# Lung Cancer Detection Using Image Processing: A Hybrid CNN-DBN Framework for Accurate and Efficient Classification

by

Arique Hussain
23341105
Rakin Abser
21241024
Rudabeh Towfique
24141228
Mohammed Wasif Islam
21201389

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

# Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**

Arique Hussain
23341105

Rakin Abser
21241024

Rudabeh Towfique
24141228

Mohammed Wasif Islam
21201389

# Approval

The thesis titled "Lung Cancer Detection Using Image Processing: A Hybrid CNN-DBN Framework for Accurate and Efficient Classification" submitted by

1. Arique Hussain (23341105)
2. Rakin Abser (21241024)
3. Rudabeh Towfique (24141228)
4. Mohammed Wasif Islam (21201389)

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 24, 2025.

**Examining Committee:**

Supervisor:
(Member)

Rafeed Rahman

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Md. Tanzim Reza

Senior Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

# Abstract

The study introduces a straightforward but powerful approach that blends today's deep-learning tools with tried-and-true image-processing tricks for spotting lung cancer on CT scans. At its heart sits CLAHE-Contrast Limited Adaptive Histogram Equalization-a workhorse in medical imaging that sharpens contrast just where small shifts in tissue density matter most for diagnosis. After that, the enhanced images are fed into a custom-built Convolutional Neural Network (CNN) that has the ability to learn strong and discriminative features with a compact architecture. These are then fed through a Deep Belief Network (DBN), made up of stacked layers of Restricted Boltzmann Machines (RBMs) that can learn hierarchical representations in an unsupervised way. These ultimate representations are then fed through classification via logistic regression, a straightforward yet effective supervised learning technique which can take advantage of the good quality of the learned embeddings. The model avoids overfitting by decoupling feature learning and decision making and employs a refinement mechanism that identifies and reprocesses edge-case test samples through the CNN and DBN recurrently for improved prediction stability. With a test accuracy of 98.63% and a FLOP count of merely 47.5 million, this model achieves a remarkable trade-off between accuracy and computational resource usage. Being highly scalable and modular, it is particularly suited for deployment in clinical settings, specifically on edge devices with limited resources. This pipeline provides a good foundation for pursuing cost-effective, high-accuracy solutions for medical imaging tasks in future studies.

# Table of Contents

**Declaration** i

**Approval** ii

**Abstract** iv

**Table of Contents** v

**List of Figures** vii

**1 Introduction** 1
- 1.1 Problem Statement . . . 1
- 1.2 Research Objective . . . 2
- 1.3 Contributions . . . 2

**2 Literature Review** 4
- 2.1 Related Works . . . 4
- 2.2 Results and Findings . . . 8
- 2.3 Drawbacks and Limitations . . . 11

**3 Work Plan** 13

**4 Preliminary Analysis** 14
- 4.1 Data Collection . . . 14
- 4.2 The CNN Model . . . 15
- 4.3 Initial Processing . . . 16
- 4.4 Feature Enhancement . . . 17
- 4.5 Creation and Application of the Tested CNN Model . . . 19
- 4.6 Testing . . . 20
- 4.7 Initial Results . . . 21

**5 Methodology** 23
- 5.1 Description of the Hybrid Model . . . 23
- 5.2 Image Preprocessing using CLAHE . . . 25
- 5.3 Feature Extraction using CNN . . . 28
- 5.4 Deep Belief Network . . . 30
- 5.5 Refinement . . . 33
- 5.6 Classification . . . 33

**6 Results, Analysis and Comparisons** **37**
6.1 Results . . . 37
6.2 Importance of Refinement . . . 39
6.3 Integrity . . . 39
6.4 Training Diagnostics . . . 40
6.5 Gaussian Blur + SVM . . . 42
6.6 Ensemble Architecture . . . 42
6.7 FLOPs Count . . . 43
6.8 Comparisons Summary . . . 44
6.9 Comparison with Literature . . . 45

**7 Challenges, Future Works and Conclusion** **47**
7.1 Challenges . . . 47
7.2 Future Works . . . 49
7.3 Conclusion . . . 50

**Bibliography** **54**

# List of Figures


3.1 Work Flow Diagram . . . . . 13

4.1 Cancerous and Non-Cancerous CT scans . . . . . 14
4.2 CNN Architecture . . . . . 15
4.3 Application of the covulation layer . . . . . 16
4.4 CT scan after undergoing initial processing . . . . . 17
4.5 CT scan after further feature enhancement . . . . . 18
4.6 Cancerous and Non-Cancerous Predictions . . . . . 21
4.7 Visualization of Accuracies and Losses . . . . . 21

5.1 The pipeline . . . . . 24
5.2 CLAHE . . . . . 26
5.3 CLAHE . . . . . 26
5.4 Deep Belief Network . . . . . 30
5.5 Logistic Regression . . . . . 36

6.1 Confusion Matrix . . . . . 38
6.2 Reconstruction loss . . . . . 41

# Chapter 1

# Introduction

Lung cancer, also known as carcinoma is the formation of lung nodules which is caused by growth of uncontrolled cells in lung tissues. Following heart attacks, lung cancer sickness poses the second greatest threat to global health since it is the type of cancer that kills more people than all other types of cancers combined. [12]. Tobacco consumption, smoking along with the increase of vaping in recent years are the main causes of these growing lung nodules which eventually lead to death. Lung cancer has led to so many unfortunate events of death due to the detection of this type of cancer in a very later stage. If the diagnosis of lung cancer is done in an earlier stage, then it increases the chances of survival for the patient up to 50-70%. Small cell lung cancer and non-small cell lung cancer are the two main forms of lung cancer. About 80–85% of instances of lung cancer are non-small cell lung cancer, with small cell lung cancer accounting for the remaining occurrences. The cancer is made up of 4 important stages where stage I is limited to the lung area, stage II and III is around the chest area and the final one which is stage IV-here it spreads to other parts from the chest and there is almost no return for a patient from this stage. As a result, to lower the number of deaths, we want to detect the cancer in the early stages with techniques such as image processing with the integration of machine learning, neural networks and artificial intelligence. The previous processes were slow in detecting the cancerous cells so the inclusion of image processing makes it faster and increases the chances of survival. Through image processing, a picture is transformed into a digital format so that it can be used to create a useful image or extract relevant data. This type of signal processing involves an input that is an image, such as a picture or a video frame, and the output can be a picture or traits connected to the picture [9]. Computerized Tomography (CT) scans are the main way to detect cancer image reports for lung cancers in the medical field to increase chances of survivable through early detection.

## 1.1 Problem Statement

Cancer is one of the deadliest diseases in the world, with the number of cases rising at a very alarming rate. There are many systems to aid in the early detection of cancerous cells in patients such as computer-aided diagnostic (CAD) systems and Machine Learning (ML) models. However, the issue of how accurately these models can detect these cancerous cells still remains, with most models still giving false positive and true negative results. Hence, this paper aims to explore multiple Ma-

chine Learning models, with lung cancer being the area of focus, and evaluate and compare how these models perform against each other, discovering common drawbacks and problems, and possibly implementing a new Machine Learning technique, by, both, developing new techniques and implementing previous and existing models.

One of the problems that exist within the existing cancer detection models is the high computational cost and tend to be too resource-heavy for practical deployment. Also, the problem with most supervised learning models is that they tend to rely on the memorization rather than focusing on generalization, which increases the risk of over-fitting.

## 1.2 Research Objective

This study aims to conduct a comprehensive review of advanced lung cancer detection techniques, focusing on the role of image processing, Artificial Intelligence (AI), Machine Learning (ML), and Neural Networks (NNs) in improving early diagnosis and classification accuracy. By analyzing existing methodologies, this research seeks to identify key challenges, assess the effectiveness of various computational models, and explore potential enhancements in automated detection systems. Furthermore, the study aims to bridge the gap between AI-driven research and real-world clinical applications, addressing critical issues such as false positives, computational efficiency, and dataset limitations. Ultimately, this research aspires to contribute to the development of more precise, cost-effective, and accessible diagnostic solutions for lung cancer, thereby improving patient prognosis and survival rates.

We want to construct an intelligent framework, which addresses the problem of accuracies, but also mitigates the problem of memorization and overfitting, whilst also being lightweight and efficient enough to be deployed in real-world settings in order for the accurate and efficient detection of lung cancer in patients.

## 1.3 Contributions

In this thesis, a distinct method of lung cancer classification using CT scans is presented, combining classical image enhancement techniques with deep learning and lightweight statistical methods. The model was designed to be efficient, adaptable, and generalizable in real medical environments, rather than solely maximizing accuracy. The key contributions of our work are as follows:

• Our new hybrid classification pipeline combines the role of pre-processing CT images using the CLAHE procedure and feature extraction using CNN, hierarchical representation learning using DBN, and a logistic regression classifier.

• An adaptive corrective mechanism is proposed, allowing the model to revisit and recompute difficult test samples without retraining, proficiently making a dynamic corrective process for edge cases.

• Contrast enhancement, via CLAHE, is better at showing subtle features com-

pared to other methods such as Gaussian blur or Gabor filters in terms of clarity and classification accuracy.

• This pipeline is associated with a high testing accuracy of 98.63% and can be characterized as operationally efficient due to a substantially reduced number of FLOPs previously required during the processing of data of other deep models.

• We demonstrate the effectiveness of this pipeline across different configurations and conclude that it generalizes better than ensemble architectures or single supervised classifiers.

In summary, the given work provides an effective, interpretable, and resource-conscious model of lung cancer detection. Its modularity, readiness to consider edge-case samples, and low memory requirements make it an interesting offering to deploy in the field in early detection cases.

# Chapter 2

# Literature Review

Recent advancements in image processing techniques have significantly contributed to various medical areas - especially in the early detection of ailments and their treatment. The list includes conditions where time is extremely crucial for identifying abnormalities, such as cancerous tumors in any part of the body like the lungs and breasts. Given the significance of such conditions, the quality and accuracy of images are critical in this research.

## 2.1 Related Works

According to Mokhled [4], image enhancement techniques are broadly classified into methods based on spatial domain and frequency domain. Three techniques, namely, Fast Fourier transform, Gabor filter, auto-enhancement were used at the image enhancement stage with the Gabor filter method resulting in a 38.025% enhancement, and the Fast Fourier transformation technique achieving 27.51% were used in the paper. Based upon the results, Gabor filter was concluded to be the most suitable method for image enhancement as it produces images with more details, clarity and brightness. Segmentation mainly relies on two intensity values; similarity and discontinuity. For discontinuity, an image is partitioned by detecting high intensity changes in places like borders and edges. The latter approach divides the image into regions according to similarities based on predefined criteria. In the paper, two methods based on the second approach were used - Histogram thresholding, and Marker-Controlled Watershed Segmentation with them resulting in accuracies of 81.835% and 85.165% respectively. Therefore, the Marker-Controlled Watershed Segmentation method proved more effective for image segmentation. To assess the likelihood of lung cancer, two feature extraction methods, binarization and masking, are used. These techniques, based on the lung's anatomical features and CT imaging data, when combined, enables the determination of the normality of a case by comparing it to a defined benchmark of normality and abnormality.

A noteworthy study conducted by Rebecca L et al.[16], proposed a 3D probabilistic deep learning system for detecting lung cancer using low dose CT scans. As per S.Narvekar et al.[20]their method, based on V-Net structure techniques, achieved an accuracy of 96.5%, demonstrating high reliability in early lung cancer detection. The paper also gives us a table listing thirty methods and their merits, demerits, datasets used, rate (False Positive Rate/ False Negative Rate/ True Positive Rate/

True Negative Rate), tools used and their results, which is beneficial for anyone who is interested to further develop this study. The paper, however, was not free of challenges. From finding the right tone to selecting relevant papers for the survey, merging various approaches from the surveyed publications into a cohesive, generalized view to determining an appropriate methodologies. Evaluating the strengths and limitations of similar studies, had also consumed significant time as well. The above mentioned framework can then be applied using segmentation techniques in CET, PET and MRI. To get more accurate lung lesion detection, methods such as p-tile and binary morphology can be used, methods like Sobel and Canny can be used for improvements in segmentation stage.

As per Disha Sharma [2], to develop a CAD system, the first step is to pre-process CT images to reduce noise and in return enhance the image quality. The two main pre-processing techniques mentioned are Wiener filter and denoising. Denoising helps reduce background noise in images using Gaussian or anisotropic filtering. The Wiener filter is then used to enhance the image quality by reducing mean square errors. The paper then talks about the use of image preprocessing techniques like Erosion, median filter and dilation to identify and extract the suspected region so it can then be processed further. Once the image has been pre-processed, segmentation is applied to isolate the lung region from other structures such as the diaphragm, ribs and blood vessels. This paper talks about the use of segmentation technique based on the "region growing" method to assign labels to pixels which share certain visual characteristics, after which Sobel edge detection is applied to delineate boundaries and allows more accurate segmentation of the lung region.

Once segmentation is completed, this paper talks about extracting features from the images to identify specific characteristics of the lung nodule. Some important features outlined by the paper are: Area, Shape, Calcification, Size and contrast Enhancement. The paper says that, from study and experimentation, patterns such as laminated or popcorn calcification suggest benign nodules. Furthermore nodules smaller than 3 cm can be either benign or malignant. and contrast enhancement less than 15 Hounsfield units strongly indicates benignity, warranting further investigation.

Chaudhury and Singh [5] used smoothing techniques by utilizing Gaussian filters and enhanced the images using Gabor filters in their pre-processing stage. As a result, the output images have greater clarity. They also used Fast Fourier Transform and Auto-Enhancement methods for the pre-processing parts. For image segmentation, they used the Thresholding and Watershed method which converts the pre-processed grayscale images to binary images. The main purpose of these techniques is to be able to differentiate between the cancerous nodules and non-cancerous nodules. The method also helps improve detection accuracy by separating overlapping nodules. When compared to the traditional Watershed method, theirs showed more accuracy. They mainly extracted features such as average intensity, area, perimeter and eccentricity. Their main focus and purpose was to portray how their feature extraction method helped identify the different lung cancer stages. By analyzing the CT images, we can identify the different stages where first stage, the cancer is only situated in the lung. In second and third stages, the cancer had spread over the chest

area, and finally in the last stage, stage IV, the cancer had spread to other parts of the body. The conclusion that was reached in this paper was that by enhancing the images and improving the image quality, we could identify the cancerous nodules faster visually and come to an early diagnosis.

Bhuvaneswari and Therese [6] proposed a method of detecting lung cancer in its early stages by integrating Artificial Intelligence algorithms such as Genetic Algorithm and machine learning tools such as K-Nearest Neighbor (K-NN) in their methodology. Initially, in the pre-processing stage, they converted the images to grayscale and enhanced it to increase its contrast for better identification. Next a Gabor filter was used to enhance the features and increase its texture analysis. [6] details how the Gabor filter can work in both the spatial and frequency domain. The paper explains how its impulse response is defined by a sinusoidal wave, this means that it can identify patterns such as edges and lines. This sinusoidal wave is then multiplied by a Gaussian function. This mainly softens out the features of the image and gives out a smooth output. The Fourier transform helps to transform the image into frequencies. Overall, the Gabor filter has very helpful applications for feature detection and analyzing the textures of an image. For classification, KNN and genetic algorithms were used. KNN is a non-parametric method used for classification and regression. The input consisted of a K number of training examples in the feature space. The closer neighbors determined the class of the sample and the distance was measured in euclidean metrics. Genetic algorithm was used to select the best K-values and sample features to provide high classification accuracy. This was done using genetic algorithm's basic steps which include, selection, crossover and mutation and assigning a fitness function to select the fittest chromosomes. Of their five test samples, the highest detection accuracy for a cancerous sample was 89% and 90% for a non-cancerous sample.

