Clin*, vol. 71, no. 3, pp. 209–249, May 2021, doi: 10.3322/caac.21660.

[4] R. L. Siegel, K. D. Miller, H. E. Fuchs, and A. Jemal, “Cancer Statistics, 2021,” *CA Cancer J Clin*, vol. 71, no. 1, pp. 7–33, Jan. 2021, doi: 10.3322/caac.21654.

[5] Y. Hamdi *et al.*, “Cancer in Africa: The Untold Story,” *Front Oncol*, vol. 11, Apr. 2021, doi: 10.3389/fonc.2021.650117.

[6] W. Mapanga *et al.*, “Consensus study on the health system and patient-related barriers for lung cancer management in South Africa,” *PLoS One*, vol. 16, no. 2 Febuary, Feb. 2021, doi: 10.1371/journal.pone.0246716.

[7] J. Li, A. S. M. Lam, S. T. Y. Yau, K. K. L. Yiu, and K. K. F. Tsoi, “Antihypertensive treatments and risks of lung Cancer: a large population-based cohort study in Hong Kong,” *BMC Cancer*, vol. 21, no. 1, Dec. 2021, doi: 10.1186/s12885-021-08971-6.

[8] A. Shankar *et al.*, “Feasibility of lung cancer screening in developing countries: Challenges, opportunities and way forward,” *Translational Lung Cancer Research*, vol. 8. AME Publishing Company, pp. S106–S121, 2019. doi: 10.21037/tlcr.2019.03.03.

[9] T. Pereira *et al.*, “Comprehensive perspective for lung cancer characterisation based on AI solutions using CT images,” *J Clin Med*, vol. 10, no. 1, pp. 1–10, Jan. 2021, doi: 10.3390/jcm10010118.

[10] M. Masud, N. Sikder, A. Al Nahid, A. K. Bairagi, and M. A. Alzain, “A machine learning approach to diagnosing lung and colon cancer using a deep learning-based classification framework,” *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–21, Feb. 2021, doi: 10.3390/s21030748.

[11] R. Daneshjou, B. He, D. Ouyang, and J. Y. Zou, “How to evaluate deep learning for cancer diagnostics – factors and recommendations,” *Biochimica et Biophysica Acta - Reviews on Cancer*, vol. 1875, no. 2. Elsevier B.V., Apr. 01, 2021. doi: 10.1016/j.bbcan.2021.188515.

[12] V. Ambrosini *et al.*, “PET/CT imaging in different types of lung cancer: An overview,” *Eur J Radiol*, vol. 81, no. 5, pp. 988–1001, 2012, doi: https://doi.org/10.1016/j.ejrad.2011.03.020.

[13] K. Latifi *et al.*, “Impact of dose on lung ventilation change calculated from 4D-CT using deformable image registration in lung cancer patients treated with SBRT,” *J Radiat Oncol*, vol. 4, no. 3, pp. 265–270, 2015, doi: 10.1007/s13566-015-0200-0.

[14] L. S.K., S. N. Mohanty, S. K., A. N., and G. Ramirez, “Optimal deep learning model for classification of lung cancer on CT images,” *Future Generation Computer Systems*, vol. 92, pp. 374–382, 2019, doi: https://doi.org/10.1016/j.future.2018.10.009.

[15] A. Bustos García de Castro *et al.*, “PET-CT in presurgical lymph node staging in non-small cell lung cancer: The importance of false-negative and false-positive findings,”

Radiología (English Edition), vol. 59, no. 2, pp. 147–158, 2017, doi: 10.1016/j.rxeng.2016.12.005.

[16] S. Yaturu and R. A. Patel, "Metastases to the Thyroid Presenting as a Metabolically Inactive Incidental Thyroid Nodule with Stable Size in 15 Months," *Case Rep Endocrinol*, vol. 2014, pp. 1–5, 2014, doi: 10.1155/2014/643986.

[17] W. K. Lee, E. W. F. Lau, K. Chin, O. Sedlaczek, and K. Steinke, "Modern diagnostic and therapeutic interventional radiology in lung cancer," *Journal of Thoracic Disease*, vol. 5, no. SUPPL.5. Pioneer Bioscience Publishing, 2013. doi: 10.3978/j.issn.2072-1439.2013.07.27.

[18] M. Zurek, A. Bessaad, K. Cieslar, and Y. Crémillieux, "Validation of simple and robust protocols for high-resolution lung proton MRI in mice," *Magn Reson Med*, vol. 64, no. 2, pp. 401–407, 2010, doi: 10.1002/mrm.22360.

[19] N. S. Burris *et al.*, "Detection of small pulmonary nodules with ultrashort echo time sequences in oncology patients by using a PET/MR system," *Radiology*, vol. 278, no. 1, pp. 239–246, Jan. 2016, doi: 10.1148/radiol.2015150489.

