Detection of Common Thorax Diseases from X-Ray Images using a Fusion of Transfer and Statistical Learning Method

by

Ahmad Abdur Rafi
19101023
Muhtasim Mahmud
22241151
Sakib Dewan Pranto
19101015
Sayemur Rahman
19101210
Shihab Rumee Chowdhury
18201064

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
School Of Data & Science
Brac University
May 2023

© 2023. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing our 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 that 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:

Ahmad Abdur Rafi
19101023

Mahmud

Muhtasim Mahmud
22241151

Sakib

Sakib Dewan Pranto
19101015

Sayemur Rahman
19101210

Shihab

Shihab Rumee Chowdhury
18201064

Approval

The thesis titled “Detection of Common Thorax Diseases from X-Ray Images using a Fusion of Transfer and Statistical Learning Method” submitted by

1. Ahmad Abdur Rafi(19101023)
2. Muhtasim Mahmud(22241151)
3. Sakib Dewan Pranto(19101015)
4. Sayemur Rahman(19101210)
5. Shihab Rumee Chowdhury(18201064)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 25, 2023.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

We, the authors, now state that the research presented in this thesis is original and accurate, including correct citations for all other sources. We are also dedicated to respecting ethical standards by applying for informed permission, protecting privacy, adhering to all applicable laws and regulations, being professional and ethical, and resolving any biases or conflicts of interest that may arise. On top of that, no other academic institution has ever been approached with the submission of this paper, either in its entirety or in any of its parts, to receive credit toward a degree.

Abstract

An essential component of medical diagnosis is the precise detection and localization of anomalies in X-rays of the chest images. It is urgently necessary to develop the most precise automated model to identify thorax diseases because the number of patients with thorax diseases is rising worldwide. In order to build a reliable prediction model for such tasks, experts will need to manually label a sizable dataset of X-ray images. Nevertheless, more data is needed to build exact models to detect these diseases automatically. As a result, we're committed to creating a model that detects the anomalies from thorax X-rays automatically, learning from a small amount of X-ray image data that is publicly available and easy to get. To do so, we propose a fusion model by combining transfer learning and statistical learning methods. The comparative reference baseline was significantly outperformed. We show that the detection of thorax diseases can be improved by using our fusion model, allowing quicker diagnosis and treatment.

Keywords: Thorax Diseases; X-ray images; Annotation; Transfer learning; Machine Learning; Fusion model

Dedication

This thesis is a dedication to our parents, who have always been there for us and given us so much love and encouragement. Their belief in us and willingness to make sacrifices have been invaluable. Without their unwavering backing, we would never have made it this far.

Acknowledgement

All thanks are due to the merciful almighty God, the most exalted, the most helpful, and the most Merciful, who has allowed us to study at BRAC University. To pursue our bachelor's degrees would not have been feasible without the help of countless others. We would like to take this opportunity to publicly thank our supervisor, Md. Golam Rabiul Alam, Sir, for all of the invaluable feedback and direction he has provided us with thus far.

Finally, we must thank our parents, without whom none of this would have been possible. Their words of encouragement and prayers have helped us come this far, and we will soon be able to celebrate our accomplishments by obtaining our degrees.

Table of Contents

Declaration i

Approval ii

Ethics Statement iii

Abstract iv

Dedication v

Acknowledgment vi

Table of Contents vii

List of Figures ix

List of Tables x

Nomenclature xi

1 Introduction 1
- 1.1 Research Problem 2
- 1.2 Research Contributions 3

2 Literature Review and Related Work 4
- 2.1 Related Works 4
- 2.2 Background Study 11
  - 2.2.1 Common Thorax Diseases 11
  - 2.2.2 Chest X-ray 14

3 Methodology 16
- 3.1 Top Level Overview of the Proposed Method 16
- 3.2 Dataset 17
  - 3.2.1 Dataset Acquisition 17
  - 3.2.2 Exploratory Data Analysis 18
  - 3.2.3 Data pre-processing 19
- 3.3 Model Specification 21
  - 3.3.1 CNN Transfer Learning Models 21
  - 3.3.2 Machine Learning Models 29
- 3.4 Proposed Fusion Model 30

3.4.1 Combining the Features from Two CNN Models 31
3.4.2 Using KNN Classifier 31
3.4.3 Using CatBoost Classifier 32
3.4.4 LIME Explainability 33

4 Result Analysis and Discussion 34
4.1 Inception-ResNet-v2 Implementation and Results 35
4.2 MobileNet Implementation and Results 36
4.3 VGG-16 Implementation and Results 37
4.4 DenseNet-121 Implementation and Results 38
4.5 Fusion Model Implementation and Results 39
4.5.1 Combining Features from Two CNN Models 39
4.5.2 KNN Implementation and Results 39
4.5.3 CatBoost Implementation and Results 40
4.6 Comparative Study 41
4.7 Explainable AI 44

5 Conclusion 45
5.1 Limitations 45
5.2 Recommendation and Future Work 46

Bibliography 48

List of Figures

2.1 Cardiomegaly X-ray Images 11
2.2 Infiltration X-ray Images 12
2.3 Nodule X-ray Images 13
2.4 Pneumonia X-ray Images 14

3.1 Top Level Overview of the Proposed Method 17
3.2 Dataset Samples 18
3.3 Dataset X-ray Image Count Visualization 19
3.4 Before and After Applying CLAHE Algorithm 20
3.5 Schema of Basic CNN Architecture 23
3.6 Schema of Inception-ResNet-v2 Architecture Baseline 24
3.7 Schema of Inception-ResNet-v2 Architecture Layers 25
3.8 Schema of MobileNet Architecture 26
3.9 Schema of VGG-16 Network 27
3.10 Schema of DenseNet-121 Network Architecture 28
3.11 Extracting and Combining the features from two CNN models 31
3.12 Train the KNN and CatBoost Model by the Training and Val Features 32
3.13 Process of Prediction using KNN and CatBoost Model 32

4.1 Training and Validation Loss & Accuracy Curves of Inception-ResNet-v2 35
4.2 Confusion Matrix of Inception-ResNet-v2 36
4.3 Training and Validation Loss & Accuracy Curves of MobileNet 36
4.4 Confusion Matrix of MobileNet 37
4.5 Training and Validation Loss & Accuracy Curves of VGG-16 37
4.6 Confusion Matrix of VGG-16 38
4.7 Training and Validation Loss & Accuracy Curves of DenseNet-121 . . 39
4.8 Confusion Matrix of DenseNet-121 39
4.9 Confusion Matrix of KNN Combining Features from CNN Models . . 40
4.10 Confusion Matrix of CatBoost Combining Features from CNN Models 41
4.11 Explainable AI for each CNN Model we used 44

List of Tables

3.1 Dataset X-ray Images Count for Each Disease 19
3.2 X-ray Images Sample Count After Splitting for Each Disease 20

4.1 Inception-ResNet-v2 Precision, Recall and F1-Score 35
4.2 MobileNet Precision, Recall and F1-Score 36
4.3 VGG-16 Precision, Recall and F1-Score 37
4.4 DenseNet-121 Precision, Recall and F1-Score 38
4.5 Precision, Recall and F1-Score After Implementing KNN 40
4.6 Precision, Recall and F1-Score After Implementing CatBoost 40
4.7 Comparing Our Model to Others 42

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
$CatBoost$	Machine Learning classifier
CNN	Convolutional Neural Network
KNN	K-Nearest Neighbor
ML	Machine Learning
$NIH\ Clinic\ X-ray$	Human Chest X-ray

Chapter 1

Introduction

The heart, lungs, major blood vessels, and other essential organs are located in the thorax, which is an important part of the body. Numerous diseases, such as Pneumonia, Cardiomegaly, Nodules, and Infiltration illnesses, can have an impact on the thorax. If not identified and treated promptly, these disorders can cause considerable morbidity and mortality. Our study focuses on the automated detection of 4 essential categories of thoracic illnesses. For determining the importance and urgency of developing automated diagnostic systems, it is crucial to comprehend the statistics and effects of thoracic disorders. Here are some important details about specific thoracic diseases:

Cardiomegaly, which is marked by an enlarged heart, is frequently an indication of underlying cardiac issues. Depending on the demographic being investigated and the individual risk factors present, Cardiomegaly prevalence varies. If untreated, it is linked to higher rates of morbidity and mortality. An estimated 18 million Americans aged 20 and over are impacted by Cardiomegaly. After Infiltration Diseases, Infiltration conditions like pulmonary fibrosis and sarcoidosis can decrease lung function and cause progressive lung damage. To stop future complications and enhance the patient's prognosis, prompt diagnosis, and treatment are essential. Another type is Nodule and Mass. Detecting lumps and nodules in the thorax is important because they may be signs of benign or malignant illnesses, including lung cancer. Treatment choices and patient outcomes are substantially impacted by early discovery and correct characterization of these anomalies. The most happening and common thorax disease is Pneumonia. It is also the main killer of elderly people and those with compromised immune systems. These figures emphasize the significance of creating effective and precise automated systems for identifying and categorizing these widespread thoracic disorders. These figures emphasize the significance of creating effective and precise automated systems for identifying and categorizing these widespread thoracic disorders.

Medical imaging has grown in importance as a diagnostic and treatment tool in recent years. Among the several imaging modalities, X-ray imaging is frequently used to examine the thorax since it is accessible, affordable, and able to produce data quickly. It enables the identification and assessment of different thoracic disorders by offering insightful information about the composition and health of the chest. To find anomalies and make a diagnosis, radiologists examine X-ray pictures. However, interpreting X-ray pictures can be difficult and arbitrary, necessitating knowledge and experience. Additionally, radiologists are under a great deal of stress as a result of the growing amount of medical imaging data, which causes delays in diagnosis and treatment. But in this situation, time is quite important. An essential part of providing successful medical therapies and improving patient outcomes is the timely, accurate diagnosis of these disorders.

To diagnose prevalent thoracic disorders from X-ray pictures, we, therefore, suggested a unique method that combines transfer learning and statistical learning techniques. Utilizing the best aspects of both methods, this technique fusion attempts to increase accuracy and resilience. Our suggested solution can help radiologists make rapid and accurate diagnoses by automating the detection process, easing the strain on healthcare systems, and enhancing patient care.