Ashwini Rejintal, Aswini N.[10] state several studies on the detection and classification of lung cancer have been conducted with MATLAB software and a variety of image processing techniques. The ideal tool for creating algorithms is MATLAB however, because of its slowness, you will need to adapt your algorithm to an object-oriented programming language in order to implement it. Thus, the entire process takes longer. According to Rahane and colleagues [13] a large and accessible repository with an effective data search engine implemented in the system can address the shortage of some of the classification and pattern recognition algorithms that were still experiencing issues with redundancy criteria for a large number of histopathological datasets. Abdul Yousuf Mohammed [8] states that a few of the systems suggested to use artificial neural networks (ANNs) to identify lung cancer give erroneous results. For abstract issues like picture recognition, it is quite easy to use and works excellent, but even a small increase in accuracy can have a significant impact on the scale. Khosravi et al.[11] writes Deep learning training is extremely sluggish, and it gets even slower when new features are added. In contrast, the linear model trains significantly more quickly. Magnetic Resonance Imaging (MRI), according to Dhanesh D. Lokhande [7], produces incredibly detailed images of body structures. Similar to an MRI, a computed tomography (CT) scan is a more sophisticated and potent type of x-ray that provides 360-degree views of the spine and internal organs.

According to [8], lung cancer is the growth of uncontrolled cells in the lung region and turning into a tumor. Cancerous cells are transported from the lungs via lymph fluid or blood. The artificial neural network(ANN) based classification and detection system of lung cancer is a conceptual system and has a very poor accuracy rate. Computer-aided diagnosis in chest radiography classifies the lung extraction into two different groups one being rule based and the other pixel classification. Another technique named local density maximum algorithm which detects small lung nodules on computed tomography(CT) also provides low detection accuracy. CAD is a technique which is classified into two categories consisting of model-based and density-based approaches. Carcinoma Identification while early detection and prediction of lung cancer survival utilizing neural network classifiers and image processing techniques have been established, their identification and detection capabilities are limited. Two segmentation techniques are presented for the diagnosis of lung cancer utilizing artificial neural networks and fuzzy clustering approaches. In Lung Cancer Detection using Curvelet Transform and Neural Network, a novel approach to LCD detection is proposed. The curvelet transform successfully recovers the features of CT scan images of lung cancer. The most current research on the subject of lung cancer detection was just finished. These investigations include Automatic Lung Cancer Detection, Bayesian Classifier-Based Lung Cancer Detection and Classification, and Machine Learning-Based Multinomial Bayesian Lung Cancer Detection and classification. The research focuses on several methods of detecting lung cancer: automatic detection in CT images, detection of lung cancer using SVM and BPNN and estimtes the size of the cancerous cells. This method uses back propagation and image segmentation, segmentation technique based on gray coefficient mass estimation to detect cancerous cells using Gabor filters.

Vikul Pawar[17] discusses in his paper, the methods for lung disease diagnosis(LDD) using CT-scan and X-ray images. He proposes image processing techniques such as enhancement, image acquisition, feature extraction, segmentation and classification are used to detect lung cancer using CT and X-ray images. Each of the stages contributes to detecting lung abnormalities such as cancerous nodules.

For image acquisition, the input CT and X-ray images are acquired with an initial noise, for example; Gaussian noise, which is then removed using Spatial filtering. Then using techniques like the Laplacian operator for edge enhancement. These enhanced images are then processed by converting them into grayscale to make sure the image segmentation results are accurate. Image segmentation is critical as the image is being divided into regions in order to find potential cancerous zones. Methods like thresholding, Otsu's methods are used in order to distinguish among the pixel groups which allows a better identification of cancerous and non-cancerous regions. Features like Histogram of oriented gradients(HOGs), statistical properties and texture characteristics from gray level co occurrence matrices are used to assist in the prediction in feature extraction in order to extract information from the segmented images. Finally, using machine learning algorithms such as Support Vector Machines (SVM), Decision Trees, K-Nearest Neighbors (K-NN) and Artificial Neural Networks (ANN) are used to classify if the tumors are benign, mild or malignant.

A computer-Assisted Diagnosis(CAD) model ties it together to identify lung nodules

using methods like E-CNN. For a more accurate diagnosis, LIDC-IDRI dataset is used for testing which basically has low dose CT images of lung cancer patients.

Bari et al.[14] focuses on image pre-processing, image enhancement, segmentation, feature extraction and classification, During pre-processing, the author specifies that the goal is to make the images more suitable for further processing by removing noises like Gaussian noise, Salt and Pepper, Poisson and speckle noise, Filters like Wiener filter and non linear filters are used in order to do that. Here, the Gabor filter is used for texture analysis to identify lung cancer cells. Techniques such as manipulating pixel values and frequency domain methods such as analyzing signals and frequencies make it easier to extract meaningful information. The image is then divided into relevant and non relevant parts for which methods like thresholding, region growing and clustering based segmentation is used. Edge based segmentation and morphological operations are used to improve accuracy. The paper also suggests supervised and unsupervised learning techniques for classifying tumors. K-Nearest Neighbors (KNN), Artificial Neural Networks (ANN), and Support Vector Machines (SVM) are some of the supervised techniques, whereas k-means clustering and Principal Component Analysis (PCA) are some of the unsupervised learning techniques which can work without predefined labels.

Lakshmi Hanisha Jyothula et al.[21] uses machine learning to differentiate between malignant and benign lung cancer cells. The CT images are preprocessed with a median filter to reduce noise while preserving edges. Regions of interest are isolated during segmentation and area, perimeter, compactness, circularity and Euler number are then extracted to help classify the images.

## 2.2 Results and Findings

Work published in [9] aims to lessen the false positive results which has an accuracy of 87.82%. Supervied Machine Learning (SVM) has classifiers which determine the nodules and non-nodules using pre-processed images. An automatic CAD system which provides accuracy of 80% by using the help of lung CT images for early detection. In [1] mainly two steps make up a computerized system for identifying cancerous in CT scan images, the first stage being lung segmentation and enhancement while the second is feature extraction and classification. Threshold segmentation is used to eliminate background and extract the nodules from a picture. To classify potentially aberrant locations, a neural fuzzy classifier computes a feature vector for those regions. By doing this, a 95% accuracy rate was attained.

Lakshmi Hanisha Jyothula et al.[21] also suggested two machine learning models, K-Nearest Neighbour (K-NN) and Decision Tree(DT). The accuracy achieved by K-NN is 94.44% which is slightly higher than that of using DT, 93.88%. The trained models can later be used to test new images. However, this shows how the K-NN approach can be used to create a reliable model to use for lung cancer detection with highest accuracy.

One of the top findings from[4] was that the Gabor Filters achieved an accuracy percentage of 38.025%, which was enough to outperform the other two image en-

hancement techniques, Auto-enhancement and Fast Fourier Transformation. It was also found out that the Marker-Controlled Watershed Segmentation method was more accurate, compared to thresholding, resulting in an accuracy of 85.165%, showing that this method was good for isolating the lung regions present in the CT scan images. Moreover using the processes of binarization and masking, the system talked about in this paper was able to achieve a true acceptance rate of 92.86% and a False Acceptance Rate of 7.14%.

Sharma and Jindal [3] talks about creating an automated Computer-Aided Diagnosis system to help in the early detection of cancerous cells in the lungs. One of the key findings from this paper was that the CAD system proposed in this paper had a sensitivity percentage of 90% and a False Acceptance Rate of 5%, which may be used to detect nodules as small as having a diameter of 3mm, enabling early detection of cancerous cells compared to an experienced radiologist. Another feature talked about in this paper was the use of Wiener filter technique to de-noise the CT scans to make them more clear and reduce as much noise as possible to increase the accuracy of the cancer detection.

One of the most important discoveries from[20] is the impressive accuracy of 3D probabilistic deep learning; this can be used to identify lung cancer from low-dose CT scans. Accuracy rate was observed to be 96.5%. This proves the potential of deep learning techniques for early stage detection of cancerous cells in the lungs.. Moreover, use of multi-class SVM classifiers for multi-stage lung cancer has shown an accuracy of 97% for identification of cancer and 97% for prediction. This demonstrates the effectiveness of using machine learning models in cancer diagnosis. Methods such as deep neural networks for nodule detection have also displayed impressive results; an accuracy of 97%, specificity of 97.85% and sensitivity of 96.26% have been observed. There results reinforce the effectiveness of deep learning in identifying lung nodules. In addition to that, techniques such as improved profuse clusturing, combined with the former, achieved 98.42% accuracy, with a minimal classification error of 0.038. this also proves the potential of clustering techniques in cancer diagnoses. Besides, advancements in segmentation techniques have also shown potential. The application of watershed segmentation in multiples studies have demonstrated its effectiveness in distinguishing lung cancer cells from CT scan images, with an accuracy of as high as 94.3%. Large datasets like LIDC/IDRI, LUNA16 and the cancer imaging archive (CIA) have been utilized in several approaches in the survey, ensuring the robustness of the techniques discussed across a variety of datasets and improving model generalization.

This research by [19] presents an advanced and comprehensive framework for the classification of lung cancer using Computed Tomography (CT) images. The main goal is the accurate distinction between benign and malignant pulmonary nodules which is of utmost significance for early diagnosis and treatment planning in patients with lung cancer, the most prevalent reason for cancer related deaths globally. The proposed method leverages the strengths of a fusion of traditional image processing techniques, deep learning and a new optimization algorithm to enhance the accuracy of classification.

The research starts by CT image feature extraction using texture and image processing methods. The Gray-Level Co-occurrence Matrix (GLCM) is used to measure texture features including contrast, correlation, homogeneity, energy(entropy) and temperature. The features have been effective in the discrimination of high metastatic and low metastatic cancer cells. In addition Local Binary Patterns(LBP) is used to extract micro-patterns from the image texture and wavelet filters are used to decompose the images and remove noise, extracting meaningful features at different frequency levels.

To deal with the high dimensionality of the features extracted and keep the most significant data the Local Tangent Space Alignment(LTSA) technique is used. LTSA helps to map the high dimensional data to a lower dimensional space in a manner that maintains the geometric structure of the data. Dimensionality reduction is required to enhance the performance and efficiency of the classification models.

A Deep Belief Network(DBN) is then used as the main classification model. DBNs are an unsupervised deep learning structure that has the capability to automatically learn abstract and complex features from the input data. They are highly appropriate for medical image analysis as they can learn hierarchical representations. In this work DBNs are used for the classification of the lung nodules based on the features extracted with the help of the CT images.

To optimize the performance of the DBN for higher classification accuracy the study proposes a novel meta heuristic algorithm known as the Opposition-Based Pity Beetle Algorithm(OPBA). OPBA is a nature inspired algorithm derived from the foraging behavior of the six toothed spruce bark beetle in which the beetle searches and discovers food and nesting sites on trees. The foraging behavior of the beetle is mimicked in the algorithm to efficiently explore and exploit the search space. OPBA is a purpose-oriented algorithm that simulates foraging behavior of the six toothed spruce bark beetle in finding food and nesting location on trees. The foraging process of the beetle is emulated in the algorithm to efficiently search and reap the search space. OPBA improves the conventional Pity Beetle Algorithm (PBA) with three strategies: Opposition-Based Learning (OBL), Position Clamping (PC), and Cauchy Mutation (CM). OBL enhances convergence rate by obtaining conflicting solutions at a time. PC keeps candidate solutions within feasible ranges and CM produces diversity for avoiding premature convergence.

The above framework was implemented on the LIDC-IDRI dataset, a benchmarking lung imaging dataset. Experimental results indicate that the above method is extremely accurate(97.92%), specific(97.24%) and sensitive(96.86%). These values of performance are much better than the traditional machine learning algorithms such as Artificial Neural Networks(ANN), Support Vector Machines(SVM) and K-Nearest Neighbors(KNN) and thus decide the viability of the proposed hybrid DBN and OPBA solution.

The article proposed a robust fully automatic lung cancer classification of CT images. According to the application of traditional texture analysis, deep feature learning and strong optimization technology the system provides a high speed solution to

early diagnosis of lung cancer. In addition to improving the accuracy of diagnosis the approach has great potential in clinical applications of computer-aided diagnosis(CAD) systems.

Abdus Saboor et al. [22] created a hybrid deep learning model that blends GRU and CNN which was known as the CNN-GRU model. They were combined in order to combine the best aspects of both approaches. While GRU is better at identifying patterns over time, CNN excels at identifying visual features from CT scan images. Together they make a great tool for detecting lung cancer. A publicly accessible dataset of 364 lung CT scans from an Iranian hospital was used to test their model. Of these, 126 images from patients with other lung conditions like COVID-19 were non-cancerous, and 238 images revealed lung cancer. The dataset is a strong basis for testing because medical professionals checked the labels to guarantee accuracy. Using a technique known as holdout validation, the team divided the data into 70% for model training and 30% for model testing. They used Python and well known tools such as TensorFlow, Keras and Scikit-learn to train the model. They adjusted parameters such as the batch size to 120, the number of training rounds to 60, and two distinct learning rates which were 0.001 and 0.0001 in order to achieve the best results using a technique called the Adam optimizer to better their performance. A number of important metrics were used to assess the CNN-GRU model's performance such as accuracy, precision, sensitivity, specificity, F1-score which is a balance between precision and sensitivity and the area under the ROC curve which signifies a measure of overall performance. With a learning rate of 0.001, the model gave the highest accuracy, attaining 90.56% accuracy, 98.34% sensitivity, 95.51% specificity, 89.49% precision and 91.65% F1-score. However the performance declined when the learning rate was reduced to 0.0001. The model effectively distinguishes between cancerous and non-cancerous lung images by combining CNN and GRU. Hence, providing hope for earlier and more precise diagnoses. According to the researchers, this model has the potential to revolutionise digital healthcare systems by assisting physicians in early detection of lung cancer.

## 2.3 Drawbacks and Limitations

However, there is still room to improve the segmentation process which may lead to higher accuracy, even though Marker-Controlled Watershed Segmentation method had achieved an accuracy of 85.165% , an accuracy rate of above 90% would seem more acceptable. The paper [4] also had a heavy reliability on the Gabor Filter process, which even though performed well with the dataset talked about in this paper, the accuracy may vary if other types of images or datasets are used. Lastly, there is still the issue of the False Acceptance Rate being 7.14%, which may not be a huge value, however the chances of a non cancerous nodule being diagnosed as being cancerous is still higher compared to the False Acceptance Rate being around 1% or 2%

Despite the high sensitivity rate in [3], there is still a 5% False Acceptance Rate, which as mentioned earlier, would still show that there is a cancerous nodule present where in reality the case is normal, hence it is crucial to keep the False Acceptance Rate as low as possible, such as 1% or 2%. Moreover, even though this system is able to detect cancerous cells as small as 3mm, it fails to detect cancerous cells

smaller than 3mm, which would have been more beneficial, aiding in an even earlier detection of lung cancer. Lastly, the system was only tested on the datasets from NIH/NCI Lung Image Database Consortium (LIDC), which leaves a chance of varying results if applied to other datasets or in a real life scenario.

The 92% accuracy rate attained while combining artificial neural networks and Haralick feature extraction with AI showed the importance of feature extraction techniques in[20]. However, a number of drawbacks have also been taken into account. To begin with, even though the accuracy rates were high, a few of the models had high false positive rates. For example, in one of the approaches, the false positive rate was observed to be 19.1 per scan, which may cause patients to undergo needless follow-up medical procedures. Moreover, many deep learning models consume a significant amount of processing time and specialized hardware, resulting in them being computationally intensive; they might not be suitable for adaptation in all clinical environments. Moreover, certain segmentation techniques, such as watershed segmentation have been seen to cause over-segmentation; this might lead to misdiagnosis by classification of healthy tissues as cancerous. Some of these deep learning models are also sensitive to noise, which has the potential to cause incorrect classifications, especially during use in low-dose CT scans. On top of that, most of the approaches discussed in the paper are focused on CT scans; there is inadequate discussion on other imaging modalities like PET or MRI. This has the potential to restrict their applications across a broad range of medical imaging techniques.