[20] A. Cieszanowski *et al.*, "Characterization of focal liver lesions using quantitative techniques: Comparison of apparent diffusion coefficient values and T2 relaxation times," *Eur Radiol*, vol. 22, no. 11, pp. 2514–2524, Nov. 2012, doi: 10.1007/s00330-012-2519-x.

[21] C. Fink *et al.*, "Lung MRI at 1.5 and 3 Tesla: Observer Preference Study and Lesion Contrast Using Five Different Pulse Sequences," *Invest Radiol*, vol. 42, no. 6, 2007, [Online]. Available: https://journals.lww.com/investigativeradiology/Fulltext/2007/06000/Lung_MRI_at_1_5_and_3_Tesla__Observer_Preference.7.aspx

[22] M. ; H. Groth F. O.; Bannas P.; Muellerleile K.; Adam G.; Regier M., "Intraindividual Comparison of Contrast-Enhanced MRI and Unenhanced SSFP Sequences of Stenotic and Non-stenotic Pulmonary Artery Diameters," *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*, vol. 183, no. 01, pp. 47–53, 2011, doi: 10.1055/s-0029-1245568.

[23] A. L. Chong, R. V. Chandra, K. C. Chuah, E. L. Roberts, and S. L. Stuckey, "Proton density MRI increases detection of cervical spinal cord multiple sclerosis lesions compared with T2-weighted fast spin-echo," *American Journal of Neuroradiology*, vol. 37, no. 1, pp. 180–184, Jan. 2016, doi: 10.3174/ajnr.A4476.

[24] S Al-Zeibak and N H Saunders, "A feasibility study of in vivo electromagnetic imaging," *Phys Med Biol*, vol. 38, no. 1, p. 151, 1993, doi: 10.1088/0031-9155/38/1/011.

[25] "Reduced Lung-Cancer Mortality with Low-Dose Computed Tomographic Screening," *New England Journal of Medicine*, vol. 365, no. 5, pp. 395–409, Aug. 2011, doi: 10.1056/NEJMoa1102873.

[26] S. Shen *et al.*, "A Bayesian model for estimating multi-state disease progression," *Comput Biol Med*, vol. 81, pp. 111–120, Feb. 2017, doi: 10.1016/j.compbiomed.2016.12.011.

[27] W.-J. Choi and T.-S. Choi, "Genetic programming-based feature transform and classification for the automatic detection of pulmonary nodules on computed tomography images," *Inf Sci (N Y)*, vol. 212, pp. 57–78, 2012, doi: https://doi.org/10.1016/j.ins.2012.05.008.