1.1 Research Problem

Due to a number of reasons, thorax disease detection is challenging. Pneumonia, cardiomegaly, masses, nodules, and infiltrations are all examples of thorax disorders. Each of these disorders has its own distinct appearance in imaging studies, making diagnosis difficult. Accurately annotating and categorizing these illnesses from the X-ray photos through the observation of subtle patterns and abnormalities requires specialized knowledge and expertise. Thus, one of the main issues is this. The creation of precise and accurate techniques that can successfully evaluate medical images, particularly X-ray images, in order to detect and classify diverse thorax disorders, is another major research challenge in the automatic detection of thorax diseases. The difficulties linked to this issue include:

1. Image Interpretation: Thorax X-ray interpretation is difficult since it calls for expert familiarity with subtle irregularities and patterns that may indicate certain disorders. Automated models must carefully examine and interpret these images, collecting correct characteristics and patterns in order to distinguish normal from abnormal circumstances.

2. Variation: The signs and symptoms of thoracic diseases can vary widely, including changes in size, shape, position, and texture. For high accuracy in detection and classification, it is essential to develop robust models that can manage a wide range of disease presentations.

3. Class Imbalance: In a dataset, the distribution of various thoracic diseases is frequently unbalanced, with some conditions being more common than others. Due to insufficient training data, the neural network model may fail to effectively detect and categorize rare diseases as a result of the class imbalance.

4. Generalization to Unseen instances: A thorax illness detection system that works well should be able to adapt to undiagnosed instances and a variety of patient demographics. Instead of recalling specific cases from the training data, the model must capture the underlying patterns and traits of the diseases.

5. Interclass Confusion: Due to similarities in radiological features, distinguishing between certain thoracic disorders can be challenging. It is necessary to use a sophisticated robust CNN model that can accurately capture small distinctions and classify diseases with overlapping characteristics in order to distinguish between them.

Comprehending these research challenges will necessitate the creation of modern machine learning and computer vision methods, including deep learning, transfer learning, and statistical learning techniques. The objective is to construct a robust deep-learning algorithm that can accurately distinguish between identical diseases and extract significant features, handle differences in disease presentations, handle imbalanced data, and generalize well to instances that haven't been seen before. The goal of the discipline of automatic thorax disease detection is to advance thoracic imaging patient care by addressing these research issues, which seek to increase diagnostic precision, enable early diagnosis, help treatment planning, and finally promote early detection.

1.2 Research Contributions

Millions of people are affected by thorax disorders, which place a huge burden on global health. Thorax diseases require an accurate and prompt diagnosis in order to receive appropriate treatment. Thorax illnesses are detected and assessed using radiological imaging techniques, such as X-ray imaging. However, due to the time and human error involved, manual interpretation of these images is often avoided. As a consequence, creating autonomous systems for diagnosing and categorizing thoracic disorders has garnered a lot of attention in recent years. This project's primary objective is to develop an autonomous model that can identify four common thoracic disorders using a mix of transfer learning and statistical learning. The data that has been enhanced using Image Data Generator will be part of the training set for this model. The study's objectives are to:

1. Create a solid model for the identification and classification of thoracic disorders by combining transfer learning and machine learning methods.

2. Conduct an evaluation of the efficacy of different pre-trained deep learning models, including MobileNet, Inception-ResNet-v2, VGG-16, and DenseNet-121, and subsequently determine the optimal model.

3. Analyze the efficacy of different classifiers to enhance the precision and accuracy of the classification outcomes by using the classifier to combine the extracted features.

4. To test the framework on an extensive collection of thorax X-rays from patients with different diseases and demographics.

5. Finally, we are offering any changes and upgrading of our model to detect thorax diseases more accurately.

Chapter 2

Literature Review and Related Work

2.1 Related Works

To better the diagnostic accuracy and openness of cardiomegaly, the authors of this paper [1] propose a system that combines ResNet-based deep learning algorithms with explainable feature maps. By training the model on an extensive set of X-ray images from the chest, it is able to assess the condition with high precision. The explainable feature map provides visual evidence for the neural network's decision-making process, ensuring accurate judgment results. Using AI algorithms, the method is cost-effective, accessible, and yields precise results. It demonstrates the usefulness of chest X-ray imaging and explainable feature maps in facilitating disease diagnosis. The proposed method addresses the limitations associated with relying solely on CNN for accurate judgments. It aids in the analysis of the internal structure of neural networks and can be applied to diseases with similar appearances. However, data resolution, quality, and diversity, as well as the scope of learning data and evaluations, require further refinement. In addition, deeper neural network designs and hyperparameter adjustments require additional research. Integrating intelligent systems that can be explained with medical support systems is essential for future advancements in medical intelligence. In addition, the study highlights the significance of configuring and evaluating neural networks for optimal performance. Different functions, learning rates, and network designs have a substantial effect on precision. SGD achieves the highest precision, while AdaGrad and RMSProp achieve lesser precision. The explainable feature map provides visual interpretations of the factors influencing disease judgments, thereby assisting medical professionals in comprehending and analyzing the neural network's decision-making process. Adjusting input pixel sizes improves the display of favorable factors. The final impact analysis incorporates all factors into a red image that is centered on the heart. This analysis provides additional information regarding the position, progression, and severity of the disease. This work combines explainable feature maps with ResNet-based deep learning algorithms to correctly and transparently identify cardiomegaly. It illustrates the effectiveness of the method and emphasizes the importance of data quality and variety using chest X-ray imaging. In addition, the study highlights the need for more intricate neural network designs and hyperparameter adjustments. Integrating intelligent systems with medical support systems is essential for the future advancement of medical intelligence. After 50 epochs of analysis, we found that SGD and Adam's results are very close to 80% accurate. AdaGrad and RMSProp achieve an accuracy of approximately 75% after 100 epochs, whereas Adam and SGD achieve an accuracy of 75% within 60-80 epochs.

The study [2] article, it is discussed how to identify pneumonia in X-ray pictures using convolutional neural networks (CNNs). The research presents six models that can accurately detect pneumonia using CNNs. For all six models, the identical data pre-processing method was used. Two models with two convolutional layers and four pre-trained models (VGG16, VGG19, ResNet50, and Inception-v) were employed to determine whether or not an X-ray image depicted pneumonia. Pre-trained models showed accuracy rates of 87.28%, 88.46%, 77.56%, and 70.99%, compared to the first two models' validation accuracy of 85.26% and 92.31%. The amount of the dataset utilized affects the model accuracy, which is enhanced by using bigger datasets. There were three different ways in which the efficacy of the model was evaluated: accuracy, recall, and the F1 Score. In this study, two CNN models and transfer learning models were tested. The model outputs were evaluated using the confusion matrix, which revealed the flaws in the classifier. In order to determine the CNN models' recall and F1 Score, the confusion matrix was used. On each of the three performance parameters, Model 2 topped 90%. Due to its superior classification accuracy and F1 Score values, VGG19 outperforms all other Transfer Learning models. Although it has a lower recall than VGG16, VGG19 performs better overall. ResNet50 and Inception-v3, on the other hand, exhibit a significant disparity between their training and validation accuracy, which suggests significant overfitting and, thus, subpar performance. Deep neural networks are anticipated to perform better with bigger datasets; however, owing to the lesser size of the dataset utilized in this research, they presently have a lower validation accuracy.

The authors of this paper [3] present an automated method for calculating and detecting the cardiothoracic ratio. It's a way to calculate the cardiothoracic ratio using X-rays of the chest. A VGG16 encoder and U-Net deep learning model isolate lung and heart masks from X-rays. The final dataset was split into 90-10, for train and validation, which includes lung samples and heart samples. The X-ray images are used to compute the cardiothoracic ratio, which is derived by dividing the heart diameter using the thoracic diameter. After collecting the heart and lung masks, the cardiac diameter is determined by determining the highest points on the heart mask's x-axis and their distances from the x-axis. After that, from the lung mask's extreme points, the thoracic diameter is computed. A CTR reading of less than 0.5 is regarded as normal, whereas one between 0.5 to 0.55 indicates moderate cardiomegaly, and one beyond 0.55 indicates cardiomegaly. The cutoff point for determining cardiomegaly in this investigation is a ratio of 0.50. The CTR value distribution revealed that false positive samples occur more often in the 0.5–0.6 range, indicating that pictures in this area need re-evaluation. There were a significant number of instances with moderate cardiomegaly that were discovered but were not labeled in the dataset, indicating that the labels could be noisy. Radiologists who need to review several X-ray films each day find the manual approach of taking these measures using Picture Archiving and Communication Systems (PACS) to be time-consuming and error-prone. By employing the automated tools, radiologists may save a significant amount of time and labor since the algorithm's performance was assessed by real radiologists and proved to be correct in 76.5% of instances. Three image segmentation models were evaluated for their ability to differentiate between the heart and lungs in chest X-rays. Automating CTR computation has been attempted using image processing, deep learning, and image segmentation. In two different investigations, deep learning for image segmentation produced great outcomes. As a result, this method has a 76.5% acceptance rate in real-world contexts and may be included in a CTR evaluation tool. Even with a small amount of training data, the deep learning method has a good degree of accuracy and can be enhanced with additional samples.

The challenges of using machine learning in medical diagnostics, specifically for analyzing chest X-rays, are discussed in the paper [4]. The model is trained to identify application-specific characteristics and exploit statistical dependencies between labels. This approach outperforms the modern cutting-edge, and the set of metrics provides meaningful quantification of the performance. The authors used a short training set to tune three different types of models and data augmentation was employed to prevent overfitting. Both models have nearly the same number of parameters and are trained with the Adam optimizer and early halting. The models are restricted to contain the same number of parameters, but they have different configurations depending on whether labels are viewed as independent or dependent. The impact of the order on the factorization of the models is investigated. The dataset used in the study's experiments was one of 112,120 chest X-rays that were made public by Wang et al. (2017). Training, validation, and testing data for 14 anomalies are randomly divided into three sets. Wang et al. (2017) found that there were only small differences in performance between the validation and test sets when using different random splits, indicating that the two sets behaved consistently. Their detection rate for Pneumonia, Nodule, Infiltration, and Cardiomegaly is gradually 0.71, 0.72, 0.70, and 0.91. The article's introduction discusses the difficulties of using machine learning for medical diagnostics, particularly in analyzing chest X-rays. The authors highlight the challenge of simultaneously predicting numerous labels while accounting for their conditional dependencies. To make the most of the data now available, they suggest a new architecture that takes the clinical context into account. The authors speculate that when there is enough medical data available, pre-training on other datasets might not be required. Along with reporting alternative measures in addition to conventional machine learning metrics in their benchmark, they also address the problem of clinical interpretability. Overall, the research makes a compelling case for using machine learning, especially in the processing of chest X-rays, to address the difficulties in medical diagnoses. The suggested model performs better than the one and offers accurate performance measurement. To investigate the possibility of learning label interdependencies and prevent biased learning from a small training set, additional research is required. The optimal strategy probably entails learning from material that has been annotated with an ontology that specifies some regular, well-known relationship structure.