# Chapter 3

# Work Plan

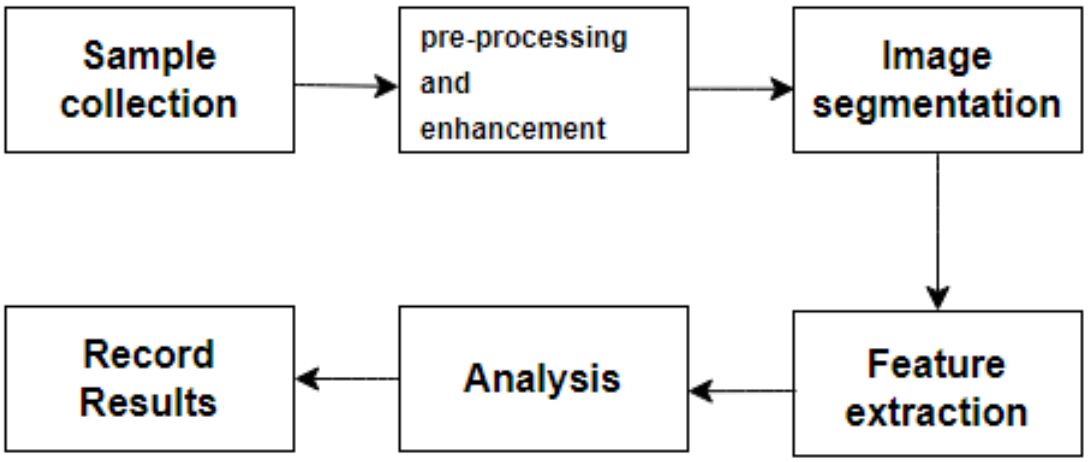



Figure 3.1: Work Flow Diagram

Our plan is to initially collect samples from available resources. We will at first pre-process the images to enhance its features for greater clarity. The sample will then be segmented so that we can distinguish between cancerous and non-cancerous nodules, and prepare it for feature extraction and analysis. We will extract important features like area, perimeter, intensity etc. We plan to apply various learning models and compare the results we obtain. Based on our results we will choose our preferred method and the best model, which will give the best output and highest accuracy.

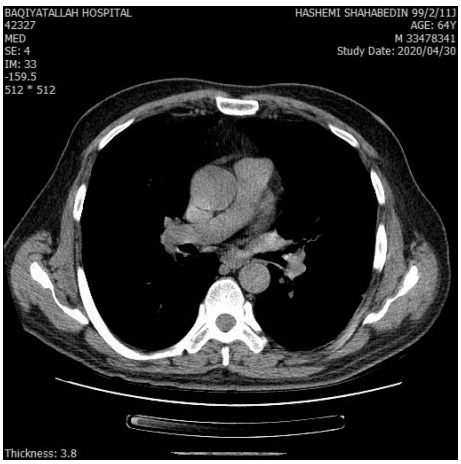
# Chapter 4

# Preliminary Analysis

## 4.1 Data Collection

The dataset we used for our thesis was uploaded by (Negar, 2020) [15], a Data Scientist based in the University of South Florida, on mendeley. The dataset consists of images of CT-Scans collected from a hospital in Iran. The images were made available in a zip folder, in which there were two more folders classified as "cancerous" and "non-cancerous". As the names suggest, inside the cancerous folder, the CT-Scan images belonged to patients who had lung cancer. On the other hand, the non-cancerous folder consisted of CT-Scan images belonging to patients who suffered from other lung related diseases such as COVID-19.

Cancerous CT scan




Non-Cancerous CT scan

Figure 4.1: Cancerous and Non-Cancerous CT scans

In order to avoid a probable error in the classification of these images, they were collected and classified with the help of a pulmonologist.

## 4.2 The CNN Model

Our first plan was to test the dataset by applying the Convolutional Neural Networks (CNN) model on it. The CNN model is very popular in the field of Computer Vision and Image Processing as it can be used to comprehend image data. In an article written by (Manav, 2024)[18], he fully described how a CNN model works.

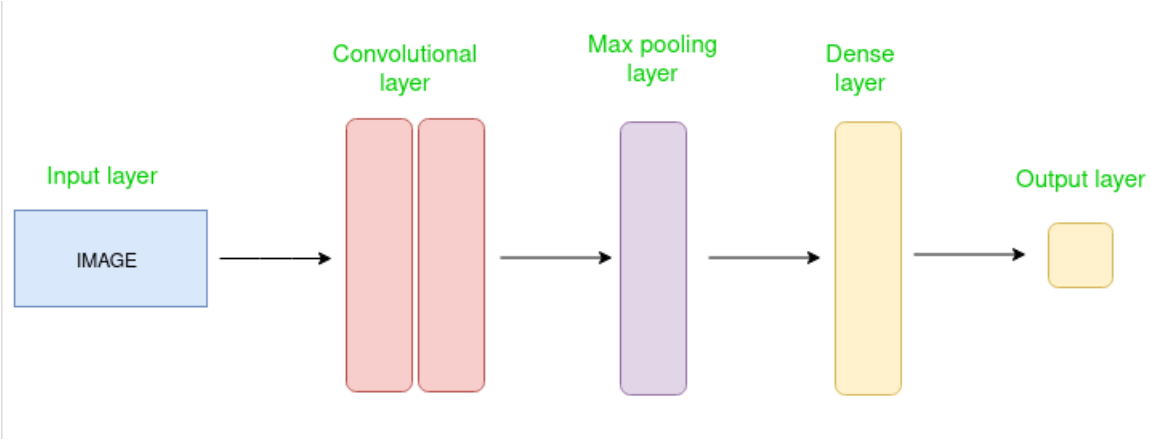



Figure 4.2: CNN Architecture
[24]

At first we should cover the basics of the matter. An RGB image basically contains pixel values in three planes, (red, green and blue). On the other hand, a grayscale image is identical, however the pixels are only present in one plane. Attached below is a visualization of the CNN Architecture.

1. Input Layer
This is where we insert the image data into our model. This layer contains the raw images. In our case, we inserted a zip file containing two different folders of cancerous and non-cancerous images.

2. Convolutional Layer
In this layer, features from the image dataset are extracted. We apply filters/kernels which have smaller heights and widths compared to the original image, but must have the same depth. More number of convolutional layers, result in more effective feature extraction.

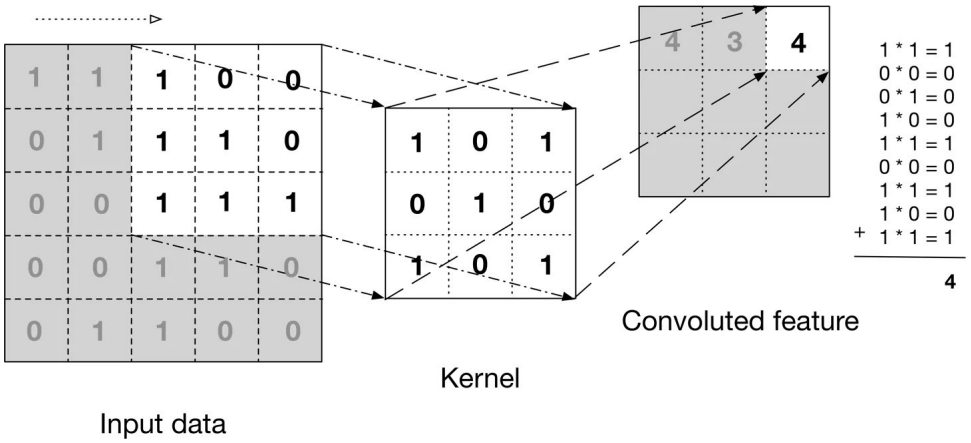



Figure 4.3: Application of the covulation layer
[18]

This is a visual representation of a convolution. In the image above, a 3x3 matrix (filter/kernel) is applied to the image to obtain a convolved feature, which is then passed on to the next layer.

3. Pooling Layer
The role of this layer is to reduce the size of the volume to decrease computational time. It also reduces memory usage and prevents overfitting. There are two types of pooling, max pooling and average pooling. In our model, we used the max pooling function. Here, the maximum value is selected and the rest are discarded. This is useful in retaining important features, which may represent significant patterns.

4. Dense Layer
The dense layer is the final layer of the model. The neurons in this layer receive inputs from its previous layers, and based on the features extracted and analyzed in the previous layers, it makes predictions and decisions.

## 4.3 Initial Processing

Initially, the zip file containing the dataset was uploaded onto Colab and was saved in a dictionary named "uploaded," in which the keys of the dictionary contain the name of the file that was uploaded, and the values are the raw binary code of the contents inside the file. Once the zip folder has been uploaded, all the contents of the zip folder, i.e., the "cancerous" and "non-cancerous" folders and all the content inside of them, are saved in a Colab directory named "lung_images." Since the zip folder was already categorised, the directory "lung_images" is also going to be categorized, that is, the "lung_imaged" directory will contain the "cancerous" and "non-cancerous" folders along with their respective contents. This initial step allows the dataset to be ready for further processing.

Once the dataset has been uploaded, the images would need to be preprocessed in order to ensure that the data set is in the desired format for the CNN model. Hence the path of the two folders, that is, "cancerous" and "non-cancerous" is saved in "cancerous_dir" and "non_cancerous_dir" and the initial processing begins. Only

the images files in the formats JPG, JPEG and PNG are accepted, after which each image is read and checked to ensure that it is not corrupted, and if an image fails to load, it is skipped over. Additionally, each image is checked to see whether or not it is a completely black image, as that might indicate that there is an issue in the dataset, hence those images will be flagged. Once all the images have been verified, each image is then resized to 224x224 pixels to ensure that there is consistency in the dimensions for the CNN model, after which the values of the pixels are normalized by scaling them between 0 and 1 with the hope of improving performance and training speed. Finally, once all the images have been processed, it is saved in a numpy array called "images". This preprocessing of all the images ensures that our dataset is clean and uniform and is ready for our CNN model.

Once this initial processing was completed, a random image from the cancerous section was selected and was converted into the NumPy array format and the entire array was checked to determine whether or not our processing techniques were actually applied correctly to all images or not. After observation, it was determined that the image was correctly resized to 224x224 format and was normalized.

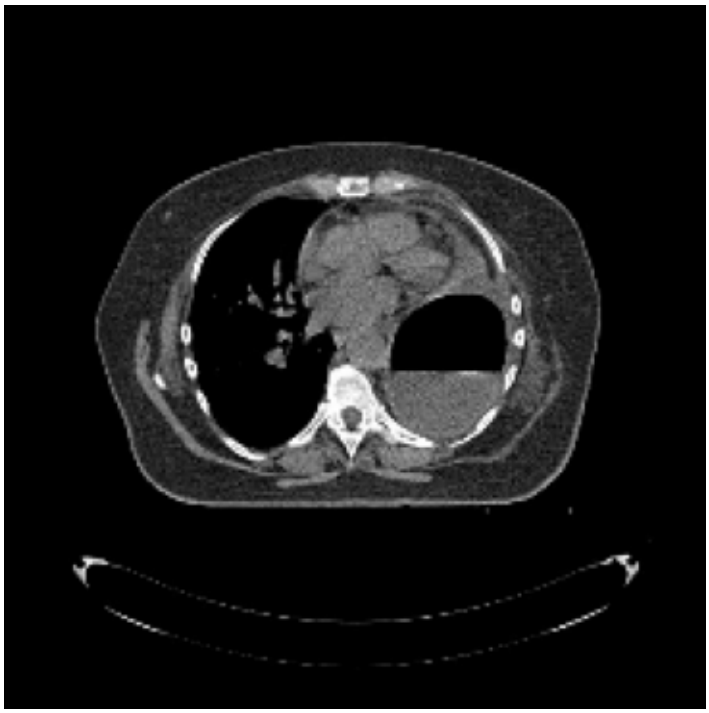


Figure 4.4: CT scan after undergoing initial processing

## 4.4 Feature Enhancement

Once the initial processing is completed, a few other feature enhancements are made to the image in order to achieve better results and speed up the training process. Firstly, the selected image, which had been normalised before, is multiplied by 255 in order to up scale it to a value between 0 and 255. The image is then converted to 8-bit integer format and is then converted from RGB to Grayscale as the image be-

ing in Grayscale would make it easier to apply thresholding for segmentation. After this, thresholding is applied to the image by increasing the intensity of pixels, which have an intensity of 30, to 25, which is white, as this creates a binary mask where the lung regions, the brighter areas, are white. After thresholding, the outer contours of the image are detected and compressed to, both, save memory and identify lung boundaries. The largest contour is then identified, assuming that it represents the lung region, to ensure more focus of the most significant lung structure. Once the region of interest, that is the lung region, is identified, the lung region is extracted by cropping the image using the bound box coordinates of the largest contour, hence removing unnecessary background information. Once the image has been cropped and the lung region extracted, an additional filtering process is applied to remove any unwanted noise and artifacts that may hinder the model's learning process. Since the image was converted into a grayscale image, specific grayscale intensity values are targeted using a predefined target grayscale values based on prior analysis, with a tolerance range to account for the slight variations in intensity. Any pixels which have the same intensity of the target values and the tolerance would be set to 0, that is black, to ensure that the most relevant lung features remain only. Once all of the feature enhancement procedures are implemented, the image is visually verified to ensure that the lung region has been extracted and the target pixels have been removed. This feature enhancement step is crucial to ensure that the model can focus on the most relevant lung features during its training phase in order to yield better results and to also speed up the training process.

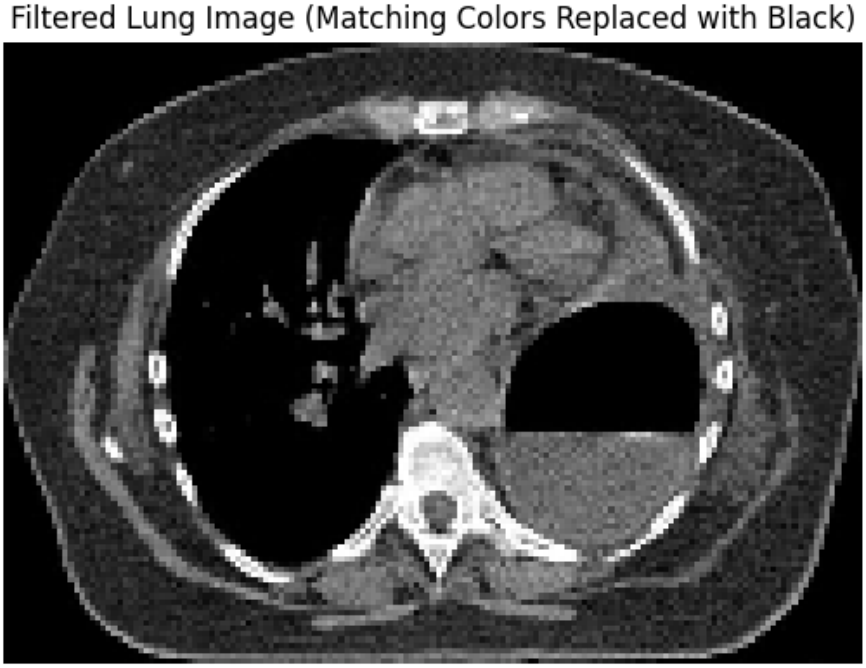



Figure 4.5: CT scan after further feature enhancement

## 4.5 Creation and Application of the Tested CNN Model

Once the dataset has been preprocessed and the features of each individual image has been extracted, it is now ready to be passed onto the model for training. Initially, the two folders containing the cancerous and noncancerous images were combined again into a single NumPy array X and its corresponding labels stored in another NumPy array y. Once concatenated, both X and y were shuffled to ensure an even distribution of both cancerous and noncancerous images during the following splitting process. After shuffling, the dataset was divided with 70% being allocated for the training phase, 15% for the validation phase which will be used for tuning hyperparameters and early stopping, and 15% being allocated for the test phase and to be used to check during the final accuracy. The dataset to be used during the training phase then goes through a data augmentation step, where random transformations such as rotation, shifts, flips and zooms is applied to the images, to ensure that there is data diversity and to prevent overfitting by making the model more robust to variations in lung scans.

The dataset to be used during the training phase is now finally passed onto the CNN model. The model initially consists of three convolutional layers, where, in the first convolutional layer has only 32 filters and the model learns and detects any straight lines, curves and color changes. In the second convolutional layer, consisting of 64 filters, the model begins to learn and detect any small textures, parts of objects, or structures like the lung nodules. Finally, in the third convolutional layer, consisting of 128 filters, the model detects full objects or regions of interest, such as abnormal lung patterns, and starts to identify tumour-like structures by combining the lower-level patterns.