[28] P. Huang *et al.*, "Added value of computer-aided CT image features for early lung cancer diagnosis with small pulmonary nodules: A matched case-control study," *Radiology*, vol. 286, no. 1, pp. 286–295, Jan. 2018, doi: 10.1148/radiol.2017162725.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[30] S. L. Fernandes, V. P. Gurupur, H. Lin, and R. J. Martis, "A novel fusion approach for early lung cancer detection using computer aided diagnosis techniques," *J Med Imaging Health Inform*, vol. 7, no. 8, pp. 1841–1850, Dec. 2017, doi: 10.1166/jmihi.2017.2280.
[31] "Unsupervised Feature Learning and Deep Learning Tutorial." Accessed: Jul. 14, 2023. [Online]. Available: http://deeplearning.stanford.edu/tutorial/
[32] X. Guan, W. Gao, H. Peng, N. Shu, and D. W. Gao, "Image-Based Incipient Fault Classification of Electrical Substation Equipment by Transfer Learning of Deep Convolutional Neural Network," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 45, no. 1, pp. 1–8, 2022, doi: 10.1109/ICJECE.2021.3109293.
[33] K. Warin, W. Limprasert, S. Suebnukarn, S. Jinaporntham, and P. Jantana, "Performance of deep convolutional neural network for classification and detection of oral potentially malignant disorders in photographic images," *Int J Oral Maxillofac Surg*, vol. 51, no. 5, pp. 699–704, 2022, doi: https://doi.org/10.1016/j.ijom.2021.09.001.
[34] A. Magge, D. Weissenbacher, A. Sarker, M. Scotch, and G. Gonzalez-Hernandez, "Bi-directional Recurrent Neural Network Models for Geographic Location Extraction in Biomedical Literature." [Online]. Available: http://www.geonames.org/
[35] "About Lung Cancer What Is Lung Cancer?," *American Cancer Society*, 1971.
[36] P. P. Massion and R. C. Walker, "Indeterminate pulmonary nodules: Risk for having or for developing lung cancer?," *Chest*, vol. 146, no. 6. American College of Chest Physicians, pp. 1173–1178, Dec. 01, 2014. doi: 10.1158/1940-6207.CAPR-14-0364.
[37] "Lung Cancer | Kaggle." Accessed: Jul. 11, 2023. [Online]. Available: https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer/discussion?resource=download
[38] hamdalla alyasriy, "'The IQ-OTHNCCD lung cancer dataset', Mendeley Data, V1," 2020, Accessed: Jul. 12, 2023. [Online]. Available: https://data.mendeley.com/datasets/bhmdr45bh2/1
[39] R. P. R, R. A. S. Nair, and V. G, "A Comparative Study of Lung Cancer Detection using Machine Learning Algorithms," in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2019, pp. 1–4. doi: 10.1109/ICECCT.2019.8869001.
[40] I. M. Nasser and S. S. Abu-Naser, "Lung Cancer Detection Using Artificial Neural Network," 2019. [Online]. Available: www.ijeais.org
[41] A. S. Oliver *et al.*, "Early detection of lung carcinoma using machine learning," *Intelligent Automation and Soft Computing*, vol. 30, no. 3, pp. 755–770, 2021, doi: 10.32604/iasc.2021.016242.
[42] E. Dritsas and M. Trigka, "Lung Cancer Risk Prediction with Machine Learning Models," *Big Data and Cognitive Computing*, vol. 6, no. 4, Dec. 2022, doi: 10.3390/bdcc6040139.
[43] A. Kumar Gulia and R. Bhatt, "Lung Cancer Prediction Using Machine Learning Classifiers," 2021.
[44] N. S. Nadkarni and S. Borkar, "Detection of Lung Cancer in CT Images using Image Processing," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 863–866. doi: 10.1109/ICOEI.2019.8862577.
[45] H. Yu, Z. Zhou, and Q. Wang, "Deep Learning Assisted Predict of Lung Cancer on Computed Tomography Images Using the Adaptive Hierarchical Heuristic Mathematical Model," *IEEE Access*, vol. 8, pp. 86400–86410, 2020, doi: 10.1109/ACCESS.2020.2992645.

[46] S. Nageswaran *et al.*, “Lung Cancer Classification and Prediction Using Machine Learning and Image Processing,” *Biomed Res Int*, vol. 2022, 2022, doi: 10.1155/2022/1755460.

[47] Q. Firdaus, R. Sigit, T. Harsono, and A. Anwar, “Lung cancer detection based on ct-scan images with detection features using gray level co-occurrence matrix (glcm) and support vector machine (svm) methods,” in *IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 643–648. doi: 10.1109/IES50839.2020.9231663.

[48] R. Tekade and K. Rajeswari, “Lung Cancer Detection and Classification Using Deep Learning,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 2018, pp. 1–5. doi: 10.1109/ICCUBEA.2018.8697352.

[49] Y. Said, A. A. Alsheikhy, T. Shawly, and H. Lahza, “Medical Images Segmentation for Lung Cancer Diagnosis Based on Deep Learning Architectures,” *Diagnostics*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/diagnostics13030546.

[50] A. Masood *et al.*, “Automated Decision Support System for Lung Cancer Detection and Classification via Enhanced RFCN With Multilayer Fusion RPN,” *IEEE Trans Industr Inform*, vol. 16, no. 12, pp. 7791–7801, 2020, doi: 10.1109/TII.2020.2972918.

[51] A. S. U., F. R. P. P., A. Abraham, and D. Stephen, “Deep Learning-Based BoVW–CRNN Model for Lung Tumor Detection in Nano-Segmented CT Images,” *Electronics (Basel)*, vol. 12, no. 1, 2023, doi: 10.3390/electronics12010014.

[52] A. A. Alsheikhy, Y. Said, T. Shawly, A. K. Alzahrani, and H. Lahza, “A CAD System for Lung Cancer Detection Using Hybrid Deep Learning Techniques,” *Diagnostics*, vol. 13, no. 6, Mar. 2023, doi: 10.3390/diagnostics13061174.

[53] C. C. Nguyen, G. S. Tran, V. T. Nguyen, J.-C. Burie, and T. P. Nghiem, “Pulmonary Nodule Detection Based on Faster R-CNN With Adaptive Anchor Box,” *IEEE Access*, vol. 9, pp. 154740–154751, 2021, doi: 10.1109/ACCESS.2021.3128942.

[54] A. Masood *et al.*, “Cloud-Based Automated Clinical Decision Support System for Detection and Diagnosis of Lung Cancer in Chest CT,” *IEEE J Transl Eng Health Med*, vol. 8, 2020, doi: 10.1109/JTEHM.2019.2955458.