The study [5] suggests a novel method for diagnosing illnesses from medical images that combines multi-resolution and multi-instance learning with a specially designed pooling function. The suggested method uses ResNets and DenseNets to create saliency maps with a resolution of 64x64 using input photos and a binary vector of the total class number. The work creates the highest-resolution saliency maps to date while generating ground-breaking results on 9 anomalies. The suggested method gives pathology-based saliency maps with higher resolution, which boosts diagnostic precision and solves the problem of localizing abnormalities of various sizes with just image-level labels. The paper suggests pre-training the proposed model on data from a different domain for improved performance in the future while using the LSE-LBA sharpness prior approach to improving the localization of anomalies. In order to increase computer performance, the study used data augmentation and downsampled the inputs to 512 × 512 during training. Models were built from scratch and trained with the Adam optimizer using just the NIH training set at a learning rate of 0.001. The top classification models' localization performance was assessed using DICE's continuous version. The model's saliency maps yield clearer findings as the resolution rises, and the selection of r0 significantly affects the location of anomalies. However, certain model outputs may be incorrectly labeled as false positives since the bounding boxes employed to construct anomalies may overstate true ROIs. In summary, [5] the suggested method overcomes the difficulties of multi-resolution with MIL and improves the accuracy of medical picture detection by producing high-resolution saliency maps. The recommended method is a promising method for diagnosing medical images because it produces cutting-edge outcomes on the majority of the 14 anomalies in the NIH's CXR14 dataset.

In the study [6], CheXNet outperformed experienced radiologists in identifying pneumonia from chest X-rays. For all 14 illnesses, CheXNet produced cutting-edge findings, including a heatmap that shows the areas of the picture that are most indicative of pneumonia and a probability score. Images were standardized, scaled down to 224x224, then randomly horizontally flipped for data augmentation during training. During training, the network utilized an Adam optimization algorithm in combination with a weighted binary entropy loss function. With a single-output layer and sigmoid nonlinearity, ImageNet's pre-trained weights have replaced the final fully connected layer. After a single epoch, the validation loss attained its utmost value, thereby prompting a reduction in the learning rate by a factor of ten. The F1 scores were computed by the authors to evaluate and compare the performance of radiologists and CheXNet in detecting pneumonia from chest X-ray images. They found that CheXNet's F1 score was significantly higher than the average F1 score of the radiologists. Their detection rate for Pneumonia, Nodule, Infiltration, and Cardiomegaly is gradually 0.77, 0.78, 0.73, and 0.92. CheXNet was also expanded to classify various thoracic disorders by producing a vector of binary labels for each of the 14 disease classes. To increase the overall amount of unweighted binary cross-entropy losses, the authors modified the loss function. In all 14 pathology classes, including Mass, Nodule, Pneumonia, and Emphysema, CheXNet attained cutting-edge results. To analyze the network's predictions, class activation mappings (CAMs) were used. CAMs produced heat maps that highlight the areas of the image most suggestive of the illness and identify the crucial elements used by the model in its prediction. The authors provide examples of CAMs for pneumonia detection and the 14-class pathology classification tests. The authors underline the potential benefits of CheXNet's automated disease diagnosis from chest X-rays for clinical settings and populations without access to diagnostic imaging specialists.

This paper [7], discusses the two primary parts of the suggested design of an image model and a fully convolutional recognition network. The Pre-Activation ResNet form of the ResNet (Residual Neural Network) architecture serves as the foundation for the picture model. The network receives the input picture and generates a feature tensor that transforms it into a collection of feature maps that have been abstracted. If K is the number of different diseases, the model will predict a binary class probability for each patch in that number. These feature maps are then divided into a grid of patches. A multi-convolutional recognition network receives feature maps after batch normalization and ReLU activation. Since the two goals are framed into the same underlying prediction model, the model is jointly optimized for both illness detection and localization during training. End-to-end learning is made possible by this collaborative training technique, which enables the two activities to be mutually beneficial. The multi-label binary cross-entropy loss was chosen as a training method when training the model. The binary classifier categorizes every disease separately for each patch. Diagnostic and therapeutic accuracy in medical imaging is dependent on the availability of large-scale datasets annotated with detailed disease localization information. Analyzing chest X-rays is especially difficult because many abnormalities have similar characteristics. A recent study suggests using a few X-ray images to improve disease localization and identification. A convolutional neural network acquires image data and encodes disease class and location in the method under consideration. In particular, the proposed model increases AUC values for tiny objects for the majority of diseases. According to the qualitative findings, there is a good correlation between the radiologist's comments and the places where illnesses were found, which may lead to further interpretation and insights into the disorders. According to the research, bounding box supervision enhances classification abilities while lowering the need for training photos. As a result, the proposed unified model performs better for the majority of illnesses, particularly for tiny objects, and simultaneously models disease localization and identification using sparse localization annotation data. This method has the potential to increase medical image analysis accuracy while decreasing the need for substantial annotation data.

In the paper [8], the authors examined two deep-learning approaches for diagnosing cardiomegaly, a condition that results in an enlarged heart. The first method employed anatomical segmentation, while the second relied on image-level classification. A total of 778 chest radiographs were used to train the segmentation model, whereas 65,000 x-rays with labels were used to train the classification model. The authors then used a methodical hyperparameter search to optimize a number of parameters relevant to architecture, learning, and regularization. The evaluation of performance was done using a variety of parameters. The findings demonstrated; the segmentation-based model outperformed the classification-based models. The scientists came to the conclusion that the segmentation-based model performed better while requiring 100 times fewer annotated chest radiographs. A more accurate interpretation of the results was made possible by the segmentation-based approach's provision of a thorough depiction of the regions of interest. This might help researchers and medical experts better understand the illness and how it affects the heart by revealing important new information. The work, according to the scientists, shows the advantages of utilizing segmentation-based deep learning models for medical imaging and raises the possibility that this method may also be effective for the early detection and diagnosis of other disorders. In addition, applying deep learning to medical imaging can significantly enhance the diagnostic procedure's speed and accuracy, thereby improving patient outcomes. In conclusion, the study showed how deep learning might enhance medical image processing, and the authors called for more investigation in this area.

In the paper, [9], A method that is based on deep learning for detecting cardiomegaly, is described. The program, known as CardioXNet, employs the cardiothoracic ratio and U-NET, two deep-learning techniques, to identify cardiomegaly. Using annotated data, U-NET learns to segment images, and OpenCV is used to fix any little mistakes in the region of interest. The existence of cardiomegaly is then determined by computing the cardiothoracic ratio from the U-NET segmentation. Additionally, a Dense-Net neural network is used as a standard of reference. The study's findings show how well deep learning and medical criteria work together to correctly identify heart problems in medical photos. Cardiomegaly was identified using the implementation of CardioXNet. CardioXNet's correctness is reliant on the precision of the hand labels applied during training. CardioXNet's effectiveness might be enhanced if specialists in radiology could provide more precise manual labels. CardioXNet displayed great segmentation accuracy in detecting the heart and chest despite having a small patient dataset, and when the cardiothoracic ratio was combined as a diagnostic metric, it achieved a 94.34% F1Score [9]. CardioXNet, an artificial diagnosis system for cardiomegaly, has demonstrated a high degree of consistency and accuracy with clinical diagnoses. This could eventually result in the manual screen drawing measurements being replaced, which would save radiologists a ton of time.

In the paper [10] titled, the author discusses that through the use of public and private hospital datasets for training, two segmentation models based on the U-Net architecture were created. DICE scores [10] between 0.989 and 0.983 on average, these models performed well and showed the capacity to generalize to different data distributions. The cardiothoracic ratio, which can serve as an indicator of enlarged hearts and probable chest diseases, was able to be automatically calculated by the models using frontal radiographs. The algorithm was put to the test by a group of radiologists in a real-world hospital scenario, and it was discovered to shorten the time they spent examining the heart and boost their F1 score for spotting cardiomegaly. The model's capacity for generalization was enhanced by the implementation of localized energy-based normalization. They supported the results of earlier investigations in their study, which they carried out with radiologists' assistance. According to their findings, utilizing an algorithm to automatically calculate the cardiothoracic ratio shortens diagnostic times and raises the F1 score for cardiomegaly detection. Radiologists can see the results clearly thanks to the visualizations, which also make it simple for untrained people to spot poorly segmented radiographs. To make sure that radiographs from X-ray machines outside of their training distribution are appropriately segmented, they assessed the generalization of their model.

In the 2020 paper [11], the authors make use of dilated convolutions and residual connections to identify pediatric pneumonia. Dilated convolutions reduce over-fitting and degradation by preventing feature space information loss from the model depth and residual connections. To further address the issues of scarce training data and structured noise, they use transfer learning to seed the model's [11] initial parameters. The strategy attempts to increase the precision and generalizability of deep learning-based pediatric pneumonia identification. Overall, the authors describe a unique method for identifying and diagnosing pediatric pneumonia that takes into account a number of issues with medical image processing. In order to detect pneumonia in children, the authors tested their technique for locating textural features and pertinent regions in the images of X-rays. The used dataset displayed a remarkable recall rate for the classification of pediatric pneumonia. Their method [11] shows great promise for precisely identifying pediatric pneumonia in X-ray pictures.

In the paper [12], the study compares the performance of the authors' suggested CNN model and six other models in the context of applying transfer learning and CNN to distinguish between normal, COVID-19, and pneumonia chest X-rays. Two types of evaluations are used in the study: one for the pre-trained models and one for the CNN model. Normal, COVID-19, and pneumonia cases are diagnosed using the first classification, known as PCN, whereas bacterial and viral pneumonia cases are diagnosed using the second classification, known as BV. During the testing phase, the CNN model for PCN classification produced acceptable results with an average accuracy of 91.2%, despite having the least performance. DenseNet 201 achieved the highest average accuracy, followed closely by VGG 19. The remaining models performed similarly. The CNN model [12] performed best in the BV classification example, with an average accuracy of 91.9% compared to all other models' 75%. When compared to the other models, DenseNet 201's average accuracy of 80% was the highest.