After the convolutional layer, the feature map created during the convolutional layer is forwarded to the pooling layer, where through three MaxPooling layers, the most important features are kept and noise is removed. In the first MaxPooling layer, fine details like small specks of noise and very sharp edges are removed whereas the basic structural patterns like the lung contours and large anomalies are kept. In the second MaxPooling layer, Small textures which are not relevant for identifying tumors and minor variations in brightness are removed and key structures of the nodules, lesions and abnormal areas are kept. Finally, in the third MaxPooling layer, tiny, irrelevant features that might lead to false positives are removed and only the most critical features, such as large tumour regions or distinctive lung patterns are kept.

The pooling layer then passes multiple small 2D feature maps to the flatten layer, which the flatten layer then maps to a 1D vector that the Dense layer can now process. There are two parts in the Dense layer, where the first dense layer exacts high-level features from the flatten layer feature maps and recognises complex patterns like tumour shape, size and textures, and in the second Dense layer, the model combines the extracted features into a final prediction and is ready to make binary classification of 0, non-cancerous, and 1, cancerous.

The model is then trained for the duration of ten epochs with a batch size of two.

During each epoch the model is saved when it reaches the best validation accuracy, ensuring that only the best-performing version of the model is obtained. A further step was also taken to ensure that the model is stopped early if the validation loss does not improve for a period of 3 epochs. With this, the initial training phase of the model is completed.

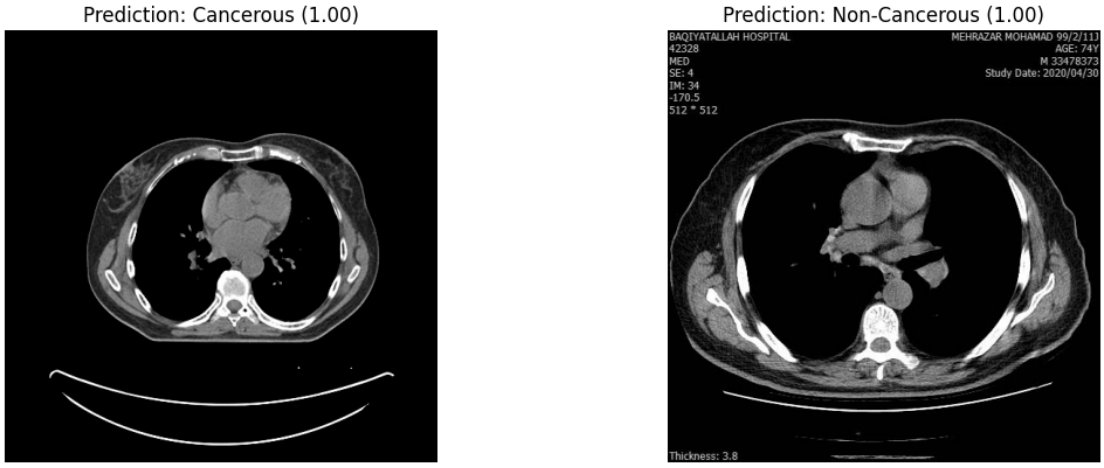
Once the initial training phase of the model is completed and the performance is evaluated on the test set of the data, further step has been taken to identify the specific images that the model had misclassified, by storing the misclassified images, the models incorrect predictions (labels) and the correct labels of those images. This step is crucial as once all the misclassified images have been collected it would help us understand the type of errors the model is making and, if the model misclassified images which are similar, it would indicate that the model is struggling to understand certain features of the images.

To address this, the model goes through a fine-tuning process, where the model is retrained on the misclassified images and the correct labels of the image through a short retraining of 10 epochs and a small batch size, allowing the model to learn from its mistakes and refine its decision making process. Once the model has been retrained, the performance is again evaluated on the full test set of the dataset to determine the improvement, if any, has been made and the overall performance of the model.

Additionally, the training history of the model is plotted to visually compare the performance, by comparing the accuracy and loss trends between the training and validation sets. Finally, the best-performing model, which was saved during the training process, is loaded, and the predictions are tested again on a subset of the test images to verify the improvements in classification accuracy.

## 4.6 Testing

After our model training was completed, we tested it by uploading randomly selected images from our dataset and observed the results. Using a predict function, where the probability is calculated, if the probability of the image having lung cancer is greater than 0.5, the model predicted it to be cancerous.




Cancerous Non-Cancerous

Figure 4.6: Cancerous and Non-Cancerous Predictions

## 4.7 Initial Results

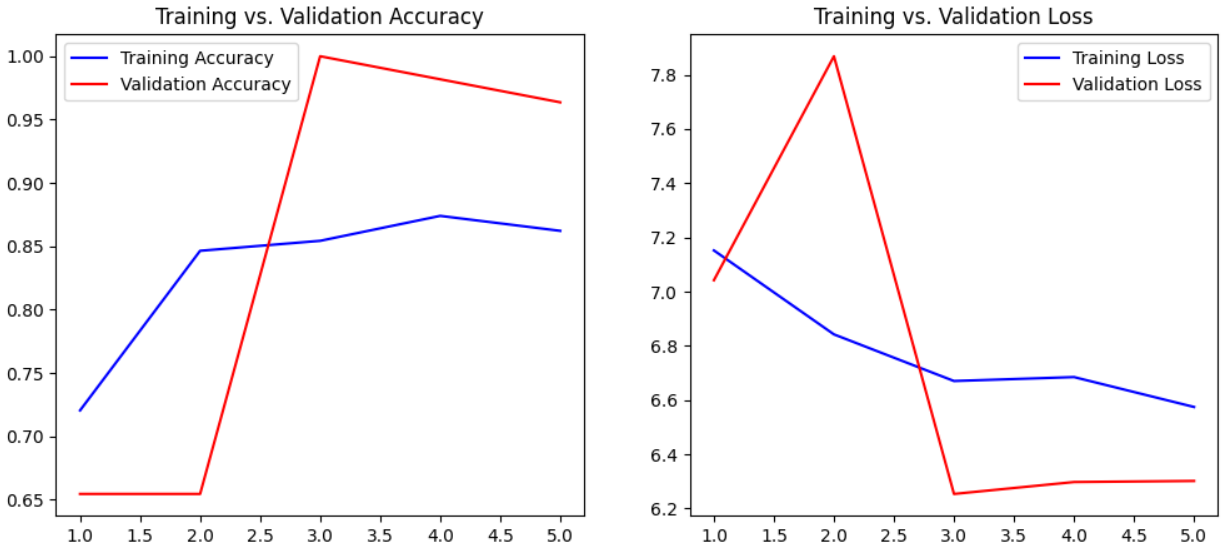



Figure 4.7: Visualization of Accuracies and Losses

After we passed the dataset through our CNN model and tested it by uploading random CT-Scan images from our dataset, we then plotted graphs of Training vs Validation in order to visually represent their accuracies and their losses. We observed that the training accuracy reached a peak of 87.70% whilst the validation accuracy reached 96.36%. On the other hand, the training and validation losses were pretty high. This revealed that whilst the model was performing well as the accuracies achieved were decent, the losses were still very high. This is due to the L2 regularization factor. The L2 regularization is essential in order to prevent overfitting of the model, however by reducing the penalty, we may able to decrease the value of losses in the future.

# Chapter 5

# Methodology

Our aim is to make sure our final model can adapt to new data based on generalization, rather than memorization, using an unsupervised feature learning strategy of a Deep belief Network. This is crucial when it comes to the medical field as a labeled data is expensive and/or hard to come by. Supervised models (like our initial CNN model) rely heavily on labeled data, whereas the Restricted Boltzmann Machines in our Deep Belief Network can learn patterns in the data and can extract deep latent representations and can generalize unseen test samples more effectively. The risk of overfitting is also lower in our final model as our DBN acts as a regulizer and learns compressed feature representations.

## 5.1 Description of the Hybrid Model

This research proposes a two-stage classification pipeline for lung CT image classification using a combination of Convolutional Neural Networks (CNN) and a Deep Belief Network (DBN) consisting of stacked Restricted Boltzmann Machines (RBMs). The proposed system would be able to reliably determine cancerous and non-cancerous lung CT images. As shown in the accompanying flowchart, the entire workflow can be divided into a number of different steps, that can, together, ensure proper preprocessing, feature extraction, classification, and model evaluation.

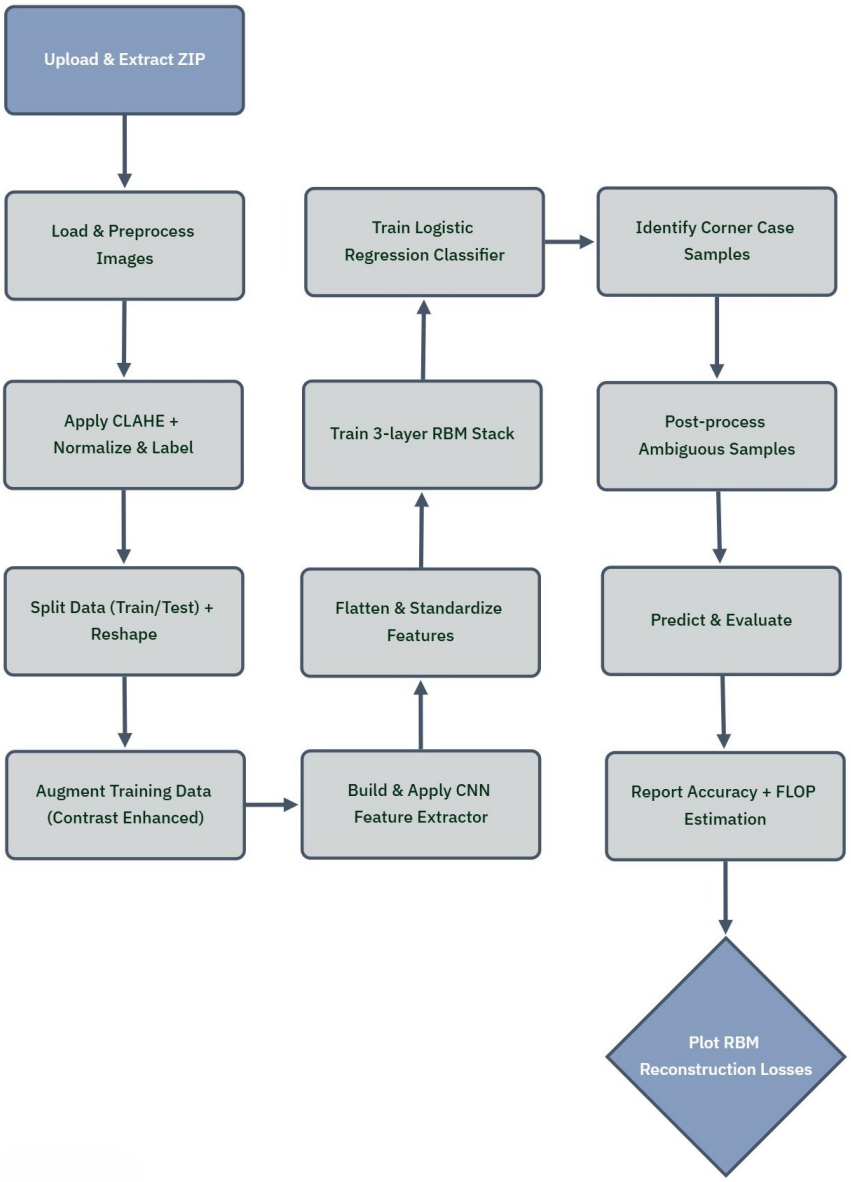



Figure 5.1: The pipeline

The pipeline starts by uploading and extracting a ZIP file that hosts the dataset. This makes sure that the data of all images are present in a structured form to be used in further tasks. Images are then loaded and preprocessed, which consists of resizing to a general form, acceptable by the CNN, and then adjusting the color space, if is required. The main enhancement made during preprocessing is the use of Contrast Limited Adaptive Histogram Equalization (CLAHE). The method enhances local contrast of the images, which is especially useful in medical imaging where the small difference in tissues may be diagnostically important. CLAHE is followed by normalization to scale the pixel values to the same range, which makes model training more likely to converge. At the same time, binary labels are applied: samples with cancer are marked as 1, non-cancerous samples as 0.

After preprocessing, the data is randomly shuffled and divided into training and testing set, which ensures unbiased evaluation. In order to further increase the di-

versity of the training data, contrast enhancement is performed on a sub-set of the training samples, which can be thought of as a simple data augmentation method. These two representations, original and enhanced, are jointly used to better generalize the model.

A CNN-based feature extractor is constructed and trained on the combined dataset. The CNN is responsible for learning low- and mid-level spatial features that are critical for distinguishing between the two classes. Once trained, the CNN is used to extract feature maps from both training and testing samples. These feature maps are flattened into 1D feature vectors and standardized using techniques like z-score normalization to prepare them for RBM processing.

A 3-layer RBM stack is then constructed. Each RBM layer progressively learns more abstract representations.The first RBM reduces the input to 512 nodes, he second to 256, and the third to 128. These stacked RBMs form a Deep Belief Network (DBN) capable of capturing hierarchical dependencies in the features extracted by the CNN. Each RBM is trained layer-wise in an unsupervised fashion using contrastive divergence, and reconstruction loss is monitored throughout to ensure convergence and feature learning effectiveness.

Once the RBM stack is trained, a Logistic Regression classifier is applied on top of the learned representations. This classifier is responsible for the final binary classification. Predictions are made on the test set, and samples with ambiguous or incorrect predictions are identified as corner cases. These misclassified samples undergo post-processing, including further contrast enhancement and re-extraction of CNN features. The updated features are passed through the RBM stack again and evaluated.

The pipeline ends with the detailed analysis of the performance, accuracy, precision, recall, and F1 score, which is displayed in the form of a classification report. Moreover, the number of floating-point operations per second (FLOPs) are estimated in CNN, RBM stack and Logistic Regression components, which provide an idea about the computational cost of each operation. The overall FLOPs can be used as an indication of efficiency and scale of applicability in a practical implementation. Finally, a line graph is used to visualise RBM reconstruction losses between epochs, which would give an understanding of the training dynamics and overfitting propensity of the unsupervised learning phases.

## 5.2 Image Preprocessing using CLAHE

The pipeline used Python libraries which included NumPy, OpenCV, and PIL (Python Imaging Library) to deal with image loading, transformation, enhancement, and labeling.

At first, we converted the images to grayscale in order to reduce computational complexity and also enhance local contrast and accentuate principal visual features using Contrast limited adaptive histogram equalization (CLAHE). The images were resized to 64x64 pixels uniformly and then normalized into [0,1], marked

with their respective class labels. CLAHE is an adaptive histogram equalization technique highly suited for medical image enhancement. CLAHE enhances local contrast without over-amplifying the noise, as compared to conventional histogram equalization. CLAHE is useful in revealing subtle structures in CT images, which are typically important in cancerous vs. non-cancerous tissue differentiation.