[55] I. Nazir, I. U. Haq, M. M. Khan, M. B. Qureshi, H. Ullah, and S. Butt, “Efficient pre-processing and segmentation for lung cancer detection using fused CT images,” *Electronics (Switzerland)*, vol. 11, no. 1, Jan. 2022, doi: 10.3390/electronics11010034.

[56] H. Cao *et al.*, “A Two-Stage Convolutional Neural Networks for Lung Nodule Detection,” *IEEE J Biomed Health Inform*, vol. 24, no. 7, pp. 2006–2015, 2020, doi: 10.1109/JBHI.2019.2963720.

[57] Y.-W. Chen and C.-J. Lin, “Combining SVMs with Various Feature Selection Strategies,” in *Feature Extraction: Foundations and Applications*, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 315–324. doi: 10.1007/978-3-540-35488-8_13.

[58] G. Erol Doğan and B. Uzbaş, “Diagnosis of COVID-19 from blood parameters using convolutional neural network,” *Soft comput*, vol. 27, no. 15, pp. 10555–10570, Aug. 2023, doi: 10.1007/s00500-023-08508-y.

[59] N. V Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” 2002.

[60] D. Elreedy and A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” *Inf Sci (N Y)*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.

[61] “SMOTE explained for noobs – Synthetic Minority Over-sampling TEchnique line by line | Rich Data.” Accessed: Oct. 28, 2023. [Online]. Available: https://rikunert.com/smote_explained

[62] M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0263-7.

[63] E.-M. Pulfer, “Different Approaches to Blurring Digital Images and Their Effect Different Approaches to Blurring Digital Images and Their Effect on Facial Detection on Facial Detection.” [Online]. Available: https://scholarworks.uark.edu/csceuht/66

[64] T. G. Devi, N. Patil, S. Rai, and C. S. Philipose, “Gaussian Blurring Technique for Detecting and Classifying Acute Lymphoblastic Leukemia Cancer Cells from Microscopic Biopsy Images,” *Life*, vol. 13, no. 2, Feb. 2023, doi: 10.3390/life13020348.

[65] B. Kurt and K. Turhan, “Medical Images Enhancement by using Anisotropic Filter and CLAHE.”

[66] S. S. M. Sheet, T. S. Tan, M. A. As’ari, W. H. W. Hitam, and J. S. Y. Sia, “Retinal disease identification using upgraded CLAHE filter and transfer convolution neural network,” *ICT Express*, vol. 8, no. 1, pp. 142–150, Mar. 2022, doi: 10.1016/j.icte.2021.05.002.

[67] L. Putu Diah Tri Cahyani and G. Made Arya Sasmita, “Adaptive Histogram Equalization to Increase the Percentage of Face Recognition,” 2019. [Online]. Available: www.ijcat.com

[68] A. S. Kornilov and I. V. Safonov, “An overview of watershed algorithm implementations in open source libraries,” *Journal of Imaging*, vol. 4, no. 10. MDPI, Oct. 01, 2018. doi: 10.3390/jimaging4100123.

[69] M. F. Abdullah *et al.*, “A Comparative Study of Image Segmentation Technique applied for Lung Cancer Detection,” in *2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 2019, pp. 72–77. doi: 10.1109/ICCSCE47578.2019.9068574.

[70] R. Rajakumari and L. Kalaivani, “Breast Cancer Detection and Classiication using Deeper Convolutional Neural Networks based on Wavelet Packet Decomposition Techniques Breast Cancer Detection and Classification using Deeper Convolutional Neural Networks based on Wavelet Packet Decomposition Techniques,” 2021, doi: 10.21203/rs.3.rs-405990/v1.

[71] F. Gholamiankhah *et al.*, “Automated Lung Segmentation from Computed Tomography Images of Normal and COVID-19 Pneumonia Patients,” *Iran J Med Sci*, vol. 47, no. 5, pp. 440–449, Sep. 2022, doi: 10.30476/IJMS.2022.90791.2178.

[72] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, “A Comprehensive Survey of Image Augmentation Techniques for Deep Learning,” *Pattern Recognit*, vol. 137, May 2023, doi: 10.1016/j.patcog.2023.109347.

[73] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[74] P. Garg and T. jain, “A Comparative Study on Histogram Equalization and Cumulative Histogram Equalization,” *International Journal of New Technology and Research (IJNTR)*, vol. 3, no. 2454–4116, pp. 41–43, 2017, [Online]. Available: www.ijntr.org

[75] B. Jijo and A. Mohsin Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, pp. 20–28, Jan. 2021.

[76] W. Yi, M. Lu, and Z. Liu, "Multi-valued attribute and multi-labeled data decision tree algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 67–74, 2011, doi: 10.1007/s13042-011-0015-2.

[77] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–7. doi: 10.1109/ICCCNT.2013.6726842.