The paper [13] shows that by putting out a Deep-Learning System (DLS), this research seeks to address the problem of accurately diagnosing lung problems using chest X-ray pictures. The method entails examining chest radiographs that have been conventionally processed as well as those that have undergone a threshold filter. To evaluate the efficacy of the DLS, a number of standard models were tested experimentally, and SoftMax was used when combined with other networks like AlexNet, and ResNet50. Compared to other models, the results revealed that VGG19 had the highest classification accuracy (86.97%). The study suggested a customized VGG19 network [13] employing the Ensemble Feature Scheme (EFS) to increase accuracy even further. The best accuracy (95.70%), according to the results, was supplied by the VGG19 with RF classifier [13].

2.2 Background Study

Many different diseases can manifest in the thorax and cause problems for the lungs and heart. Pneumonia, cardiomegaly, masses, nodules, and infiltrations are examples of frequently occurring diseases. Cigarette smoking, airborne toxins, heredity, and advancing age are all contributors. Diagnostic tests commonly use imaging methods like X-ray, CT, and MRI of the chest to check on the state of the organs there. The key to successful treatment is early diagnosis. The automated detection and classification of thorax diseases in medical images have been significantly aided by machine learning and deep learning techniques. Thorough knowledge of these conditions, their risk factors, and diagnostic techniques are required in order to accurately detect, diagnose, and treat diseases of the thorax.

2.2.1 Common Thorax Diseases

a) Cardiomegaly

Expanding of the heart muscle is medically referred to as cardiomegaly. A disorder known as cardiomegaly causes the heart to grow to abnormal proportions. This disease can result in a reduction in the function of the heart. In the event that cardiomegaly disease is not addressed, it can lead to a number of issues in the heart. It is very difficult to determine the actual global statistics of cardiomegaly disease due to the fact that the majority of patients with cardiomegaly disease stay undiscovered. Yet heart failure is the leading killer worldwide, responsible for the deaths of 8 million people every year. Cardiomegaly is frequently linked to heart failure as an underlying cause.

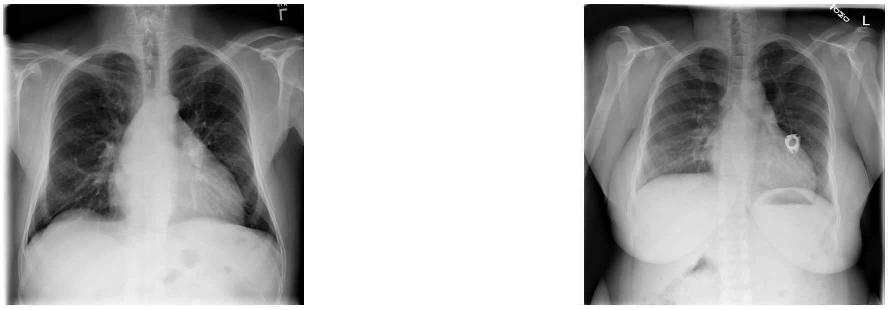


Figure 2.1: Cardiomegaly X-ray Images

If cardiomegaly disorders are present, early detection is essential for effectively treating them. Methods of deep learning have the potential to be useful in the diagnosis of cardiomegaly illnesses. If the methods of deep learning can be employed in cardiomegaly illness detections, then this might potentially assist save lives as well as resources. Fig 2.1 shows two x-ray images of cardiomegaly disease.

b) Infiltration

Infiltration is the unintended entry of fluid or medication into the surrounding tissues during intravenous (IV) therapy or other medical procedures. Utilizing a needle or catheter, IV treatment includes injecting fluids, medicines, or other substances directly into a patient's veins. When the IV needle or catheter is not correctly placed into the vein or when the vein or IV site is damaged, infiltration may happen. The term "infiltration" describes the unintentional flow of fluid or medicine into the surrounding tissue from the intended circulatory channel. The fluid or medication seeps into the adjacent tissue rather than entering the circulation immediately, perhaps leading to difficulties.

When infiltration occurs, the impacted region may show obvious symptoms. Since the infiltrated fluid accumulates in the tissue, swelling is a frequent symptom. The fluid's presence and the impeded blood flow may make the area that was invaded feel cooler to the touch. Additionally, a less blood supply may cause the region to seem darker than the tissue around it. Patients who have infiltration could feel pain or discomfort there. The pain this causes can range from mild discomfort to severe pain.

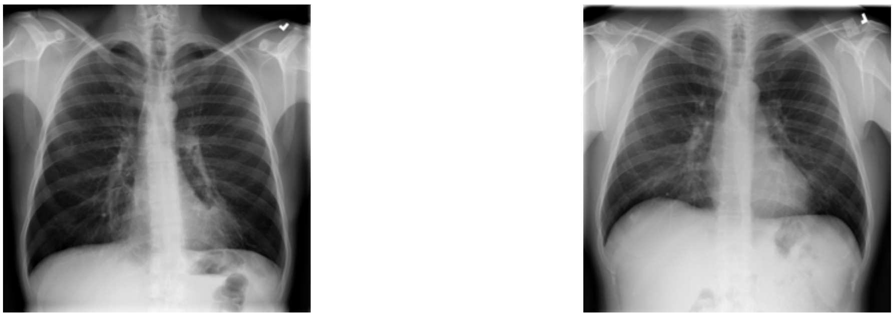


Figure 2.2: Infiltration X-ray Images

Timely diagnosis and effective management of infiltration are essential for minimizing patient harm. Healthcare providers need to keep a close eye on the IV site while administering fluids or drugs. As the patient may have mentioned swelling, coldness, pallor, or pain in the area, it is important to look for these and other signs of infiltration. Immediate action should be taken if infiltration is suspected or proven. To stop additional infiltration, the IV should be stopped, and other ways to provide medicine should be looked into. Affected limbs might be elevated to lessen edema and enhance fluid outflow as a way to lessen the consequences of infiltration. Compresses can be used on the infiltrated region to enhance tissue healing and ease pain. Fig 2.2 shows two x-ray images of Infiltration disease.

c) Nodule

Nodule is a term used to describe an odd growth or bump that may appear anywhere on the body. Physical examination or several image detection techniques may be used to find it. These are typically solid growths with a diameter of less than 1 cm. These growths may be cancerous or benign. But there can be other explanations as well. There can be different kinds of nodules, such as thyroid nodules, lung nodules, skin nodules, vocal cord nodules, etc. Nodules can occur for a variety of causes, like infections, inflammation, and abnormal cell growth.

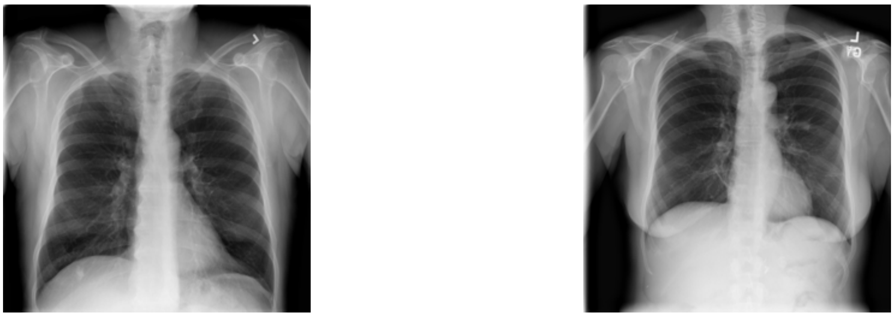


Figure 2.3: Nodule X-ray Images

The detection of lung nodules may be greatly aided by chest X-rays or CT scans. Nodules of this sort often develop on lung tissue. Normally, skin nodules are visible on the skin's surface and may cause severe conditions like skin cancer or acne. The Nodule's condition as benign or malignant can affect the strategy of diagnosis and treatment. Depending on the size, location, and symptoms of the Nodule, it can be measured whether that Nodule is benign or malignant. Nodules larger than 1.2 inches in diameter may have a higher cancer risk. Besides, it is also required for effective treatment according to the condition of the Nodule. Sometimes a biopsy may be required to examine a small tissue as a sample and identify the type of lesion. If a Nodule is assumed to be a cancerous or a malignant one, then some measures can be taken, such as surgery, radiation treatment, chemotherapy, or a mix of these methods. Fig 2.3 shows two x-ray images of Nodule disease.

d) Pneumonia

Pneumonia is a type of infection that affects one or both of the lungs. Numerous microorganisms, including bacteria, viruses, fungi, and others, may be to blame. Inflammation and fluid in the lung can be brought on by a lung infection or pneumonia. Coughing, fever, chest pain, and breathing difficulties are just a few of the symptoms of this pneumonia. Even though this illness can strike anyone at any age, it primarily affects young children and the elderly. In other words, it has an impact on those who have weakened immune systems. Pneumonia can affect a person's body for a variety of reasons, including smoking, chronic lung diseases, heart problems, etc.

Depending on the source of the underlying infection and the severity of the sickness, pneumonia symptoms might differ. Healthcare professionals diagnose pneumonia through physical exams, symptom assessments, chest X-rays, blood tests, and sputum cultures. A chest X-ray is routinely carried out to confirm the presence of pneumonia and evaluate the level and location of lung involvement. It may be used to find swelling, consolidation (a big group of sick tissue), or fluid accumulation in the lungs. When the severity of the symptoms and the patient's general health are taken into consideration, it is possible that additional tests will be required. These could include pulse oximetry, bronchoscopy, or computed tomography (CT) scans, which use flexible tubes to see within the airways. Treatment options for pneumonia may include antibiotics, antiviral medications, and supportive measures, including rest and fluid intake, depending on the original cause. Fig 2.4 shows two x-ray images of pneumonia disease.