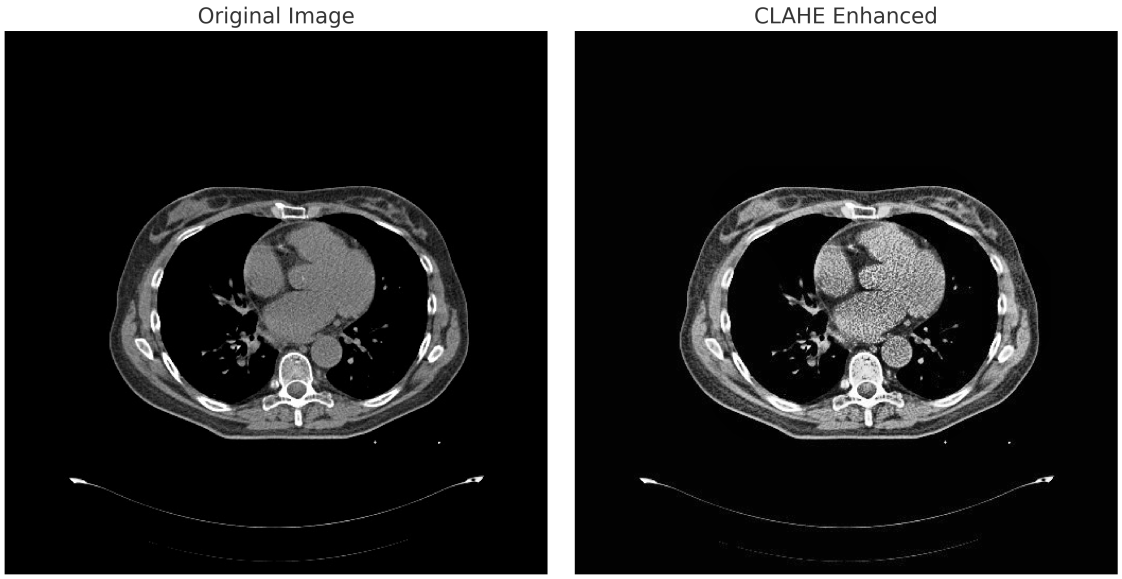



Figure 5.2: CLAHE

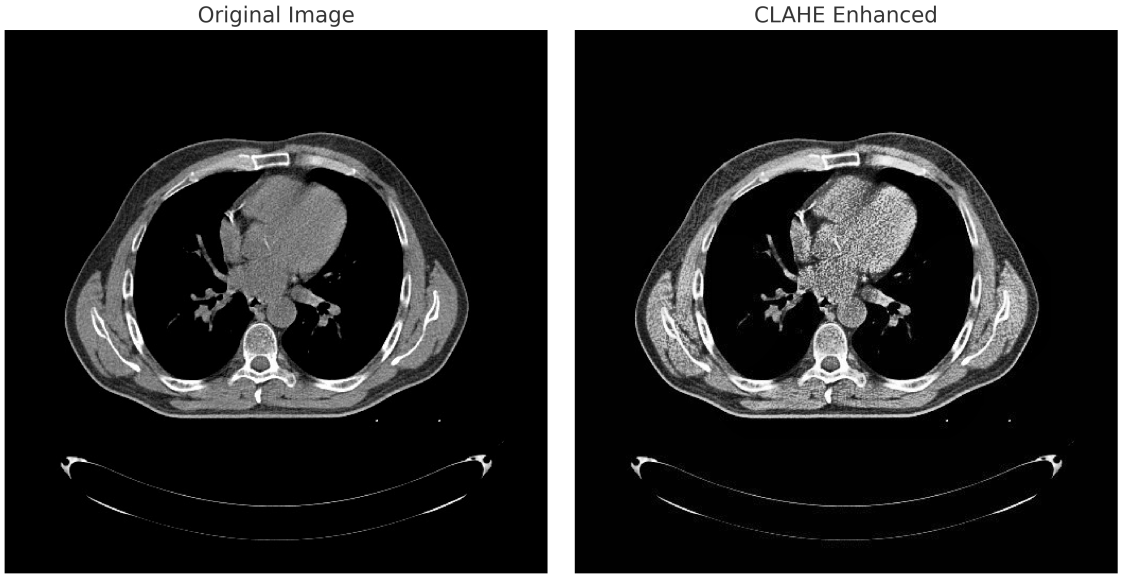



Figure 5.3: CLAHE

Gabor filters were also contemplated but ultimately not utilized because they are computationally expensive and have a tendency to reinforce unimportant patterns or background information. In contrast, CLAHE retains anatomical boundaries more strongly and adapts to local variations within the image without introducing pattern artifacts.

There were two types of lung images processed. A binary label was assigned to each image accordingly. The cancerous images are labeled as "1" and the non cancerous images as "0".

All the images were normalized by scaling its pixel values from [0, 255] to [0, 1]. This normalization is common in image based machine learning pipelines and averts making all inputs numerically unstable and incompatible with standard neural network activation functions. The images are finally processed and stored in a NumPy array which offers efficient storage and is well supported by deep learning frameworks.

The images are also read off disk in a fixed and sequential order to ensure the results were reproducible for several runs. Any files that were damaged or unreadable upon load were skipped and an error message was the output for debugging assistance and auditing data.

This preprocessing step was an essential data preparation process since it basically enhanced the visual integrity of lung images and standardized them for it to be usable in the classification model. It ensured that high quality, uniformly formatted inputs were fed to the deep learning model which in return improved the reliability and accuracy of the classification task. After the preprocessing of the lung images, the next step was to prepare the latent dataset for training and testing through stratified splitting and data augmentation. It was expected that the model should be trained on a balanced dataset that could generalize well to unknown data.

The dataset is then shuffled to eliminate any bias that have been introduced through the order of image loading. Then each image and their labels are also shuffled using a random seed. This randomization is helpful for generating a test set and training set that can help represent the general data distribution. The dataset is divided in an 80:20 split of training and test set. It is to be noted that stratified sampling was employed at this split. Stratification implied that the proportion of cancer to non cancer samples were preserved in both training and test set. This is especially valuable in medical datasets and since class imbalance can lead to unbalanced models with mediocre performance on minority classes.

After splitting the grayscale lung images, which were originally 2D arrays of 64x64 size, they were transformed into 4D tensors of size 64x64x1. By this transformation, each image gets a single channel depth dimension added to it so that they can be used with convolutional neural networks which are designed to take input as height x width x channels. To enhance the robustness of the training data even more, a contrast based data augmentation is used. Specifically, the contrast enhancement was performed on the training images with a factor of contrast adjustment as 1.5. This simulates imaging situations where the contrast varies, which is common under

real medical imaging due to variation in scanning devices of status of the patients.

The contrast augmented images were then merged with the original training set to form a combined training set. This basically doubled the training data and improved the generalization ability of the model by exposing it to both standard and contrast changed representations of the lung tissues. The contrast augmented labels were then duplicated accordingly to make them correspond with the amplified inputs.

By Combining the original and augmented images in training, the pipeline presents a controlled variability that forces the model to learn features invariant to varying levels of contrast. This kind of augmentation strategy is important in the healthcare industry where minor variations of image contrast can highly affect diagnostic accuracy.

## 5.3 Feature Extraction using CNN

Previously, we had observed promising results when we used the CNN model, we decided to utilize it's effective hierarchical feature learning to capture complex and abstract features in our image data.

Our CNN design consisted of four blocks of convolution, where each block, a convolution layer with ReLu activation and max pooling layer were combined together. This had allowed the model to learn more advanced spatial features up to a high capacity final layer of 256 filters in order to enhance discriminative ability further. Rather than using CNN for classification, we employed it as a feature extractor by flattening the final feature maps to produce short and abstract representations for the input images. In order to prepare these representations for further modeling, we employed Scikit-learn's StandardScaler to normalize the features to unit variance and zero mean. It was a necessary step to optimize performance of the subsequent machine learning models. We also employed an unsupervised Bernoulli Restricted Boltzmann Machine (RBM) to learn a compact latent space from CNN features. By combining the strength of deep convolutional feature extraction and probabilistic modeling, we had attempted to learn fine grained patterns and complex structures in the lung image data.

In order to utilize the representational power of deep models on image-based tasks as much as possible, a Convolutional Neural Network (CNN) was employed as a feature extractor. CNNs are very well adapted to image data because they are able to learn spatial hierarchies using convolutional layers. In this case, the CNN was not utilized for end-to-end classification but rather used to extract dense high-dimensional feature representations from the final or the intermediate layer of the network. Specifically, the combined training and test image datasets, namely X_train_combined and X_test_img respectively, were passed to the CNN model using the .predict() method provided by the Keras API. This technique performs a forward pass on each image in the training dataset, feeding the training convolutional filters and pooling layers on the uncooked pixel values to transform them into higher-level features. The output of this process was two NumPy arrays, cnn_train and cnn_test, where each sample

was the high-level feature map generated by the CNN for an input image.

The outputs cnn_train and cnn_test from that were three-dimensional or four-dimensional tensors, depending on the architecture of CNN and on the output layer used (typically with shape (number_of_samples, height, width, channels)). Since most classical machine learning algorithms—like Support Vector Machines (SVM), Logistic Regression, or k-Nearest Neighbors (k-NN)—prefer their inputs to be in some sort of flat, vectorized form, the high-dimensional feature maps were flattened to two-dimensional matrices. This reshaping was achieved by employing the .reshape() function of NumPy in a manner that the number of rows was equated to the number of input samples, and the number of columns was equated to the total number of features learned for each sample. i.e., cnn_train.reshape(cnn_train.shape[0], -1) and cnn_test.reshape(cnn_test.shape[0], -1) reshaped the 3D/4D tensors into 2D arrays X_train_flat and X_test_flat of shapes (n_samples, n_features). This flattening maintained the data present in the feature maps but reshaped the data for consumption by downstream algorithms.

The features, after flattening, had to be standardized so that each feature contributed equally to learning. Raw output from CNN comprises a huge range of numerical values, and this may bring bias into models that use distance metrics or gradient descent. To counteract this, standardization of the features was done using StandardScaler class of sklearn.preprocessing. StandardScaler normalizes features by subtracting the mean and scaling to unit variance, by the formula:

$$x' = \frac{x - \mu}{\sigma} \tag{5.1}$$

where $x$ is the original value of the feature, $\mu$ is the mean of the feature, and $\sigma$ is the standard deviation. The scaler was first used on the training data through `scaler.fit_transform()`, which computed the mean and standard deviation of all the features in `X_train_flat` and applied the transformation. This left the transformed training features with a zero mean and unit variance. The same transformation (with training data parameters) was then applied to transform the test features using `scaler.transform()` in order to preserve consistency and prevent data leakage. In addition, all feature arrays were also intentionally converted to 32-bit floating-point type (`np.float32`) prior to scaling, to ensure compatibility with machine learning libraries and to maintain numerical stability.

Two new datasets, X_train_scaled and X_test_scaled, were obtained as an outcome of this preprocessing pipeline. These datasets are highly flattened CNN feature vectors that are normalized and now prepared to be fed as input to standard machine learning classifiers. This classical machine learning and deep learning model hybrid approach of combining is well proven to be very successful, especially in those cases where there is limited labeled data or where interpretability is a key factor.

## 5.4 Deep Belief Network

A Deep Belief Network is a deep neural network constructed by multiple stacked layers of Restricted Boltzmann Machines (RBMs), one on top of the other. The RBMs are then followed by a feedforward neural network (FFNN) in a supervised learning process.

Formally, a DBN with $L$ layers defines the joint probability distribution over the observed vector $x$ and the hidden layers $h^1, h^2, \ldots, h^L$ as follows:

$$P(x, h^1, \ldots, h^L) = P(h^{L-1}, h^L) \prod_{l=1}^{L-2} P(h^l | h^{l+1}) P(x | h^1)$$

Here, the top two layers form a Restricted Boltzmann Machine, and the remaining lower layers form a directed sigmoid belief network.

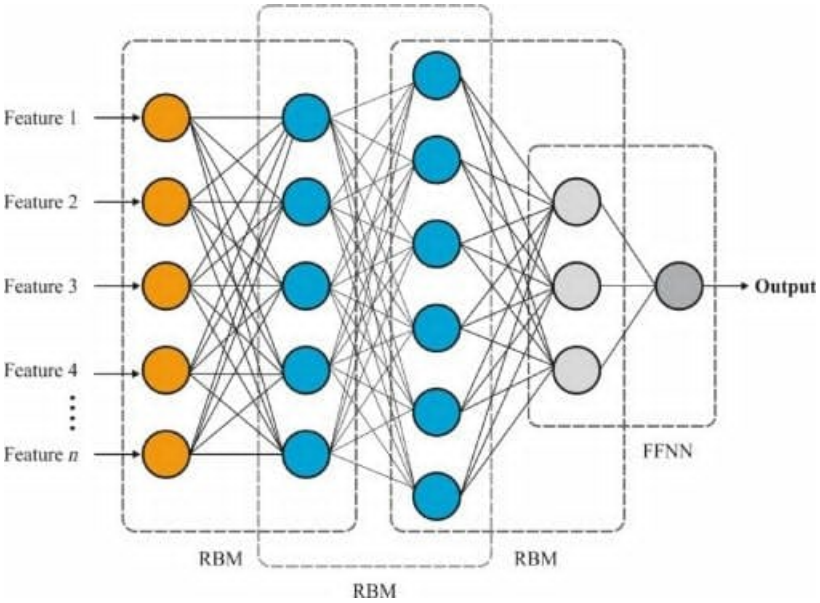



Figure 5.4: Deep Belief Network
[23]

The above picture depicts the structure of a standard Deep Belief Network (DBN). The left-most part of the diagram is the input layer which has the input nodes and each node is associated with one feature of the dataset. This is fed into the first RBM which consists of a visible layer (the inputs) and a hidden layer and is connected in a bipartite manner. The RBMs are considered as the weighted logistic regression on a binary machine. The initial RBM is trained (in an unsupervised fashion) to learn patterns in the training set, and it learns low-level representations of the kind that might be found in the first few layers of a convolutional neural network.

Restricted Boltzmann Machines (RBMs) are generative neural networks that are used in unsupervised learning, which are extremely useful in tasks such as dimensionality reduction, feature learning, and pre-training deep nets. Two layers form an RBM: a visible layer that is utilized to represent the input of the data and a hidden layer that learns to identify the structure or features of the data. The key characteristic of an RBM is that it is "restricted," i.e., there are no connections within a layer only between visible and hidden units so that inference and training are easier. RBMs learn to map the probability distribution of input data by minimizing an energy function that quantifies the compatibility between visible and hidden states.

Training RBMs typically reduces the maximum likelihood of observed data. But because calculating the correct expectations across all possible configurations is computationally intensive, an effective approximation method known as Contrastive Divergence (CD) is generally used. CD employs a short Markov chain in order to compute an estimate of the model distribution so that RBMs may be efficiently learned over vast datasets. While a simplification, CD works well in practice, though sometimes it forms a trade-off between learning rate and generalization performance.

Generalization ability in RBMs is the potential of RBMs to learn useful patterns from training data and generalize the learning to unseen data from the same distribution. Generalization ability is influenced by a very large number of factors. The most important one is the capacity of the model, based on the number of the hidden units. If there are too few, the model will underfit and will not be able to capture complex patterns; if there are too many, it can overfit, learning noise instead of structure. Regularization techniques such as weight decay, sparsity penalties, and dropout alternatives can be used to encourage generalization. Furthermore, early stopping in training based on validation performance prevents overfitting and results in better generalization.

RBMs have been shown to possess good theoretical generalization properties. They are discrete distribution universal approximators, i.e., they can approximate any data distribution with enough hidden units. RBMs also learn to project the data as being located on a lower-dimension manifold in the hidden space. This capability makes them capture the underlying structure of high-dimensional inputs effectively, leading to generalization. The manifold learning behavior is what makes them so effective in tasks like anomaly detection, where distinguishing between normal and outlier patterns is crucial.

Practically, the generalization ability of RBMs is experimented with many meth-

ods. Standard approaches include calculation of reconstruction error on held-out validation data, calculation of log-likelihood on test data using approaches like Annealed Importance Sampling, or examining how well RBM-learned features generalize to downstream applications such as classification. Collaborative filtering, image classification, and speech modeling tasks have demonstrated that RBMs will generalize well if trained with appropriate hyperparameters and regularization. Their tractability, theoretical foundation, and practicality all act together to make RBMs an important tool for constructing robust machine learning systems.

All the RBMs in this stack are trained separately with a layer-wise greedy training algorithm, usually with Contrastive Divergence, to approximate the gradient of the log-likelihood. This unsupervised pre-training step enables the network to initialize weights in a meaningful manner, particularly useful in deep networks where random weight initialization usually results in slow or unstable training because of problems such as vanishing gradients. Once all the RBMs have been trained, the network is converted into a standard deep feed-forward neural network, with the pretrained weights serving as initial weights in the corresponding layers. More fully connected layers are added (with an output layer), to complete a supervised task, either classification or regression. End-to-end network parameters are then refined on labeled data with backpropagation, to maximize task performance.

Deep Belief Networks come with several advantages to show, particularly in cases when labeled data is limited, or when it is preferable to extract features in an unsupervised manner. Their hierarchical architecture allows complex structures in the data to be learned, enabling better generalization and robustness to noise.