[78] S. B. Kotsiantis, "Decision trees: a recent overview," *Artif Intell Rev*, vol. 39, no. 4, pp. 261–283, 2013, doi: 10.1007/s10462-011-9272-4.

[79] "Decision Tree Algorithm in Machine Learning - Javatpoint." Accessed: Aug. 22, 2023. [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[80] C. E. Shannon, "A Mathematical Theory of Communication," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001, doi: 10.1145/584091.584093.

[81] Y.-Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Arch Psychiatry*, vol. 27, pp. 130–135, Apr. 2015, doi: 10.11919/j.issn.1002-0829.215044.

[82] W. A. W. A. Bakar, N. L. N. B. Josdi, M. B. Man, and Y. S. Triana, "An Evaluation of Artificial Neural Networks and Random Forests for Heart Disease Prediction," *Journal of Hunan University Natural Sciences*, vol. 49, no. 2, pp. 41–49, Feb. 2022, doi: 10.55463/issn.1674-2974.49.2.4.

[83] G. Gao, M. Wang, H. Huang, and W. Tang, *Agricultural Irrigation Area Prediction Based on Improved Random Forest Model*. 2021. doi: 10.21203/rs.3.rs-156767/v1.

[84] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics and Proteomics*, vol. 15, no. 1. International Institute of Anticancer Research, pp. 41–51, Jan. 01, 2018. doi: 10.21873/cgp.20063.

[85] "Guide on Support Vector Machine (SVM) Algorithm." Accessed: Sep. 01, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/

[86] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mech Syst Signal Process*, vol. 21, no. 6, pp. 2560–2574, 2007, doi: https://doi.org/10.1016/j.ymssp.2006.12.007.

[87] M. Alqahtani, A. Gumaei, H. Mathkour, and M. M. Ben Ismail, "A genetic-based extreme gradient boosting model for detecting intrusions in wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 20, Oct. 2019, doi: 10.3390/s19204383.

[88] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", Accessed: Sep. 01, 2023. [Online]. Available: https://github.com/dmlc/xgboost

[89] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." [Online]. Available: https://github.com/Microsoft/LightGBM.

[90] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, and W. Zeng, "Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data," *Agric Water Manag*, vol. 225, Nov. 2019, doi: 10.1016/j.agwat.2019.105758.

[91] R. Wang, "AdaBoost for Feature Selection, Classification and Its Relation with SVM * , A Review," *Phys Procedia*, vol. 25, pp. 800–807, 2012, doi: 10.1016/j.phpro.2012.03.160.

[92] Y. Freund and R. E. Schapire, “A Short Introduction to Boosting,” *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999, Accessed: Sep. 01, 2023. [Online]. Available: www.research.att.com/

[93] T. N. Abu-, J. Samy, and S. Abu-Naser, “Classification of Sign-language Using VGG16,” *International Journal of Academic Engineering Research*, vol. 6, pp. 36–46, 2022, [Online]. Available: www.ijeais.org/ijaer

[94] “VGG-16 | CNN model - GeeksforGeeks.” Accessed: Feb. 17, 2023. [Online]. Available: https://www.geeksforgeeks.org/vgg-16-cnn-model/

[95] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition”, Accessed: Aug. 14, 2023. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[96] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION,” *ICLR*, 2015, Accessed: Aug. 14, 2023. [Online]. Available: http://www.robots.ox.ac.uk/

[97] T. H. Nguyen, T. N. Nguyen, and B. V. Ngo, “A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease,” *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Dec. 2022, doi: 10.3390/agriengineering4040056.

[98] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions”.

[99] H. Hao, Q. Wang, P. Li, and L. Zhang, “Evaluation of ground distances and features in EMD-based GMM matching for texture classification,” *Pattern Recognit*, vol. 57, pp. 152–163, 2016, doi: https://doi.org/10.1016/j.patcog.2016.03.001.

[100] P. Kuang, T. Ma, Z. Chen, and F. Li, “Image super-resolution with densely connected convolutional networks,” *Applied Intelligence*, vol. 49, Jan. 2019, doi: 10.1007/s10489-018-1234-y.

[101] W. Islam, M. Jones, R. Faiz, N. Sadeghipour, Y. Qiu, and B. Zheng, “Improving Performance of Breast Lesion Classification Using a ResNet50 Model Optimized with a Novel Attention Mechanism,” *Tomography*, vol. 8, no. 5, pp. 2411–2425, Oct. 2022, doi: 10.3390/tomography8050200.

[102] A. V. Ikechukwu, S. Murali, R. Deepu, and R. C. Shivamurthy, “ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images,” *Global Transitions Proceedings*, vol. 2, pp. 375–381, 2021, doi: 10.1016/j.gltp.2021.08.027.