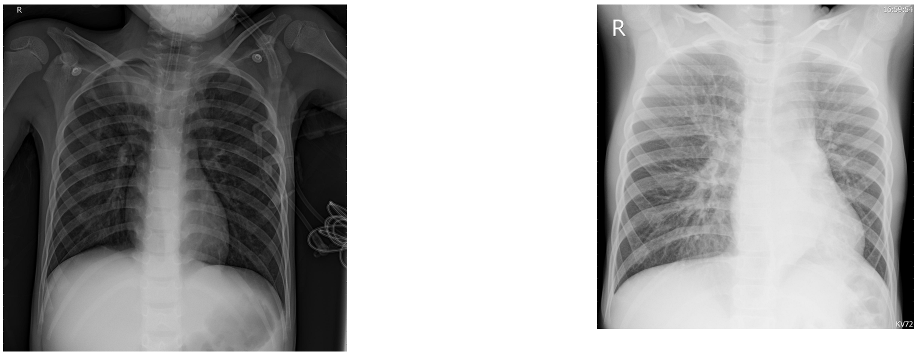


Figure 2.4: Pneumonia X-ray Images

2.2.2 Chest X-ray

Electromagnetic waves make up one category of energy, which also includes X-rays. Through their application in medical imaging, they contribute to the generation of visual representations of the interior of the body. The images that were produced have a variety of grayscale tones that correlate to the differing degrees of radiation absorption that were achieved by the various tissues of the body. Frontal and lateral views are the most common types of image depictions.

The photons that makeup X-rays are created when electrons in an X-ray tube are accelerated. These photons condense into a beam that is divergent, which then travels via a collimator and is aimed at the body of the patient. To differing degrees, X-rays are absorbed by the various types of tissues. The X-rays that are being sent are detected by a detector system, which then turns them into electrical impulses. A computer receives these signals, processes them to create a digital image of tissue absorption, and shows it. Images can be improved with the help of image processing algorithms, which can also be used to spot anomalies. When completion of processing of an X-ray image, a radiologist is able to utilize it for the purpose of rendering diagnoses and providing recommendations for treatment.

For the analysis of medical images, the application of machine learning algorithms is gaining popularity, and chest X-rays constitute an important dataset that may be utilized for the purpose of training such models. The use of X-ray images can be done to teach a machine learning algorithm how to recognize specific patterns and features in the image, such as the presence of nodules or pneumonia. This is accomplished by giving the system a generous amount of chest X-ray images. As a consequence of this, these models are able to lend a hand in the process of disease diagnosis and aid in the identification of anomalies in X-ray pictures.

Chapter 3

Methodology

3.1 Top Level Overview of the Proposed Method

To demonstrate our proposed model, at first, we have to collect the data containing Chest X-rays from which we will work to detect the abnormalities. Next, we'll need to pre-process the X-ray images so the model can more clearly distinguish between subtle differences. The next step is to divide the data into three equal parts, one each for training, validation, and testing, using a 70:20:10 split. The next step is to instruct the CNN models with the aforementioned training and validation data. Features will be extracted from the trained models by removing the lowest layers till the global average pooling 2D layers. The next step is to use the combined features extracted by the CNN models for both the training and validation datasets to train the Machine Learning classifiers. To get the final accuracy for the automatic detection of thorax diseases, we will use the testing dataset in the ML classifier to predict by comparing the features obtained from the CNN models previously. Fig 3.1 provides a high-level outline of the model we propose.

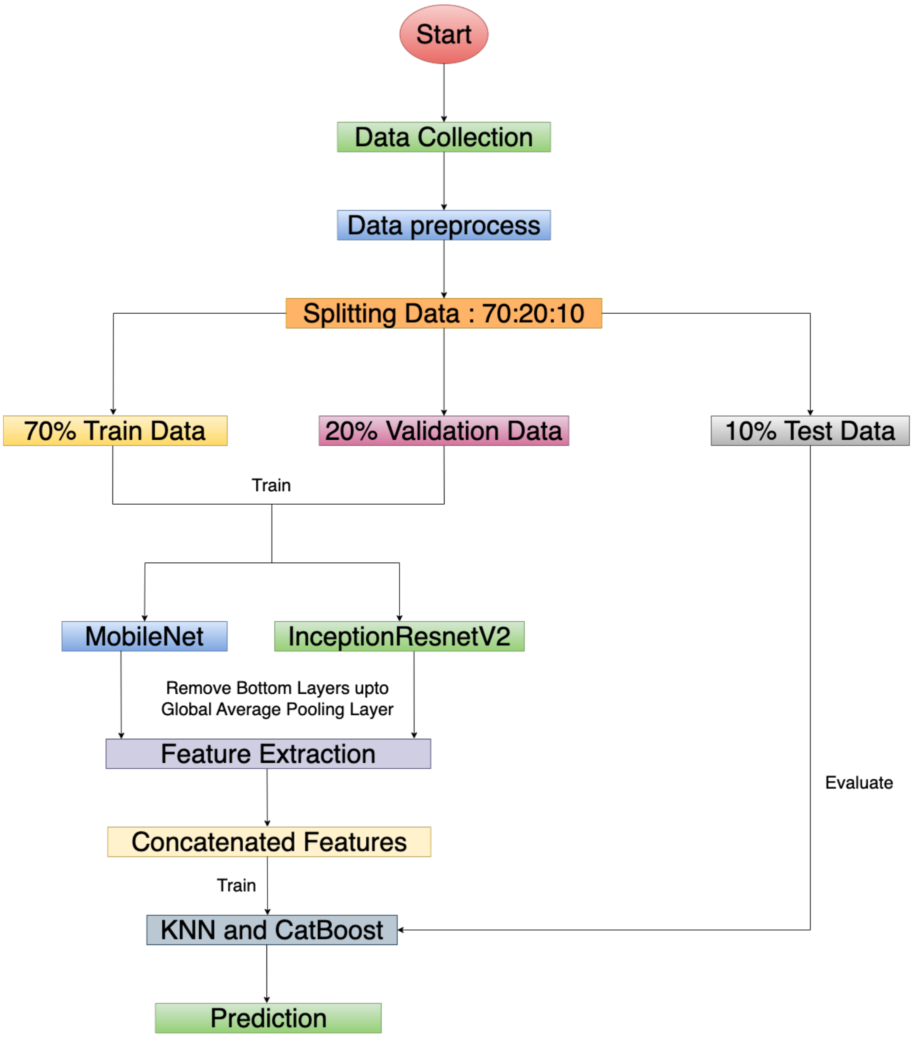


Figure 3.1: Top Level Overview of the Proposed Method

3.2 Dataset

3.2.1 Dataset Acquisition

For our work, we mainly used two data sets. Both of them are publicly available on the Kaggle website. The datasets "ChestX-ray8" [14] and "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification" [15] contain X-rays for 14 diseases and pneumonia, respectively.

The dataset named "ChestX-ray8" was compiled using clinical Picture Archiving and Communication System (PACS) data obtained from the National Institutes of Health Clinical Center [14]. Sixty percent of all frontal X-rays taken at the time of data collection are included in this set. As a direct consequence of this change, the entire dataset possesses a significantly increased level of both diversity and realism. A few examples of clinical and diagnostic difficulties drawn from real-world settings are included in this collection. In this dataset, we found poor data on pneumonia. So, we had to use another dataset containing the x-ray of pneumonia.

The second dataset [15] comprised anterior and posterior chest radiographs of children aged one to five years, which were sourced from the Guangzhou Women and Children's Medical Center. The X-ray imaging was conducted on youth people as a component of their routine preventive healthcare evaluations. To ensure the accuracy of the X-ray image analysis, each and every radiograph was thoroughly examined at the outset in order to identify and exclude any scans that were of poor quality or difficult to interpret. Two experienced medical professionals reviewed the images and rated them in order to determine whether or not the diagnoses included within them were suitable for use in training the AI system. The set used for evaluation was additionally reviewed by another expert in order to address and remedy any potential grading errors.

3.2.2 Exploratory Data Analysis

The dataset known as ChestX-ray8 comprises around 112,120 X-ray images obtained from a total of 30,805 patients. A number of text labels that describe various thoracic disorders or diseases are connected to each image. This group comprises widespread thoracic ailments such as atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, and pneumothorax. The X-rays have a resolution of 1024x1024. The 12 zipped files add up to between 2 and 4 GB in total. Nearly 10,000 .png images make up the majority of them. In all, 14 diseases are included in the dataset.

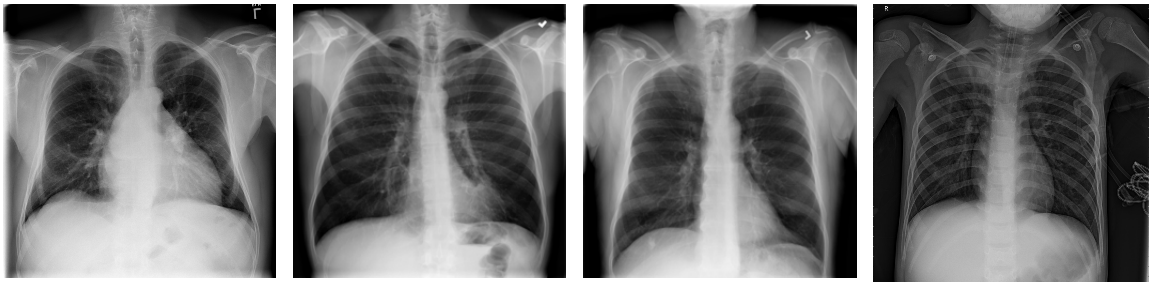


Figure 3.2: Dataset Samples

The second dataset we use is "Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification". There are a total of 5,868 X-ray pictures, all of which are in .jpeg format. Three distinct directories, train, test, and val, contain the entire dataset. And inside each of them are two subfolders called Pneumonia and Normal which each contain a collection of X-ray images showing normal and pneumonia-related cardiac conditions. The X-rays of the chest between the ages of 1 and 5 were gathered and evaluated for quality. The photos were evaluated by two licensed physicians before being used to train an AI system. A third expert checked the exam set to make sure there were no grading mistakes. Figure 3.2 shows 4 X-ray images of specific 4 diseases from our dataset.