The feature vectors generated by CNN are then passed through three Restricted Boltzmann Machines (RBMs) sequentially. Each of these RBMs in this Deep Belief Network learns more abstract and feature-extracted representations of the data through an unsupervised learning process. Contrastive divergence is employed to enable the learning process by reducing the reconstruction error between the input vector and its reconstructed version. Interestingly, while training, it is seen that the first RBM exhibits a rise in reconstruction loss because of the challenge in modeling raw features. Nevertheless, as the representation becomes more structured and compact by the stacked layers, the final RBM reaches close-to-zero reconstruction loss, reflecting efficient encoding of high-level patterns.

We stacked three RBMs, one after another to build a deep unsupervised learning pipeline. All RBMs reduced the dimensionality of the feature space without destroying the structure that was required. The first RBM reduced the 4096-dimensional flattened feature from CNN to 512 units, the second reduced 512 to 256, and the third from 256 to 128. This step by step reduction created a hierarchical abstraction of the image data that helped improve downstream classification. In order to keep track of the training progress, we had added to the RBM class, the ability to monitor reconstruction loss per epoch which has facilitated real time visual inspection of convergence and model stability.

## 5.5 Refinement

In the final stages of our pipeline, we incorporated a refinement strategy in order to allow our model to deal with corner-case images which are harder to classify and categorize. Instead of changing the decision boundary or going for retraining, in this stage our model focuses on identifying subtle inputs that are hidden or masked by ambiguous or obscured features. As these inputs are edge-cases and rare in the training data, they require additional attention as they are difficult to classify.

The model identifies instances where it struggles to assign a high-confidence value on the corner-case samples. CLAHE is reapplied on these images in order to target the latent structures that require more attention. The resulting images are processed a second time with the CNN to produce another set of refined feature maps. These new features are then again sent through the already trained RBM stack to produce improved high-level representations using the model's original unsupervised encoding, hence no retraining or architectural changes are required. In the final test feature set, only the feature representations related with these edge case undergo modification. This approach guarantees that the refinement process does not alter well generalized predictions. This approach incorporate model-driven feedback which enhances performance by refinement without disrupting the overall coherence of the model.

The model revisits to reassess confusing samples instead of relying on the first output. This is crucial if we consider how experts might need to re-evaluate questionable CT Scan images as it may be crucial in saving a patient's life. This feature of the model suggests how it mitigates the problem of overfitting and demonstrates its ability to rely on generalization and refinement.

## 5.6 Classification

In the final classification layer in the Deep Belief Network (DBN), Logistic Regression takes the high level feature representations learned by successive layers of Restricted Boltzmann Machines (RBMs) and projects them into an interpretable binary output. Once the DBN has been trained in an unsupervised layer-wise fashion using the RBM to extract hierarchical and abstract features on raw input data, the resultant transformed feature vector (the output of the final RBM) is fed as input to the logistic regression classifier.

Logistic Regression is a common machine learning algorithm, a supervised learning algorithm to be more precise, which is used in problems involving binary classification, that is, the output variable can only have one of two possible values, most often 0 and 1. The model works by receiving many input features, each with an accompanying weight and calculates a linear combination of the inputs. This weighted sum is also referred to as the net input and it is the basis of the prediction. In order to transform this linear output to a probability, Logistic Regression uses a sigmoid activation function which squashes the input to a value between 0 and 1 (a continuous value). This probabilistic output is then run through a decision-making operation-commonly a threshold of 0.5, whereby values beyond the threshold belong to one

class (normally assigned the value 1), and the rest belong to the other class (normally assigned the value 0). One of the most significant reasons logistic regression is used so widely in practice is that it has a very high generalization property — the capacity to maintain high predictive performance when applied to out-of-sample or unseen data. The low complexity of the model and the bias that prevents the model from capturing spurious patterns or noise in the training data are what give the model its generalization capacity. Compared to overly flexible models that readily overfit, logistic regression models are actually very good with the belief that the underlying relationship between the predictors and target is roughly linear at the log-odds scale. It is especially robust in high-bias, low-variance scenarios where interpretability and stability are deemed more important than fitting intricate nonlinear relationships. Moreover, logistic regression naturally follows the maximum likelihood estimation principle, whereby the resultant parameters minimize the likelihood of the training data observed and hence promote a statistically robust form of generalization.

Regularization techniques improve the generalization capability of logistic regression in practice. Regularization introduces a penalty to the loss function used in training models to inhibit very complex models by shrinking the size of the model coefficients. L2 regularization (ridge) penalizes proportionally to the square of the coefficients, while L1 regularization (lasso) sparsifies by forcing some of the coefficients towards exactly zero, thus acting as an embedded feature selection tool. The two techniques are most useful in high-dimensional data or multicollinearity situations, where they reduce variance without excessively increasing bias. Regularization not just improves generalization by regulating the model capacity but also helps to disconnect the strongest predictors, thereby making the model more interpretable and resilient.

Another problem that influences logistic regression generalization is training data quality and preprocessing methods employed. Feature scaling, missing value handling, outlier detection and removal, and proper encoding of categorical attributes all lead to improved generalization. In addition, techniques such as cross-validation are also very crucial in testing the model and improving the ability of the model to generalize. By recursively splitting the training and validation data, cross-validation provides a fair measure of how well the model will generalize to new data and avoids over-estimation of the performance of how good it is on the training set alone. Use of learning curve and validation curve can also decide if the model is high variance or high bias and guide additional attempts at model tuning or feature engineering.

Unlike more complex classifiers such as neural networks or support vector machines, logistic regression provides a strong baseline that is both efficient and interpretable. Its generalization performance can be outperformed in challenging nonlinear worlds; but in most practical problems with decently sized samples and pretty clean, nicely structured data, logistic regression performs more or less as well, especially when supplemented by heavy regularization and feature selection techniques. Its probabilistic output also facilitates fine-grained decision-making, e.g., threshold adjustment to application-specific false positive and false negative costs. This is particularly useful in applications such as medical diagnosis, fraud detection, and risk assessment where interpretability and trustworthiness of the model are as impor-

tant as raw predictive accuracy.

Last but not least, logistic regression is not only a statistically sound and computationally tidy classification algorithm but also a good generalizing one when appropriately regularized and tested. Its simplicity relative to efficacy, combined with its transparency and ease of application, render it a preferred answer in academia and industry alike. For model generalization research-based theses, logistic regression is a key reference and benchmark point of consideration that yields insightful information concerning the complexity, interpretability, and efficacy trade-offs of predictive modeling.

Logistic regression models the probability of the binary outcome $y \in \{0, 1\}$ as:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-(\mathbf{w}^\top \mathbf{x} + b))}$$

where:

- $\mathbf{x}$ is the input feature vector,
- $\mathbf{w}$ is the weight vector,
- $b$ is the bias term.

The binary cross-entropy loss function used during training is defined as:

$$\mathcal{L}(\mathbf{w}, b) = -\frac{1}{N} \sum_{i=1}^{N} \left[ y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

where $\hat{y}^{(i)} = P(y^{(i)} = 1 \mid \mathbf{x}^{(i)})$ is the predicted probability for the $i$-th example.

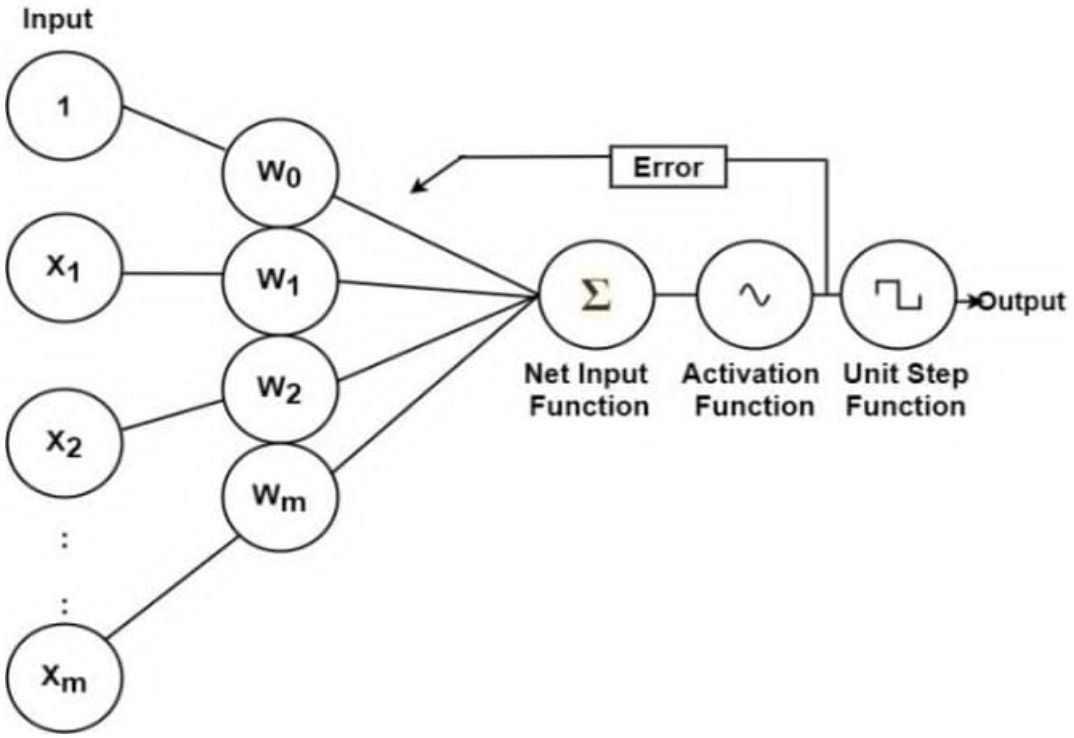



Figure 5.5: Logistic Regression

This is shown in the diagram and starts with the input layer and associated weights, net input calculation, application of activation function and the binary decision at the end using a unit step function. Notably, the Logistic Regression has a feedback mechanism built on the difference between the predicted output and the actual output. It is this error that is used to update the weights of the model in an iterative manner using optimization algorithms like gradient descent and thus the accuracy of the model increases with time. Not only is Logistic Regression computationally efficient and simple to evaluate, but it is also highly interpretable: the weights learned by the model can be directly related to the effect that each feature has on the prediction. This has made it especially useful in areas like medical diagnosis, finance, and social sciences where it is important to explain the models. Even though it might sound basic, the Logistic Regression is a strong baseline classifier, and many researchers take it as a point of reference when testing more advanced machine learning models.

# Chapter 6

# Results, Analysis and Comparisons

## 6.1 Results

Finally, after going through the entire pipeline which included a CLAHE and CNN based feature extraction, data augmentation, an unsupervised learning stage using a Deep Belief Network, classification using logistic regression, and very crucial post-processing refinement to deal with ambiguous and corner-case images, out model produced a highly impressive accuracy of 98.63%.

Not only is this value numerically strong and impressive, it also illustrates the fact that this pipeline is effectively efficient in generalizing unseen test samples across both cancerous and non-cancerous domains, highlighting its robustness.

The classification report provides us with a deeper breakdown of the model's performance, as the values of the F1-score, precision and recall demonstrate how effective the model is in differentiating between benign and malignant CT-Scan images.

The precision of the model is calculated by dividing the number of predicted positives by the model by the number of true positives. For the non-cancerous class, the precision result came out to be 0.96, which means that the model successfully identified benign samples 96% of the time. Conversely, the precision for the cancerous class was 1.00, which means that the model was successful in identifying all cancerous samples and that not a single malignant prediction was a false positive.

On the other hand, the recall value is a measure of the proportion of the true positive samples that had been correctly identified. The non-cancerous class had a recall value of 1.00, which means that the model was successful in correctly identifying every single benign image, whilst the recall of the cancerous. In the field of oncological medical diagnosis, this is crucial because missing a non-cancerous case (false negative) can result in needless medication of the patient, which is not only detrimental to the patient's physical health, but also their mental health as it may cause unnecessary distress and anxiety. Also, the high recall value of 0.98 for the cancerous class means that 98% of the time the model was successful in identifying a cancerous image, indicates that there is very little possibility that the model might

miss a sample where the patient is suffering from a malignant case.

The F1-score is the harmonic mean between the precision and recall values. The values of 0.98 for non-cancerous and 0.99 for cancerous respectively, indicate that the model is performing exceptionally well across both classes in terms of accuracy and robustness. This balance is imperative in a medical setting as any false alarm can prove to be of great harm to a patient.

The macro-average is a calculation of the mean without taking the class distribution into account. This guarantees that both cancerous and non-cancerous classes are treated and evaluated equally. The macro average precision, recall and F1-score had valued equaling to 0.98, 0.99 and 0.98 respectively.

The weighted average on the other hand, takes into account the number of samples belonging to each class. In this case as well, we can see very high values being delivered by the model. This highlights how the pipeline is able to perform impressively both when the class sizes are taken into consideration, as well as on a per-class basis as the precision, recall and F1-score all has a value of 0.99, which sheds light on the model's consistency and dependability.

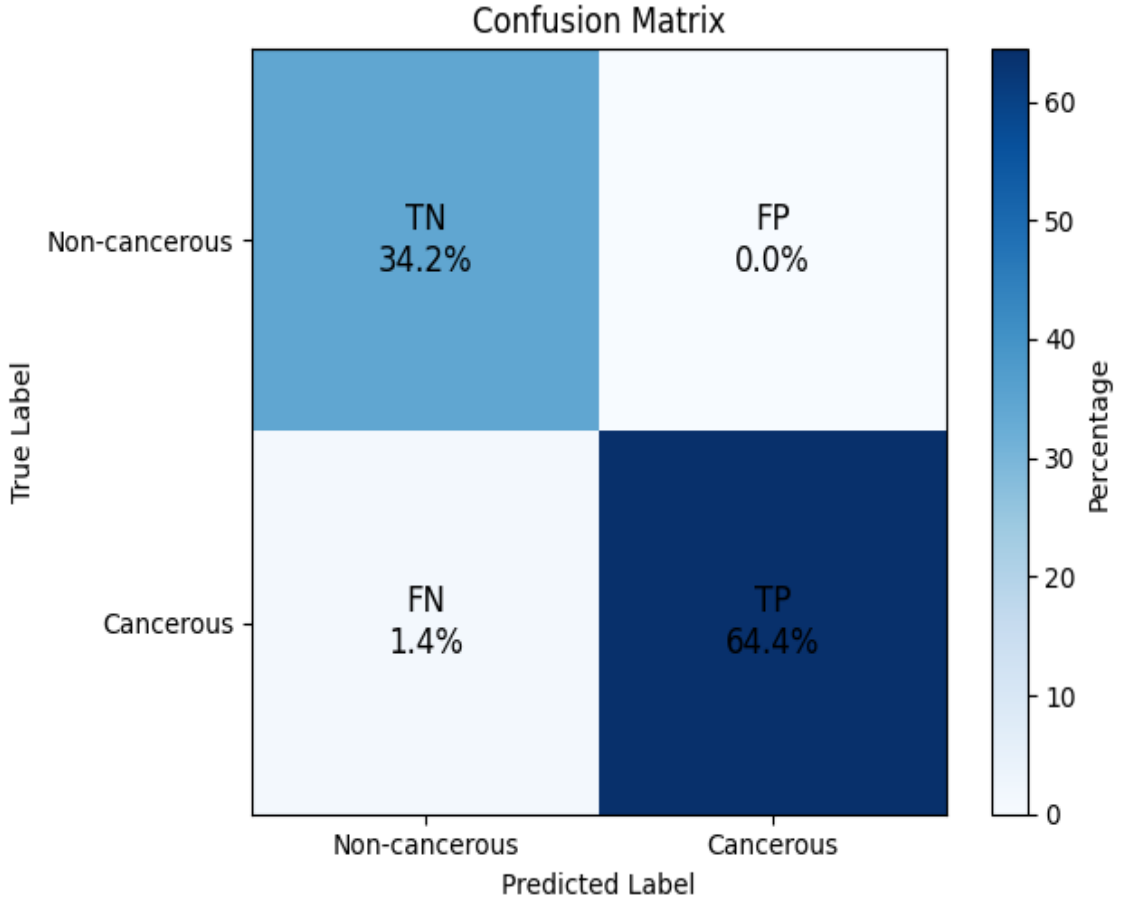



Figure 6.1: Confusion Matrix

The confusion matrix displays the classification results of the last CNN + DBN model after employed post-processing techniques. Model exhibits good discrimina-

tive ability between cancerous and non-cancerous lung CT images:

- True Positives (TP): 64.4% — The proportion out of the entire test set that was marked as positive was quite high, thus demonstrating good detection sensitivity.
- True Negatives (TN): 34.2% — The non-cancerous samples were identified with strong accuracy thus indicating good specificity too.
- False Negatives (FN): 1.4% — The proportion of diagnosed cases which were not detected is small, indicating that the model is calibrated well enough to avoid under-diagnosis.
- False Positives (FP): 0.0% — There were no non-cancer cases classified as cancer cases which enhances clinical safety.