[103] X. Chen *et al.*, “Application of EfficientNet-B0 and GRU-based deep learning on classifying the colposcopy diagnosis of precancerous cervical lesions,” *Cancer Med*, vol. 12, no. 7, pp. 8690–8699, Apr. 2023, doi: 10.1002/cam4.5581.

[104] T. A. Putra, S. I. Rufaida, and J. S. Leu, “Enhanced Skin Condition Prediction through Machine Learning Using Dynamic Training and Testing Augmentation,” *IEEE Access*, vol. 8, pp. 40536–40546, 2020, doi: 10.1109/ACCESS.2020.2976045.

[105] M. Tan and Q. V Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”.

[106] M. Sharma, H. Gunwant, P. Saggar, L. Gupta, and D. Gupta, “EfficientNet-B0 Model for Face Mask Detection Based on Social Information Retrieval,” *International Journal of Information System Modeling and Design*, vol. 13, no. 7, pp. 1–15, Nov. 2022, doi: 10.4018/ijismd.313444.

[107] R. Liu, D. Jiang, L. Zhang, and Z. Zhang, “Deep Depthwise Separable Convolutional Network for Change Detection in Optical Aerial Images,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 13, pp. 1109–1118, 2020, doi: 10.1109/JSTARS.2020.2974276.

[108] L. SIfre and S. Mallat, “Rigid-Motion Scattering for Texture Classification,” Mar. 2014, [Online]. Available: http://arxiv.org/abs/1403.1687
[109] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.
[110] F. Sultonov, J. H. Park, S. Yun, D. W. Lim, and J. M. Kang, “Mixer U-Net: An Improved Automatic Road Extraction from UAV Imagery,” *Applied Sciences (Switzerland)*, vol. 12, no. 4, Feb. 2022, doi: 10.3390/app12041953.
[111] “Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification) | by Sik-Ho Tsang | Towards Data Science.” Accessed: Aug. 18, 2023. [Online]. Available: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568
[112] A. Howard *et al.*, “Searching for MobileNetV3,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324. doi: 10.1109/ICCV.2019.00140.
[113] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”.
[114] L. SIfre and S. Mallat, “Rigid-Motion Scattering for Texture Classification,” Mar. 2014, [Online]. Available: http://arxiv.org/abs/1403.1687
[115] “Different Types of CNN Architectures Explained: Examples - Data Analytics.” Accessed: Sep. 01, 2023. [Online]. Available: https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/
[116] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[117] L. Cun *et al.*, “Handwritten Digit Recognition with a Back-Propagation Network.”
[118] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.
[119] “Unsupervised Feature Learning and Deep Learning Tutorial.” Accessed: Sep. 01, 2023. [Online]. Available: http://deeplearning.stanford.edu/tutorial/
[120] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.
[121] P. Liashchynskyi and P. Liashchynskyi, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS,” Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.06059
[122] “scikit-learn 1.1.1 documentation - Decision Tree classifier”, Accessed: Jul. 22, 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html?highlight=decision%20tree#sklearn.tree.DecisionTreeClassifier
[123] “scikit-learn 1.1.1 documentation - Support Vector Classification”, Accessed: Jul. 22, 2022. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[124] “sklearn.ensemble.RandomForestClassifier — scikit-learn 1.3.1 documentation.” Accessed: Sep. 29, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn-ensemble-randomforestclassifier

[125] "sklearn.ensemble.GradientBoostingClassifier — scikit-learn 1.3.1 documentation." Accessed: Sep. 29, 2023. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html
[126] "LightGBM's documentation! — LightGBM 4.0.0 documentation." Accessed: Sep. 29, 2023. [Online]. Available: https://lightgbm.readthedocs.io/en/stable/index.html
[127] Q. Ren, M. Li, and S. Han, "Tectonic discrimination of olivine in basalt using data mining techniques based on major elements: a comparative study from multiple perspectives," *Big Earth Data*, vol. 3, no. 1, pp. 8–25, Jan. 2019, doi: 10.1080/20964471.2019.1572452.
[128] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *IEEE Access*, vol. 6, pp. 33789–33795, May 2018, doi: 10.1109/ACCESS.2018.2841987.
[129] S. Badillo *et al.*, "An Introduction to Machine Learning," *Clin Pharmacol Ther*, vol. 107, no. 4, pp. 871–885, Apr. 2020, doi: 10.1002/cpt.1796.
[130] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning," *Circuits Syst Signal Process*, vol. 39, no. 2, pp. 757–775, 2020, doi: 10.1007/s00034-019-01246-3.
[131] I. Markoulidakis, I. Rallis, I. Georgoulas, G. Kopsiaftis, A. Doulamis, and N. Doulamis, "Multiclass Confusion Matrix Reduction Method and Its Application on Net Promoter Score Classification Problem," *Technologies (Basel)*, vol. 9, no. 4, 2021, doi: 10.3390/technologies9040081.
[132] M. M. Rodríguez-Hernández, R. E. Pruneda, and J. M. Rodríguez-Díaz, "Statistical analysis of the evolutive effects of language development in the resolution of mathematical problems in primary school education," *Mathematics*, vol. 9, no. 10, May 2021, doi: 10.3390/math9101081.
[133] S. T. Rikta, K. M. M. Uddin, N. Biswas, R. Mostafiz, F. Sharmin, and S. K. Dey, "XML-GBM lung: An explainable machine learning-based application for the diagnosis of lung cancer," *J Pathol Inform*, vol. 14, p. 100307, 2023, doi: https://doi.org/10.1016/j.jpi.2023.100307.
[134] S. Abunajm, N. Elsayed, Z. ElSayed, and M. Ozer, "Deep Learning Approach for Early Stage Lung Cancer Detection," Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.02456