3.2.3 Data pre-processing

(i) Extracting the Specific Diseases Images

In our first dataset, there are x-ray images of 14 diseases, and in the second dataset, we have one disease. As we are working with a total of 4 diseases, of which 3 diseases are from the first dataset, we needed to make the 3 diseases' x-ray images filtered from the 14 diseases. To extract those 3 diseases' x-ray images from the 14 diseases, we wrote a script and took our specific 3 diseases' x-ray images. Then we achieved our specific x-ray images, and by adding the second dataset's one disease, we made our expected 4 diseases x-ray images dataset and resized every image in 224x224. Table 3.1 and Fig 3.3 provide the numbers and a bar chart, respectively, for better understanding.

Name of the Disease	Number of X-ray Images
Cardiomegaly	2776
Infiltration	3200
Nodule	2703
Pneumonia	4254

Table 3.1: Dataset X-ray Images Count for Each Disease

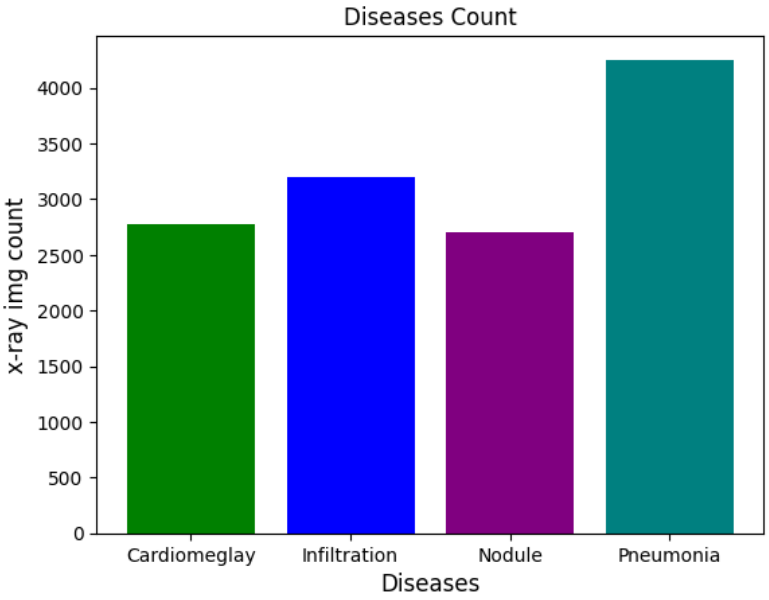


Figure 3.3: Dataset X-ray Image Count Visualization

(ii) Applying CLAHE Algorithm for Contrast Enhancement

One Algorithm of image processing, which is commonly used in the medical sector and the sectors where the quality of the image is a priority, is known as the CLAHE algorithm. CLAHE stands for ”Contrast Limited Adaptive Histogram Equalization”. This algorithm is much better than the traditional histogram equalization. This has the ability to limit the amount of contrast enhancement which helps preserve the detail without introducing any new noise to the image.

This algorithm is used by us because we are working with X-Ray images. The X-ray images of cardiomegaly disease had very low contrast, which can make it difficult to accurately interpret and diagnose certain conditions. So, we applied the CLAHE algorithm to the X-ray images of cardiomegaly. We show the difference between before and after applying the CLAHE algorithm in the cardiomegaly x-ray in Fig 3.4.

CLAHE equalizes the image histogram locally, which means that it enhances the contrast in areas of the Image that need it the most while preventing over-enhancement and loss of detail in bright or dark regions. This results in improved visualization and diagnosis. This whole process can retain or enhance the visibility of features in low-contrast images.

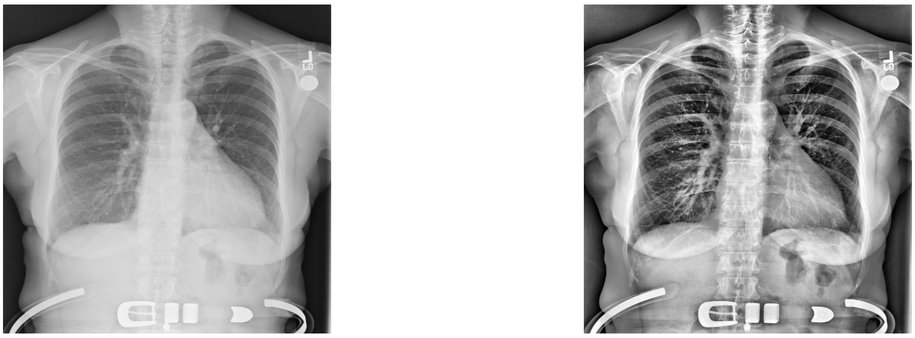


Figure 3.4: Before and After Applying CLAHE Algorithm

(iii) Data Splitting

We have split our dataset into train, test, and validation. The ratio of train:test:val is 70:20:10. Table 3.2 is given below to see the count of x-ray images in each class after splitting in train:test: validation.

Diseases Name	Train Sample	Val Sample	Test Sample
Cardiomegaly	1943	554	279
Infiltration	2240	640	320
Nodule	1893	540	271
Pneumonia	2980	851	423

Table 3.2: X-ray Images Sample Count After Splitting for Each Disease

(iv) Data Augmentation

We needed to expand the training dataset because the amount of X-ray image data we had was extremely limited. Thus, we expanded the training set using the Image Data generator's flip and fill modes.

3.3 Model Specification

3.3.1 CNN Transfer Learning Models

The Convolutional Neural Network (CNN) is an advanced deep learning architecture that has been specifically developed to independently classify and forecast novel data categories through the assimilation of comparable training data. In the 1980s to 1990s, researchers started to explore the idea of visual pattern recognition. From that idea, they successfully implemented it, and we achieved immense success in the autonomous detection and classification of images, object detection, and many more. A paper from 2012 introduced a new CNN model named AlexNet [16] was a great invention in the world of autonomous classification.

CNN analyzes and processes the given input data with a grid structure. It learns from the given data directly and finds the patterns of images, and makes the prediction or classifies them into categories. The layers of convolutional, pooling, fully connected, activation, dropout, and batch normalization make up classic neural networks. The layer with complete connectivity precedes the output layer.

$$W2 = (\frac{(W1 - F + 2P)}{S} + 1) \tag{3.1}$$

$$H2 = (\frac{(H1 - F + 2P)}{S} + 1) \tag{3.2}$$

So, at first, the network takes the images in M x M or N x N size in the input layer. Normally the recent CNN model prefers the 224 x 224 size of the input image. Then the data is passed through a series of kernels or filters in the convolutional layers, which do most of the computation. Commonly used kernel sizes in image processing include 1 x 1, 3 x 3, 5 x 5, and 7 x 7. The process involves sliding filters or kernels over the input images and performing a dot product between the resulting features and the filter component. Now a features map is created to give in the next layer. Assuming an input size of W1 x H1 x D1, the resulting output of the convolution layer can be expressed as W2 x H2 x D2.

$$D2 = k \tag{3.3}$$

The equation (3.1) utilizes symbols to represent certain variables, where F denotes the spatial extents, P represents the zero-padding, and S denotes the stride. The term "stride" relates to the magnitude of the step taken by the convolutional kernel or filters as they traverse the feature map. For example, if the stride value is 1, it means that at a time, the kernel/filter will move by 1 pixel. Normally the smaller kernel size is good for catching the spatial details of the input. Equation (3.2) can be used to determine the output height, H2, of a convolutional layer in a neural network. In equation (3.3), k represents the count of filters or kernels used in the convolution layers. After doing the computational work and making the features map, the convolution layers pass the features map to the pooling layers.

There are two pooling layers: maximum and average. Pooling layers compress features from the features map, reducing the computational cost of generating the conv features map. Therefore, the pooling layer's output size is (W3 x H3 x D3):

$$W3 = W2/2 \tag{3.4}$$

$$H3 = H2/2 \tag{3.5}$$

Equations (3.4), (3.5), and (3.6) calculate the width W3, height H3, and depth D3 of a feature map or tensor in a neural network layer. These equations compare the current layer's width W3, height H3, and depth D3 to its predecessor's W2, H2, and D2.

$$D3 = D2 \tag{3.6}$$

The max pooling layer takes the features map output by the preceding convolutional layers and selects the largest element for further processing. However, in the average pooling layer, it takes each feature map element and averages it out.

Then finally, there are some fully connected layers. These layers are also known as dense layers. The neurons from the preceding pooling layers are used in these layers, but they are dependent on weights and biases. All features from the previous layers start to be flattened in these layers, and then the flattened vector goes through multiple fully connected layers when the mathematical functions operations take place. Here starts the process of making predictions in the new set of images. This layer is just before the output layer.

In between the layers and also on the inner side, there are some functions and methods used. Some of them are dropout, activation functions, and batch normalization. To overcome the overfitting problem, we use the dropout method. It actually drops some features from the features map to force the model to learn more robustly. Another method is the activation function. The activation function decides which features are important and need to be activated and which features are not important and don't need to be activated. It adds non-linearity to the network. Some activation functions are ReLU, Softmax, Tanh, and Sigmoid functions. Sigmoid and Softmax functions are commonly employed in the context of binary classification. The preferred activation function for multi-class classification is Softmax.

The ultimate approach refers to the supervised learning technique of batch normalization. While the training process goes on, it normalizes the data within each mini-batch by calculating the mean and standard deviation. The standard deviation is divided by the mean and then subtracted from the input data to create a normal distribution. The model determines the mean by averaging the channels of the feature maps. Equation (3.7) shows the formula for calculating the mean. The formula of variance to find the standard deviation is also shown in the equation (3.8)

$$\text{Mean} = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{3.7}$$

$$\text{Variance} = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \text{Mean})^2 \tag{3.8}$$

Fig 3.5 shows the functional basic process of a CNN, which we have explained [17].