In summary, the matrix indicates that the model has high precision and recall but particularly excels in identifying conditions correctly associated with cancer diagnoses. The lack of FP's and low numbers of FN's reinforces how reliable the system could be deemed for use in critical clinical situations where trust in provided information must be very high.

## 6.2 Importance of Refinement

A very significant observation is the role played by the post-refinement stage of our pipeline. As we target the edge-case samples which the model was unclear about and initially faced struggles in classifying. In this stage, we find these ambiguous samples, enhance their contrast, and re-process them rather than retraining the whole model or tune any hyperparameters. This mirrors the role of a diagnostician or a radiologist who may need to take a closer look at a confusing CT-Scan or when a patient decides to consult another doctor for a second look or opinion regarding a disease diagnosis which may alter the course of their life if misdiagnosed.

Those particular corner-case samples undergo feature enhancement and the remaining test set samples remain untouched. Similar to how real-world diagnosis happens, our model focuses on these complex samples for a closer analysis, reflecting a self-correcting method.

On an important note, this stage of our pipeline further boosts the generalization capability of our model instead of artificially inflating accuracy through data leakage or going through a process of retraining on test data. All the model weights remain unchanged during this process, and only the input samples of these ambiguous cases is enhanced and re-evaluated. This process highlights the fact that the model is learning discriminative representations of the samples instead of memorizing the input data labels and that the model is robust enough to handle data variability.

## 6.3 Integrity

To ensure strict evaluation and accuracy for each test sample, we enhance the images with contrast adjustment and use these versions for inference. During prediction,

only the original version of the sample is used unless a certain degree of ambiguity exists, in which case the contrast-enhanced version can be used as a substitution. Neither the boosted nor unaltered versions are included into the test set. This way we ensure there is no overwriting bias or duplication bias that could lead to an inflated estimation in precision by ensuring every instance serves as an input just once to conditionally decided evaluations. The size and structure of the test set remains intact, neither retraining nor modification of labels occurs preventing estimation inflation caused by modified evaluation protocols. Thus what is reported reflects improvements solely attributed to representational enhancements and not any evaluative protocol manipulation. Such measures uphold model evaluative quality while providing clearer images to advance reliability in decisions from model outputs.

## 6.4 Training Diagnostics

In order to gain more insights about the representational learning behavior of Deep Belief Network (DBN), we have tracked the reconstruction loss at every layer during 50 training steps. This was drawn based on negative log-likelihood values, which is a conventional proxy to measure the discrepancy between the original input and the reconstructed approximation by the RBM. The DBN structure employed here consists of three stacked Restricted Boltzmann Machines (RBMs) with diminishing hidden units: RBM1 having 512 units, RBM2 having 256 units and RBM3 having 128 units.

The reconstruction loss plot shows clear patterns in the layers and each can provide information about the learning challenge in various levels of abstraction. RBM1 that receives the direct input of CNN-extracted feature vectors starts with a small reconstruction loss, which slowly rises during iterations. This is an indication that the original representation space, though already compacted by the CNN, still has a significant amount of noise or variability that is difficult to reproduce by the RBM in a precise way. The increasing loss can also be interpreted as the aim that RBMs as unsupervised learners are not directed by label information and have to learn structure based on purely statistical relations.

RBM2, whose input is the hidden layer activations of RBM1 exhibits a very different trend. It starts with the same initial loss, but then it rapidly displays a linear growth of reconstruction error. This may be ascribed to compounded noise or unstable representations that may be forwarded by the first layer. Alternatively, it may be that the smaller number of hidden units in RBM2 simply cannot reproduce the rich distributions of intermediate features, and hence is more vulnerable to noisy variations in the input data. This sharp increasing trend indicates that RBM2 is having a hard time to learn a stable encoding of the already-transformed features, and hints at the fact that dimensionality reduction ratios in stacked RBM structures should be selected carefully. As a sharp contrast, the last RBM in the stack, RBM3, shows a flat curve quickly approaching zero. The virtually zero reconstruction error at this third layer suggests that the features are abstracted into a highly regular and compressible representation by the time the input waveform gets to this third layer. This convergence indicators that RBM3 is learning a highly structured latent space, linear separability is finally exploited by the logistic regression classifier that

is positioned on top. The fact that this curve is flat can alternatively be viewed as an indication of representational maturity: the last RBM no longer is learning high-variance patterns, but rather encodes stable, high-level abstractions that are well correlated with the downstream task of binary classification.

In general, the plot verifies that although the lower layers of the DBN perform a significant amount of transformation and compression of features, and in fact, tends to be unstable, the last layer of RBMs converges to a low-entropy representation. This speaks in favor of the architectural choice of logistic regression as the terminal classifier, because a model is expected to work best when the input space is already discretely divided. That pattern of increasing reconstruction loss between RBM layers therefore confirms both the unsupervised character of the DBN and its value as a regularizing, feature-structuring element of the end-to-end pipeline.

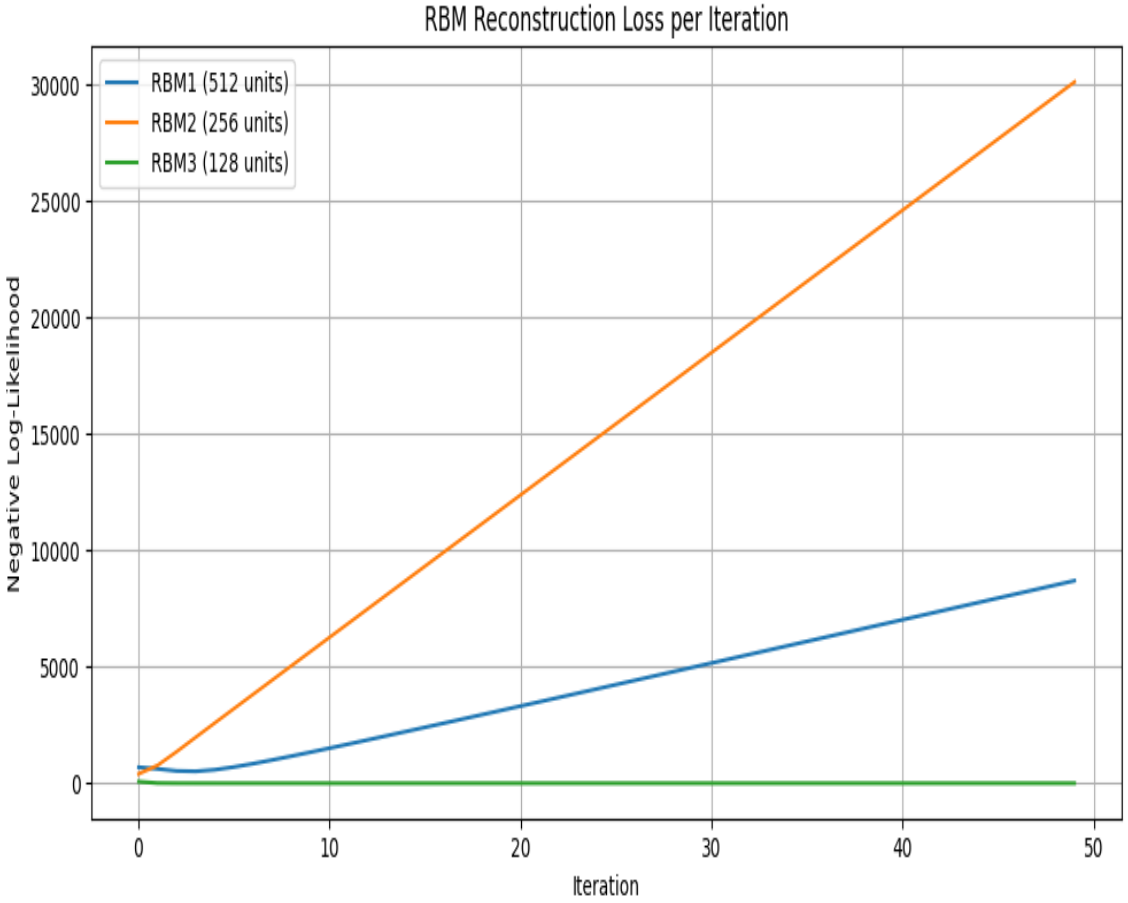



Figure 6.2: Reconstruction loss

## 6.5 Gaussian Blur + SVM

We tried an alternate preprocessing classification technique as a comparison to measure the resilience of our own model's architecture. Here, we used the Gaussian blurring technique as the primary image preprocessing technique instead of CLAHE and Support Vector Machine (SVM) instead of logistic regression for the final classification step.

Gaussian blur is a smoothing technique which is used to help reduce noise and detail in images, it is mainly used to not blur out unwanted textures and spackle patterns. This is useful in stabilising the feature extraction where there would be too much detail to fool the downstream model. To maintain the architectural consistency, the images are fed through the same CNN-DBN feature extraction backbone.

The SVM classifier essentially returns a decision boundary which is margin based and useful when working in high dimensional space and typically against overfitting. The post processing refinement had performed the same way as the primary pipeline when the output of the last layer was used as input features for SVM. This had achieved an accuracy of 82.19% at the final stage. No false positives were predicted for this critical class as the precision for cancer images was 1.00.

The benign cases were also being identified correctly as the recall for non cancer images was 1.00, these are extremely important in decimal diagnosis as the specificity might have life changing consequences. The average f1 score over all classes was 0.82 and the weighted average was 0.83, which shows the strong yet unbalanced performance of the model across the classes.

This model had functioned comparatively well, especially in processing non cancer images, however it was not able to produce better results than our CLAHE + Logistic Regression configuration. The accuracy gained suggests that even though Gaussian blur and SVMs can provide a good accuracy, they are not going to be as good at extracting the subtle patterns in low contrast CT scans as our model can.

## 6.6 Ensemble Architecture

To evaluate the performance of the proposed CNN, DBN and logistic regression pipeline, we compared it with an alternative solution using a traditional ensemble classifier. The ensemble setup included three common supervised learning models which includes logistic regression, random forest and support vector machine(SVM). Each classifier was trained separately on the final DBN converted feature space and their predicted probabilities were averaged to produce a final decision output. The results of this ensemble classifier method were a test accuracy of 81.0%, which is significantly lower than the 98.63% accuracy of our proposed method. While the ensemble classifier performed well in being able to correctly classify non-cancerous cases with perfect recall of 1.00 for that class, it showed a clear imbalance in performance with just 71% recall for detecting cancerous images. This is particularly disconcerting in the context of medical diagnosis where false negatives(i.e., failing to identify cancer) can be fatal.

In contrast, the current pipeline has 99% overall test accuracy after refining with balanced performance for both classes. Specifically, the model depicts precision as 1.00 and recall as 0.98 for the cancer class, whereas precision is 0.96 and recall is 1.00 for the non-cancer class. This depicts a high sensitivity and specificity rate and the ability of the model to efficiently perform in terms of avoiding false negatives as well as reducing false positives.

One of the essential constraints of the ensemble approach is that it is supervised learning in its pure form. While it benefits from the overt classification cues being directly correlated with label distributions it loses out on the abstracted, unsupervised feature compression of DBN layers. Further the ensemble also doesn't get the advantage of the test-time refinement mechanism that the proposed pipeline has a process that improves borderline predictions by re-running borderline test samples with higher contrast and improved features.

The ensemble classifier, despite the incorporation of multiple decision boundaries, is also more prone to be impacted by spurious correlations or to overfit the training distribution. In comparison, the DBN model regularizes the feature space through layerwise unsupervised learning so that the final classifier is able to generalize more effectively on unseen data. Briefly, although ensemble classifiers can be made to generate decision making diversity and improve marginal performance compared to single learners, they don't perform well in high dimensional and low sample size cases like in medical CT imaging. The CNN features extraction or DBN representation learning or logistic regression classification hybrid pipeline with test-time refinement for targeted expertise that surpasses the ensemble by significant margin both in unprocessed accuracy and class specific trustworthiness.

## 6.7 FLOPs Count

Whilst raw accuracy is very crucial, it is not the only important measurement metric used in the evaluation of machine learning models, especially those which have real life applications such as in the medical field. We can judge how practical a model is by its computational efficiency, which is often measured in Floating Point Operations, also known as FLOPs count. This metric plays a vital role in judging a model's practicality and efficiency.

In our pipeline, we achieved an accuracy of 98.63% with a FLOP count of about 47.5 million, 47,500,544 to be exact. Whereas the FLOPs count of a similarly high performing model reached an accuracy of 97.5%, had a FLOP count of 787 million, which is 16 times more than the computational cost of our model. So why is this beneficial? The much smaller computational footprint of our pipeline not only ensures faster inference, but is also compatible with scalable diagnostic systems. The lightweight design of our model makes it highly suitable for real world applications in clinical environments.

The well-planned and structured architectural design of our model results in very powerful efficiency. As we combine three effective yet lightweight components con-

sisting of a CNN to extract the rich local features from the CT Scans, a Deep Belief Network for the purpose of unsupervised hierarchical feature learning and finally, a logistic regression classifier which fulfills its role as a simple but strong classifier, we negate the need for brute-force methods such as excessive depth or inflated parameter count. Especially, our DBN not only acts as a feature compressor, but it also plays the role of a regularizer which allows the model to prevent over-fitting and it generalizes effectively to unseen data. The astute allocation of roles across the layers of the pipeline boosts the model's resilience and robustness. Whilst other architectures and models may produce accuracies which can compete with ours, the possibility of a real-world deployment is severely deterred by their computational requirements.

## 6.8 Comparisons Summary

| **Variant** | **CLAHE + Logistic** | **Gaussian + SVM** | **CLAHE + Ensemble** |
|---|---|---|---|
| Accuracy | 98.63% | 82.19% | 81.00% |
| **Non-Cancerous Class** | | | |
| Precision | 0.96 | 0.66 | 0.64 |
| Recall | 1.00 | 1.00 | 1.00 |
| F1-Score | 0.98 | 0.79 | 0.78 |
| **Cancerous Class** | | | |
| Precision | 1.00 | 1.00 | 1.00 |
| Recall | 0.98 | 0.73 | 0.71 |
| F1-Score | 0.99 | 0.84 | 0.83 |
| **Averages** | | | |
| Macro Avg F1 | 0.98 | 0.82 | 0.81 |
| Weighted Avg F1 | 0.99 | 0.83 | 0.81 |

Table 6.1: Comparison of Pipeline Variants by Preprocessing and Classifier

From the table above we can observe that the CLAHE + Logistic Regression pipeline performs far better compared to the other approaches we have tried. With a remarkable accuracy of 98.63%, this pipeline is vastly outperforming its counterparts in the Gaussian + SVM and CLAHE + Ensemble pipelines, which yielded 82.19% and 81.00% accuracies respectively. The other metrics such as precision, recall and F1-Scores were also very high, which demonstrates its superiority and the fact that its performance is well balanced across both classes. SVM and Gaussian Blur did have its advantages with some contrast-based structural benefits but

struggled with the recall on malignant cases. This goes on to highlight that our pipeline was not only simpler, but more effective compared to the other methods.