Appendices

Arab American University
Deanship of Scientific Research
Tel: 04-241-8888, ext 1196
mail: src@aaup.edu-E

الجامعة العربية الأمريكية
عمادة البحث العلمي
تلفون: 04-241-8888, ext 1196
البريد الالكتروني: src@aaup.edu

SRD:22/23-O-065 6/8/2023

حضرة الدكتور عبد الله القواسمة المحترم
رئيس وحدة التعليم الصحي والبحث العلمي في وزارة الصحة الفلسطينية

تحية طيبة وبعد،

الموضوع: تسهيل مهمه بحثية

تهديكم الجامعة العربية الامريكية أطيب التحيات، وبالإشارة الى الموضوع أعلاه يرجى من حضرتكم التعاون بتسهيل مهمة طالبة الماجستير في تخصص علم البيانات **"يارا زايد"** لتتمكن من الحصوص على بيانات صور الرئة المقطعية المصنفة إلى فئات سرطانية وغير سرطانية (Lung CT Images) من خلال قاعدة بيانات وزارة الصحة؛ حيث ستقوم الباحثة باستخدام تقنيات الذكاء الاصطناعي: التعلم الالي والتعلم العملق

عنوان البحث: **Lung Cancer Detection System Using Medical Images Processing, Machine and Deep learning approaches**

لا حاجة لأية بيانات أخرى مثل التفاصيل الشخصية.
نتعهد لكم أنه سيتم استخدام البيانات بشكل صارم لأغراض البحث العلمي فقط. سيتم التعامل مع أي بيانات يتم مشاركتها بمنتهى السرية ووفقًا لجميع لوائح حماية البيانات ذات الصلة.

مرفق:
المشروع البحثي

شاكرين لكم حسن تعاونكم،،،

عميد البحث العلمي
أ.د. محمد عوض

State of Palestine
Ministry of Health
Education in Health and Scientific
Research Unit

دولة فلسطين
وزارة الصحة
وحدة التعليم الصحي
والبحث العلمي

Ref.: .....
Date: .....

قسم: ...
تاريخ: ...

حضرة مدير عام الادارة العامة للمستشفيات المحترم،،،

تحية واحترام،،

الموضـــوع: تسهيل مهمة بحث

يرجى تسهيل مهمة الباحثه: يارا زايد – الجامعة العربية الامريكية ، في عمل البحث بعنوان:

Lung Cancer Detection System Using Medical Images Processing Machine and Deep Learning Approaches

من خلال السماح للباحث بالحصول على بيانات صور الرئة المقطعية المصنفة إلى فئات سرطانية وغير سرطانية (Images CT Lung) من خلال قاعدة بيانات وزارة الصحة؛ حيث ستقوم الباحثة باستخدام تقنيات الذكاء الاصطناعي: التعلم الآلي والتعلم العميق ، وذلك في:

- مستشفى الوطني

على ان يتم الالتزام باساليب واخلاقيات البحث العلمي.

على ان يتم الالتزام بجميع تعليمات واجراءات الوقاية والسلامة الصادرة عن وزارة الصحة بخصوص جائحة كورونا.

على ان يتم تزويد الوزارة بنسخة PDF من نتائج البحث.

مع الاحترام،،

رئيس وحدة التعليم الصحي والبحث العلمي

دولة فلسطين
وزارة الصحة
مكتب الوكيل المساعد لشؤون المستشفيات والطوارئ
دائرة الشؤون الادارية والمالية
وارد رقم: ٢٠٢٣/٩/٥
التاريخ: ٥/٩/٢٠٢٣

دولة فلسطين
وزارة الصحة
الإدارة العامة للمستشفيات
مستشفى الوطني الحكومي
الرقم: ٢٠٢٣/٩/٥
التاريخ: ٥/٩/٢٠٢٣

State of Palestine
Ministry of Health
Education in Health and Scientific Research Unit

دولة فلسطين
وزارة الصحة
وحدة التعليم الصحي
والبحث العلمي

Ref.: ....
Date:..................