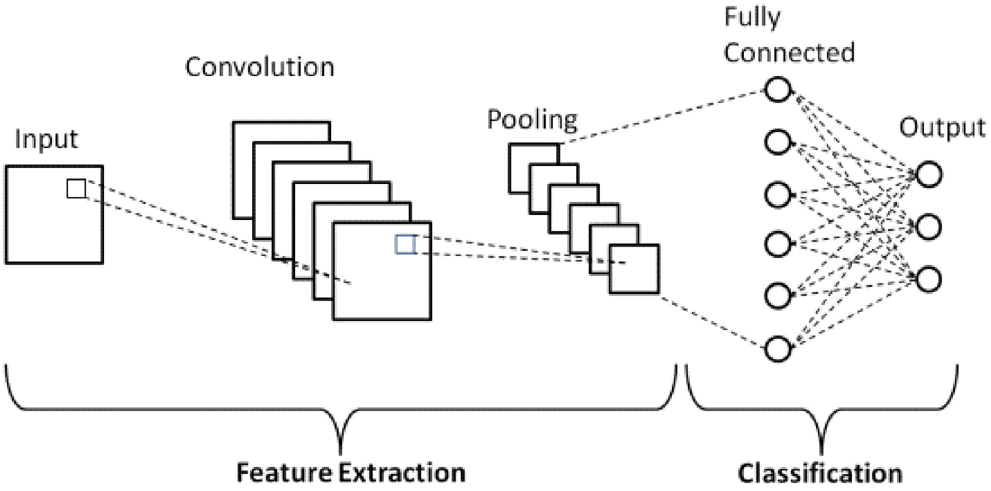


Figure 3.5: Schema of Basic CNN Architecture

(i) Inception-Resnet-v2

An effective technique for picture categorization is Inception-Res-Net-v2. This convolutional neural network has 164 layers. It has shown to be very successful for picture categorization. It is also a relatively efficient model, which means that it can be used to classify images. The proposal was made by Christian Szegedy et al. in their 2016 work ”Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” [18].

The Inception module is the core building block of the architecture and is responsible for capturing multi-scale features. It has a max pooling branch in addition to three other convolutional branches with varying filter sizes (1x1, 3x3, and 5x5) in parallel. The design of Inception-ResNet-v2 incorporates residual connections, which were initially presented in the ResNet model. The vanishing gradient problem is lessened, and deeper networks may be trained with the help of residual connections, which let the model learn from both the initial input and the intermediate feature maps. On the other hand, by utilizing filters of various sizes inside the same layer, Inception modules are made to gather information on various scales. The network may learn rich representations by processing local and global input effectively. The advantages of the Inception architecture, which records multi-scale characteristics, and the ResNet architecture, which permits the training of extremely deep networks, are combined in Inception-ResNet-v2.

When there isn't much data available to train a model on the given problem, this method is quite useful. Applying ImageNet dataset weights customizes Inception-ResNet-v2 as a pre-trained model. As a result, the network may learn the characteristics that are unique to the new job while also utilizing the features that were discovered when analyzing the huge dataset. Object detection, Scene recognition, Face recognition, Image segmentation, and other image classification tasks are just some of the many uses for Inception-ResNet-v2. The Inception and Residual connection architectures have been combined to form Inception-ResNet-v2.

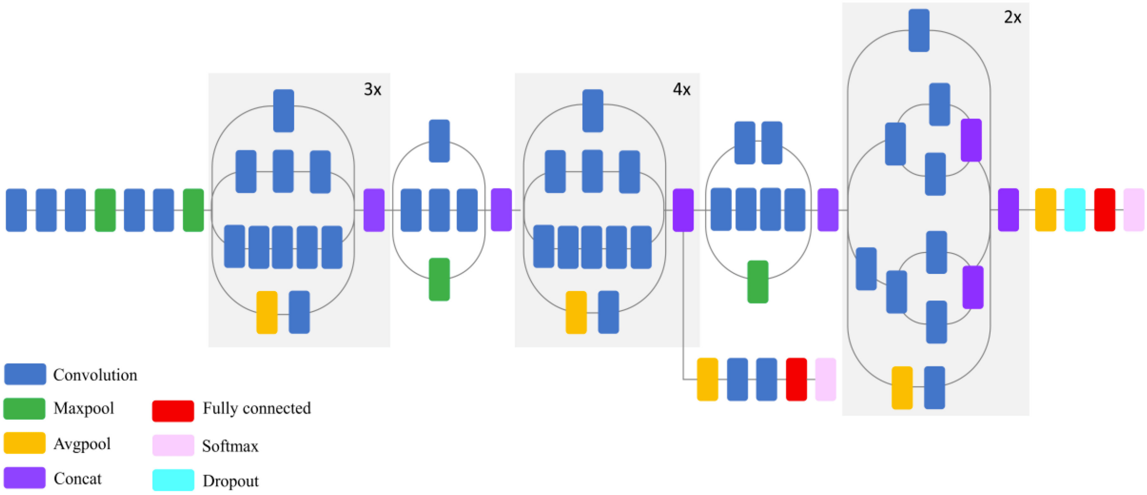


Figure 3.6: Schema of Inception-ResNet-v2 Architecture Baseline

Fig.3.6 shows this network's baseline architecture [19], while Fig.3.7 shows its Inception-ResNet-v2 layers [18]. First, this network takes the input images in 299x299 size. The initial phase of this network is referred to as Stem, which employs a sequence of traditional convolutional layers, pooling layers (such as max pool and avg pool), and normalization techniques to extract basic features from images. Subsequently, the network comprises three Inception-Resnet blocks, which serve as the fundamental components of the architecture. In each Inception-Resnet block, there are inception modules that have different sizes of filters - 1x1, 3x3, and 5x5. By using these different sizes of filters, it captures the multi-scale features from the input images. Between each Inception-ResNet block, there are Reduction Blocks which reduce the spatial dimensions of the features map, which is given from the previous Inception-ResNet block. Its combination of convolutional and pooling layers helps to reduce computational complexity. In the end, the ultimate prediction process is carried out in layers that pool global averages and layers that are fully connected. The final output layers are where this network's structure comes to an end.

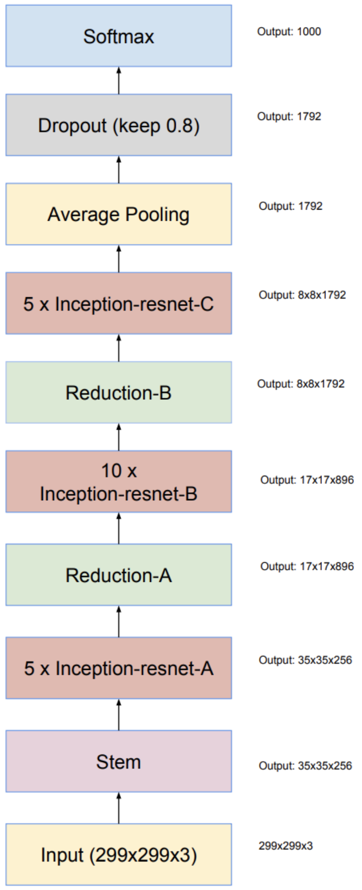


Figure 3.7: Schema of Inception-ResNet-v2 Architecture Layers

(ii) MobileNet

A convolutional neural network architecture called MobileNet is compact and optimized for computation with constrained computing resources. Given that the model is compact and lightweight, it may be used for a number of picture classification applications to provide results with high accuracy. This idea was put out by Andrew G. Howard et al. in their 2017 publication, ”MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” [20].

MobileNet’s primary goal is to find a middle ground between computational efficiency and model accuracy, making it open for use on devices with limited resources. MobileNet reduces the computational cost of typical CNNs while retaining sufficient accuracy on a variety of computer vision applications by combining depth-wise separable convolutions with point-wise convolutions.

Convolutions that are depthwise separable are utilized by MobileNet in order to reduce the number of model parameters. The utilization of depthwise and point-wise convolutions involves a bifurcation of the convolution process into two distinct stages. Point-wise convolutions integrate the results of depth-wise convolutions across channels by performing a 1x1 convolution, whereas depth-wise separable convolutions apply a different convolutional filter to each input channel. MobileNet is able to achieve high accuracy while utilizing fewer parameters and less memory than other CNN designs because of this method. Several image classification tasks, including object identification, scene recognition, and face recognition, have proven to be successful using MobileNet.

Another parameter is the resolution multiplier (ρ), which scaled down the input image resolution, in MobileNet. It lessens the model's computational and memory requirements. However, it can mean losing precision and fine-grained details. Additionally, MobileNet uses a bottleneck structure to further cut the cost of computing. It builds a bottleneck layer to reduce input channels with 1x1 convolutions before depthwise convolution. This bottleneck layer helps to ensure that low-dimensional representations are captured as accurately as possible.

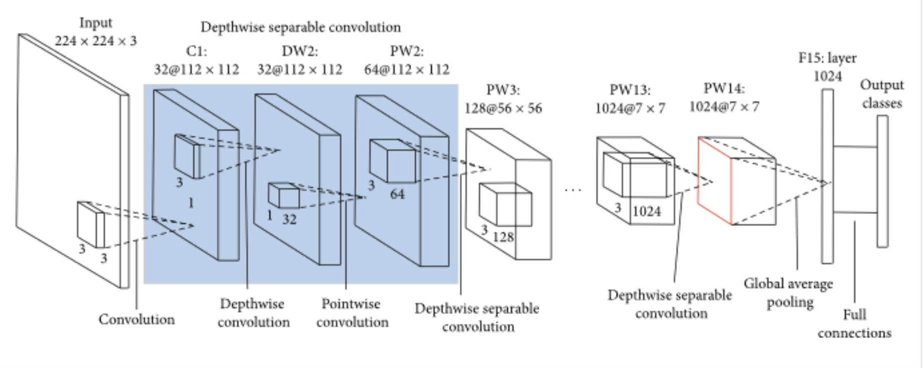


Figure 3.8: Schema of MobileNet Architecture

The MobileNet network takes the input images in a size of 224x224 [21]. After the input layer, there are some traditional convolution layers. After that, the depthwise and pointwise convolution layers come. A depthwise convolution layer is responsible for performing spatial convolutions on each input channel separately by applying the convolution filter and generating a features map. Then the depthwise convolution layer gives that feature map to the pointwise convolution layer, which is also known as the 1x1 conv layer. It expands the number of channels and allows learning more complex feature relationships. After these two layers, the depthwise separable layer comes, which combines the depthwise and pointwise conv layers and achieves a significant reduction in computational cost and model size. It allows MobileNet to be more efficient. Subsequent to the output layer, there exists a sequence of layers that includes the fully connected layer, the global average pooling layer, and another fully connected layer. Figure 3.8 shows this network's baseline schema.