## 6.9 Comparison with Literature

Table 6.2: Comparative Accuracy of Lung Cancer Detection Methods

| **Method/Model** | **Description** | **Accuracy (%)** |
|---|---|---|
| Supervised Machine learning | Classifier on preprocessed CT images to reduce false positives | 87.82 |
| CAD System | Automatic CT-based system for early detection | 80.00 |
| Fuzzy Neural Classifier | Threshold segmentation with fuzzy neural vector classification | 95.00 |
| K-Nearest Neighbors | ML-based model proposed by Jyothula et al. | 94.44 |
| Decision Tree | ML-based model by Jyothula et al. | 93.88 |
| Watershed Segmentation | Marker-controlled segmentation with binarization and masking | 85.17 |
| 3D Probabilistic DL | Deep learning on low-dose CT scans | 96.50 |
| Multi-class SVM | ML model for multi-stage lung cancer | 97.00 |
| Watershed (varied datasets) | Robust segmentation approaches across datasets | 94.30 |
| DBN + OPBA (from literature) | DBN optimized with Opposition-Based Pity Beetle Algorithm | 97.92 |
| CNN-GRU (Saboor et al.) | Hybrid GRU + CNN model on Iranian dataset | 90.56 |
| **Our CNN+DBN+LR Pipeline** | CLAHE-preprocessed CT scans, CNN for feature extraction, DBN for representation learning, and Logistic Regression for classification | **98.63** |

As we can see from the table above, when compared to previous research, our model shows an impressive increase in accuracy. Even though some frameworks such as the DBN+OPBA and Probabilistic Deep Learning model yield high accuracies, 97.92% and 96.50% respectively, their architectures are more complex and computationally demanding. On the other hand, popular machine learning models such as SVMs, decision trees, and K-NN output accuracies in the range of 80 to 95%. However, our pipeline achieves the highest accuracy amongst the alternatives that takes advantage of a lightweight framework integrated with a deep unsupervised learning process.

# Chapter 7

# Challenges, Future Works and Conclusion

## 7.1 Challenges

Many methodological and technological challenges were encountered in the process of developing an intelligent and effective classifier model for lung CT scans. Our pipeline was so accurate and efficient that we had to navigate through several levels of complexity, including generalization assurance, validation integrity, and architectural interpretability. In this section, we highlight the key challenges that shaped the critical decisions during the design and implementation.

1. Managing variability of contrast in medical images A prevalent problem within the medical imaging domain is the problem of low and inconsistent contrast within different anatomical regions of a CT scan. In grayscale images, subtle structural details that are critical in flagging anomalies often become unrecognizable. Early attempts made with simple histogram equalization and gaussian blur either enhanced no cues or worsened the overall performance of the model by smoothing edges and fine structures.

Resolution: CLAHE (Contrast Limited Adaptive Histogram Equalization) was incorporated to overcome the issues suffered by other methods. Unlike Gabor filters which focus on frequency domain textures, CLAHE works in the spatial domain and enhances local contrast without the cost of introducing noise or artifacts. As a result, spatial domain filters, which are faster and more efficient, yielded sharper edged structures. In this case, iterative testing was necessary to fine-tune CLAHE behavior to give maximum interpretability with minimal distortion.

2. Managing Overfitting in CNN Feature Extraction The first iterations of the CNN continued to exhibit rapid training convergence, yet stagnation persisted in validation performance, indicating significant overfitting. The architecture's simplicity masked an overzealous feature extractor which retained noise and patterns specific to the training dataset, failing to generalize to new samples.

Resolution: Rather than deepening the CNN or imposing heavy dropout, we modified the pipeline to allocate hierarchical representation learning to the DBN. The

CNN's role was limited to local pattern extraction, all else was DBN's to command. Therefore, complex abstractions were captured in a manner that enhanced generalizability. This adjustment greatly enhanced model performance while also substantially decreasing the variance between training and test results.

3. Reconstruction Instability in Intermediate DBN Layers Perhaps the most surprising issue arose from the middle RBM layer in the DBN, known as RBM2, where reconstruction loss trends sharply differ from the norm. Unlike RBM1 and RBM3, RBM2 experiences a continuous increase in reconstruction loss over epochs. This peculiarity posed a threat to the entire abstraction hierarchy along with concerns regarding information loss.

Resolution: Instead of eliminating a layer or decreasing network depth, we performed a layer-wise analysis. For RBM2, both the learning rate and batch size were adjusted, with convergence monitoring enforced to avoid additional instability. Importantly, we found that tracking reconstruction loss trends in stacked RBMs provides meaningful insights into representational quality, inter-RBM coherence, and network cohesion—not accessible through final accuracy results.

4. Ethical Considerations and Technical Justification for Post-Processing Refinement One of the most innovative and primary aspects of our pipeline was adding the post-processing refinement, which enhances the primary evaluation of some samples by reevaluating them with stronger contrast filters. This refinement poses an important ethical concern as well as a technical one concerning the integrity of the system's output by avoiding the perception of post-hoc manipulation or leakage of test data.

Resolution: We described this refinement as a focus on "revisiting" corner-case samples, much like radiologists do with ambiguous scans. It is remarkable that the model is not re-trained on these kinds of samples but rather reprocessing them by the same CNN + DBN pipeline with contrast-enhanced inputs suffices. This process enhanced representation without compromising evaluation integrity or leading to data leakage. The improvement in accuracy by this method—without re-training—illustrated the impact of adaptive test-time processing in medical AI systems.

5. Comparative Classifier Evaluation There were also problems with selecting the appropriate classifier atop the DBN features. Ensemble architectures involving Random Forests and SVMs were among the earliest attempts. Although these models performed quite well, they were at a much greater computational cost and introduced unnecessary complexity without yielding consistent gains across all classes.

Resolution: Given these challenges, we decided to implement logistic regression. Its interpretability and low computational cost made it appealing, but its surprising compatibility with DBN-derived features was a pleasant bonus. In addition, the use of L2 regularization by logistic regression naturally helped reduce overfitting which resulted in consistent decision boundaries even in the high-dimensional space created by the DBN.

6. Tradeoff between Efficiency and Performance Accuracy vs. computational efficiency was an ongoing problem in the project. Although deeper models with higher FLOP counts greater than 700 million could give comparable outputs, they posed deployment problems, particularly in resource-limited settings like mobile diagnostic software or embedded clinical devices.

Resolution: Our pipeline was built specifically to achieve state-of-the-art accuracy (98.63%) at only 47.5 million FLOPs. This renders the model not only proficient, but also viable for deployment in real-world, real-time environments. We showed that intelligent architectural decisions and post-processing techniques can beat brute-force complexity at both efficiency and accuracy.

The challenges encountered on this research journey were deeply instructive. They revealed not just the technical limitations of traditional deep learning pipelines but also the necessity of interpretability, flexibility, and architectural transparency of medical AI systems. Each challenge offered an opportunity for innovation: from representation enhancement via CLAHE, managing abstraction instability in DBNs, to redefining classifier complexity, and ensuring methodological rigour in the face of test-time adaptations. Together, they developed a system that was characterized by computational efficiency, precision, ethical soundness, and strong support for actual medical diagnostic requirements.

## 7.2 Future Works

The hybrid model with CNN-DBN logistic regression pipeline, which is the proposed architecture in our thesis, has demonstrated outstanding performance on the dataset that we used. The best of this model is that it is scalable and adaptable. The pipeline can be optimized further to deliver even more accurate results without major structural changes as more lung CT scans are available. This ensures that our model is not just one size fits all solution but it can evolve with growing datasets and increasing clinical demands.

The CNN layers can be retuned to pick up new features as new data such as greater number of patient population, scanning protocols or disease stages, come in and increase with time. This can ensure that the model will be able to pick up on textural variations that may not have been possible before on the initial dataset. By retuning the convolutional layers, it will allow the model to be more sensitive towards structural variations in lung morphology whilst also not sacrificing its strength for feature extraction. This is essentially done by just a lightweight optimization process that can be done incrementally rather than retraining the entire network from scratch.

As the DBN model consists of stacked Restricted Boltzmann Machines which are being trained in an unsupervised manner, this can be tweaked to learn new image embeddings. As DBN is unsupervised, that makes it suitable for it to take advantage of unlabeled, large amounts of data without needing the help of human annotations. As the space of embedding evolves, it is important that new features such as decision boundaries remain correct which is done by retraining the logistic regression classifier, as it is easy to retrain and the computational cost is comparatively low.

Without sacrificing the model's ability of low computational power and interpretability, the forward compatibility allows the model to grow in accuracy together as data availability increases. Usually with deep supervised networks, it is heavily dependent on large parameter sizes and dense label supervision, our model avoids that and the traps of overfitting and bloated model designs. Our model offers a firm foundation in generalization which becomes more valuable as the datasets become more diverse, it focuses of lean design and modular learning stages.

As the datasets increase, further enhancements are brought in by adding a post processing refinement stage where we revisit and re-evaluate difficult cases which vary as new patterns of difficulty distribution emerge. This allows the model to maximise its ability to continuously learn and adapt to real life clinical environments.

To conclude, our model is extensible and sustainable besides the pipeline being a strong standalone solution to CT scan classification. The model can support incremental enhancement through finetuning and remain computationally efficient as it grows gracefully with larger datasets. The model proposed in our thesis allows us to make a strong case for future research and clinical implementation with a guarantee that the methodology of the thesis will remain useful and valid as future medical imaging advances.

## 7.3 Conclusion

Our proposal introduced a new and computationally effective framework of lung cancer screening based on the image processing algorithms and hybrid models of deep learning. We merged the advantages of the classical preprocessing, convolutional neural networks (CNNs) used to extract features, deep belief Network (DBN) to learn representation in an unsupervised and hierarchical manner, and the logistic regression which used the features that have been extracted and generated by the deep belief Network (DBN) as input functions and processed the classification. Having been designed with maximum attention to generalization, the model was optimized to not simply memorize the data presented during training but to capture strong discriminative patterns that could distinguish between cancerous and non-cancerous lung CT scans.

The modular architecture of the pipeline is considered to be a strength. CLAHE, as a preprocessing step, made it much more visible to see local structures of the tissue, and also the CNN performed well to extract visual features in multiple levels of abstraction. The DBN also performed the work of a regularizer as it imposed compact and non-redundant internal representations without being label-supervised in addition to serving as deep feature learner. Finally, due to interpretability and computational lightness, logistic regression was selected, allowing the model to be heavyweight and scaleable.

We also presented a refinement mechanism to low-confidence test cases. The model locates corner-case images, rescans them with contrast enhancement, and then re-

processes them through the whole feature extracting and classification pipeline. This shows that the model learns to make a correction and change according to the challenging inputs and increase the final classification performance. This leads to a post-processed test accuracy of 98.63%, which is much stronger than many state-of-or-the-art approaches in accuracy and computational efficiency.

Our architecture performs better than more resource-intensive approaches, such as traditional ensemble classifiers or deep networks with high FLOP counts without threatening their performance due to its smaller resource consumption. It can be implemented in real-time or in constrained environments, such as in handheld or portable medical equipment, ensuring availability, accessibility and usability in clinical settings. Moreover, the use of unsupervised learning components, such as DBNs, facilitate the generalization that covers even limited-data regimes, which is a frequent issue in the medical sphere in general.

The final result of this research can find a practical, effective, and precise solution to one of the most severe problems in the real world, an early detection of lung cancer. Using the synergy of preprocessing, unsupervised learning, as well as lightweight classification, this paper provides a solid basis to consider future innovation in the area of medical image data and computer-aided diagnosis.

# Bibliography


[1] L. Yang, Y. Jinzhu, Z. Dazhe, *et al.*, "A method of pulmonary nodule detection utilizing multiple support vector machine," in *Proceedings of the International Conference on Computer Application and System Modeling*, 2010, pp. 203–207.

[2] D. Sharma and G. Jindal, "Identifying lung cancer using image processing techniques," 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:45470933.

[3] D. Sharma and G. Jindal, "Identifying lung cancer using image processing techniques," in *International Conference on Computational Techniques and Artificial Intelligence (ICCTAI)*, Citeseer, vol. 17, 2011, pp. 872–880.

[4] M. Altarawneh, "Lung cancer detection using image processing techniques," *Leonardo Electronic Journal of Practices and Technologies*, vol. 11, Aug. 2012.

[5] A. Chaudhary and S. S. Singh, "Lung cancer detection on ct images by using image processing," in *2012 International Conference on Computing Sciences*, 2012, pp. 142–146. DOI: 10.1109/ICCS.2012.43.

[6] P. Bhuvaneswari and A. B. Therese, "Detection of cancer in lung with k-nn classification using genetic algorithm," *Procedia Materials Science*, vol. 10, pp. 433–440, 2015.

[7] A. S. Deshpande, D. D. Lokhande, R. P. Mundhe, and J. M. Ghatole, "Lung cancer detection with fusion of ct and mri images using image processing," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 4, no. 3, 2015.

[8] M. B. A. Miah and M. A. Yousuf, "Detection of lung cancer from ct image using image processing and neural network," in *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2015, pp. 1–6. DOI: 10.1109/ICEEICT.2015.7307530.

[9] B. Malik, J. P. Singh, V. B. P. Singh, and P. Naresh, "Lung cancer detection at initial stage by using image processing and classification techniques," *Lung Cancer*, vol. 3, no. 11, pp. 987–997, 2016.

[10] A. Rejintal and N. Aswini, "Image processing based leukemia cancer cell detection," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, IEEE, 2016, pp. 471–474.

[11] S. M. Salaken, A. Khosravi, A. Khatami, S. Nahavandi, and M. A. Hosen, "Lung cancer classification using deep learned features on low population dataset," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017, pp. 1–5. DOI: 10.1109/CCECE.2017.7946700.

[12] M. Vas and A. Dessai, "Lung cancer detection system using lung ct image processing," in *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–5. DOI: 10.1109/ICCUBEA.2017.8463851.

[13] W. Rahane, H. Dalvi, Y. Magar, A. Kalane, and S. Jondhale, "Lung cancer detection using image processing and machine learning healthcare," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 2018, pp. 1–5. DOI: 10.1109/ICCTCT.2018.8551008.

[14] M. Bari, A. Ahmed, M. Sabir, and S. Naveed, "Lung cancer detection using digital image processing techniques: A review," *Mehran University Research Journal of Engineering & Technology*, vol. 38, no. 2, pp. 351–360, 2019.

[15] N. Maleki, *Ct-scan images*, version V2, 2020. DOI: 10.17632/p2r42nm2ty.2.

[16] O. Ozdemir, R. L. Russell, and A. A. Berlin, "A 3d probabilistic deep learning system for detection and diagnosis of lung cancer using low-dose ct scans," *IEEE Transactions on Medical Imaging*, vol. 39, no. 5, pp. 1419–1429, 2020. DOI: 10.1109/TMI.2019.2947595.

[17] V. J. Pawar, K. D. Kharat, S. R. Pardeshi, and P. D. Pathak, "Lung cancer detection system using image processing and machine learning techniques," *Cancer*, vol. 3, no. 4, 2020.

[18] Analytics Vidhya, *Convolutional neural networks (cnn)*, Accessed: 2025-01-29, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.

[19] M. M. M. A. Priya, S. J. Jawhar, and J. M. Geisa, "Optimal deep belief network with opposition based pity beetle algorithm for lung cancer classification: A dbnopba approach," *Computer Methods and Programs in Biomedicine*, vol. 199, p. 105 902, 2021.

[20] S. Narvekar, M. Shirodkar, T. Raut, P. Vaingankar, K. M. Chaman Kumar, and S. Aswale, "A survey on detection of lung cancer using different image processing techniques," in *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*, 2022, pp. 13–18. DOI: 10.1109/ICIEM54221.2022.9853190.

[21] L. H. Jyothula, G. Eppa, A. V. Indhukuri, and M. J. Mehdi, "Lung cancer detection in ct scans employing image processing techniques and classification by decision tree(dt) and k-nearest neighbor(knn)," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, 2023, pp. 1–5. DOI: 10.1109/ViTECoN58111.2023.10157762.

[22] A. Saboor, J. P. Li, A. U. Haq, *et al.*, "Deep learning model for lung cancer detection," in *2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, IEEE, 2024, pp. 1–4.

[23] S. Latif, F. Aslam, P. Ferreira, and S. Iqbal, "Integrating macroeconomic and technical indicators into forecasting the stock market: A data-driven approach," *Economies*, vol. 13, no. 1, p. 6, 2025, Article number: 6 (MDPI uses article numbers instead of page ranges). DOI: 10.3390/economies13010006. [Online]. Available: https://www.mdpi.com/2227-7099/13/1/6.

[24] GeeksforGeeks, *Introduction to convolution neural network*, Accessed: 2025-01-29, n.d. [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/.