الرقم: ٢٦٣٥/١٦٣٤/٢٠٢٤
التاريخ: ١٢/٨/٢٠٢٤

حضرة مدير عام الادارة العامة للمستشفيات المحترم،،،

تحية وأحترام..

<u>الموضـــوع: تسهيل مهمة بحث</u>

يرجى تسهيل مهمة الباحثه: **يارا زايد – الجامعة العربية الامريكية ، في عمل بحث بعنوان:**

Lung Cancer Detection System Using Medical Images Processing Machine and Deep Learning Approaches

من خلال السماح للباحث بالحصول على معلومات ، وذلك في:

- مستشفى رفيديا

على ان يتم الالتزام باساليب واخلاقيات البحث العلمي.
على ان يتم الالتزام بجميع تعليمات واجراءات الوقاية والسلامة الصادرة عن وزارة الصحة بخصوص جائحة كورونا.
على أن يتم تزويد الوزارة بنسخة PDF من نتائج البحث.

مع الاحترام..

د. عبد الله القواسمي
رئيس وحدة التعليم الصحي والبحث العلمي

نسخة: عميد البحث العلمي / الجامعة العربية الامريكية المحترم

Telfax.:09-2333901 scientificresearch.den@gmail.com تلفاكس: 09-2333901

الملخص

اكتشاف سرطان الرئة في مراحل مبكرة يمثل نقطة تحوّل بالغة الأهمية في مجال الرعاية الصحية، حيث يمنح اكتشاف هذا المرض في وقت مبكر الفرصة للاطباء لوصف العلاج الأكثر فعالية، مما يؤدي في النهاية إلى تقليل معدلات الوفيات وإنقاذ الأرواح البشرية. لقد بحث العديد من الباحثين هذا الموضوع بشكل مكثف، باستخدام نهج متعدد الجوانب لتشخيص سرطان الرئة. هذا النهج يعتمد على تحليل البيانات الديموغرافية وعوامل الخطر بالإضافة إلى فحص صور الأشعة المقطعية CT التي تشمل كل من الصدر والرئتين. إن استعمال تقنيات التعلم الآلي والتعلم العميق لفحص هذا الكم الهائل من المعلومات قد شكّل قفزة كبيرة في المجال الطبي. ومع ذلك، ما زال هناك تحديات كتيرة مثل اختيار المنهجيات الامثل في كل مرحلة من مراحل تطوير هذه الأنظمة لضمان أداء عالٍ بشكل مستمر.

في هذا السياق، تقدم هذه الرسالة نظامين مميزين، أحدهما مبني على تعلم الآلة والآخر على التعلم العميق، وكلاهما مصمم لتسهيل تشخيص سرطان الرئة باستخدام بيانات غنية بالمعلومات وصور طبية.

في مجال تعلم الآلة، كان لدينا مجموعة بيانات عالمية شاملة تضم مجموعة متنوعة من المعلومات الديموغرافية وعوامل الخطر المتعلقة بسرطان الرئة. هذه المجموعة البيانات شكلت أساسًا لتدريب العديد من نماذج تعلم الآلة، بما في ذلكXGB ، RF، SM، DT، GBM، و AdaBoostبالإضافة إلى ذلك، تم استخدام تقنيات متنوعة لمعالجة البيانات مسبقًا وقد أسفر هذا النهج عن تحقيق دقة ملحوظة ، حيث حقق نموذج XGB نسبة دقة تبلغ 99.07%.

من ناحية أخرى، بني نموذج التعلم العميق بناء على مجموعة بيانات عالمية تحتوي على صور CT scan مصنفة بدقة تحت ثلاث تصنيفات: الطبيعي، الحميد، والخبيث. خضعت هذه الصور لمجموعة من مراحل تحضير دقيقة وزيادة حجم البيانات، وتقسيم البيانات، وتدريب ثماني خوارزميات تعلم عميق، بما في ذلكCNN ، MobileNet، Xception، DenseNet، VGG16، VGG19، و .Efficient Net بالاضافة الى من استكشاف متعمق للمعاملات المختلفة للنماذج الذي سمح بانتقاء أفضل هياكل معمارية لكل نموذج. تم التوصل إلى أقصى دقة أداء باستخدام نموذجCNN ، بنسبة دقة استثنائية تبلغ 99.70% ونسبة 100% لكل من الدقة والاستدعاء و درجة F1..