(iii) VGG-16

The efficiency and clarity of VGG16 are well recognized. 16 layers are present in the convolutional neural network (CNN) known as VGG16. Because it is a deep CNN, it can learn to represent pictures in a way that is more sophisticated than shallower CNNs. The ImageNet dataset was used for pre-training, so it could identify a wide variety of objects in pictures. However, as it is an extremely deep CNN, both training and using it may require a significant investment in processing power. The depth of the architecture makes it possible to photograph fine details and patterns. The use of lower filter sizes by VGG16, despite its deeper depth than earlier models like AlexNet, aids in the preservation of spatial information. However, compared to other subsequent models, its depth also makes it computationally pricey and memory-intensive. Compared to certain more recent CNN designs, including ResNet and Inception, VGG16 is less effective. VGG16 is not as effective as the more recent CNN models in the detection of tiny objects in pictures.

The authors of a scholarly article titled ”Very Deep Convolutional Networks for Large-Scale Image Recognition” introduced the VGG-16 model [22]. VGG16 architecture comprises 13 convolutional layers and 3 fully connected layers. The subsequent layers in this convolutional neural network architecture consist of 3x3 kernels and utilize max pooling. VGG16 has gained popularity as a CNN architecture and is used as a standard for assessing the effectiveness of new models. Training on new datasets or specific tasks can begin with the weights of the pre-trained model, allowing for faster convergence and better performance with less training data. As a result, this pre-training enables VGG16 to pick up on and recognize a range of items in photos. A number of image categorization tasks may be performed with VGG16. It may be used, for instance, to categorize pictures of people, automobiles, or animals. The job of detecting and finding things in photographs is called object detection, and VGG16 may be utilized for this purpose as well.

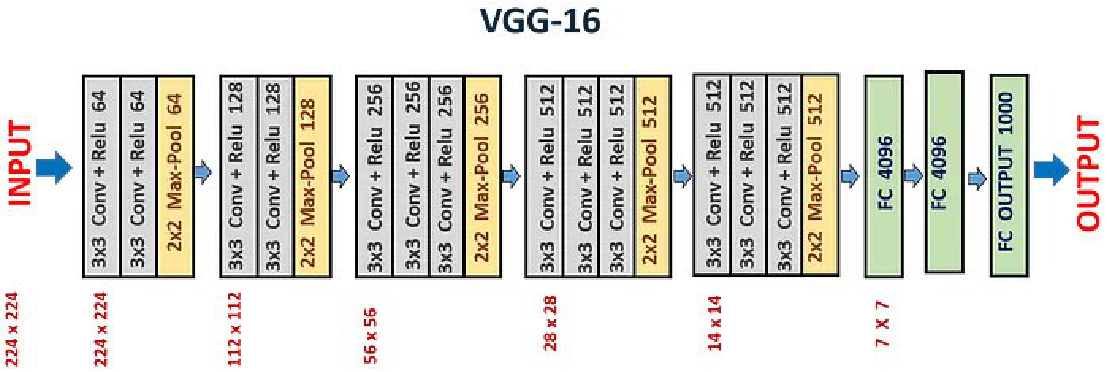


Figure 3.9: Schema of VGG-16 Network

This network was introduced in 201. It takes the input images in 224x224. The model’s non-linearity is introduced by the ReLU activation functions that follow each of the model’s 13 convolutional layers. It helps the model learn complex relations and mapping. A max-pooling layer compresses the features map’s spatial dimensions and extracts its key features after a predetermined number of convolutional layers. At last, there are 3 fully connected layers. Here dropout is also used to prevent overfitting. The VGG-16 network’s schema [23] is depicted in Fig 3.9 3.9.

(iv) DenseNet-121

Keras applications include a model called DenseNet121 [24], which has an accuracy of 75% for the Top-1 score and a score of 92% for the Top-5 score. Having said that, it also has a parameter count of 8.1M. DenseNet121 was proposed in their paper titled ”Densely Connected Convolutional Networks,” published in 2017.

DenseNet stands for Densely Connected Neural Network. DenseNet is known for its densely connected pattern. In this architecture, each layer is connected in a feed-forward fashion. The primary benefit of this architecture is that it facilitates improved information flow.

DenseNet121 is a DenseNet Architecture variant. Densenet121 consists of 121 distinct layers. In DenseNet121, each dense block consists of multiple convolutional layers which have ReLU activation and batch normalization. Each dense block's output is passed to a transition block, which reduces the feature maps' dimensions before forwarding them to the next dense block. Typically, the transition blocks consist of a pooling layer for spatial dimension reduction and a convolutional layer for dimension reduction.

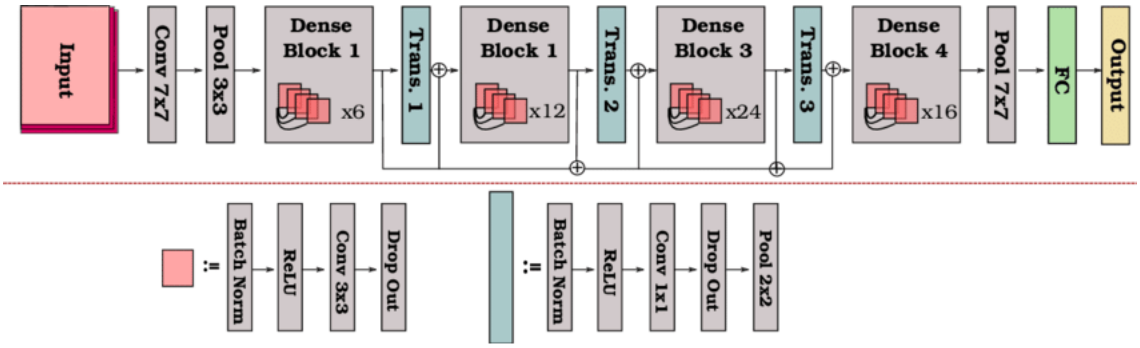


Figure 3.10: Schema of DenseNet-121 Network Architecture

DenseNet121 has shown brilliant performance in image classification and object detection. It has demonstrated high precision across multiple benchmark datasets. In the initial layer of DenseNet-121, there are 7x7 kernels. The output layer is sequenced after the global average pooling, transition, and fully connected layers. Figure 3.10 depicts the schematic representation of the network architecture of DenseNet-121 [25].

3.3.2 Machine Learning Models

(i) KNN

K-nearest neighbors, or KNN, is an acronym. The non-parametric supervised learning algorithm K-nearest neighbors, also known as KNN, can be applied to classification and regression issues. This means that it doesn't make any assumptions about the distribution of the underlying data. Predicting an unknown instance's label requires first finding the k most similar instances to the unknown.

Since KNN doesn't explicitly construct a model during the training phase, it is regarded as a lazy learning algorithm. The feature vectors and accompanying labels of the training samples are stored in memory by KNN during the training phase. When making a prediction for a new, unlabeled instance, the algorithm first calculates the distance between the new instance and each instance in the training set. Distances can be measured using Manhattan and Euclidean distances. The equation of Euclidean distance is (3.9), where (x1, y1) are the origin point's coordinates, (x2, y2) are the destination points, and d is the distance between them. The ”k” closest neighbors are then chosen based on distance. Using KNN for classification, the labels of the new instance's k nearest neighbors are used to determine the new instance's label. It counts the instances of each class among the k neighbors and assigns the class label with the highest count as the label for that class. There are other voting methods that can be used, including majority voting and weighted voting. When doing regression tasks, KNN may forecast a continuous value by averaging or weighting the target values of the k closest neighbors. As a result, using the training data that has been saved, it executes the computation during the prediction phase. KNN is hence adaptable and good at managing dynamic or changing datasets.

$$d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2} \tag{3.9}$$

Although KNN is an easy-to-use technique, it may be costly to compute, especially for big datasets. The user must select the k value in KNN as a hyperparameter. The efficacy of the model may be influenced by the magnitude of k; generally, an increased value of k produces a swifter yet more precise model. KNN is a versatile method that may be used for various problems. It is commonly used for classification difficulties like spam filtering and photo categorization. Regression problems, like predicting the price of a home or the quality of a movie, are also within its purview.

(ii) CatBoost

CatBoost is mainly based on the concept of decision trees. It is a gradient-boosting approach that is particularly successful for issues with mixed data types, including both numerical and categorical variables. It is created primarily to handle categorical features in machine learning applications. This method may be used for classification, regression, and ranking since it is quick, scalable, and accurate.

A powerful predictive model is produced using the ensemble learning approach known as gradient boosting by combining a number of weak learners, often decision trees. The link between characteristics and labels may be modeled using decision trees, which are straightforward yet effective. Gradient boosting is a method that CatBoost utilizes to increase the precision of decision trees. The technique of adding additional trees to a model to fix the errors generated by the earlier trees is known as gradient boosting. CatBoost is unique in that it effectively handles categorical variables without the need for explicit preprocessing. The approach used to handle categorical data internally is called ”Ordered Boosting,” and it combines permutations and gradient-based optimization methods to determine the optimal split points for categorical features during tree construction. Fast algorithms include CatBoost. On huge datasets, it can train models fast. The algorithm CatBoost is also scalable. Large datasets may be utilized to train models with it. An accurate algorithm is CatBoost. It can do a range of machine learning tasks with state-of-the-art outcomes. Data scientists and machine learning experts are increasingly using CatBoost, especially in situations where categorical variables are important. It has been effectively used in a number of fields, including image processing, recommendation systems, advertising, and finance.

3.4 Proposed Fusion Model

The purpose of fusing multiple models is to create a new and improved model that's better performing than the individual models that have been fused to create it. When a model is fused, we usually use multiple different models by extracting features and combining the features obtained from each individual model, and we use the combined features to make predictions. We do so with the hope that the different feature extractors will extract features that overlap as little as possible. To simplify, the goal is to have multiple feature extractors, each pulling features from a unique subset of the input data, then using those features in conjunction with each other to train an ML classifier. The classifier will use the combined features to make more accurate predictions than the individual models. Using combined features makes it likely that the weaknesses of one feature extractor will be compensated for by another feature extractor, and we will achieve a better model as a result.

Our fused model uses inception-Resnet-V2 and MobileNet and combines their features of them by extracting from the training and validation set. These two models are used because, after testing several models, we have identified these two models to be the best performing on our training data. Then we used ML classifiers and trained those by extracting features and then also extracting features from the testing data, and the final prediction will be done by the ML classifier by comparing the features.

3.4.1 Combining the Features from Two CNN Models

In order to combine the features from multiple CNN models, we need to go through the following steps.

(i) Perform training and validation of the Inception-ResNet-V2 and MobileNet models using the respective datasets.

(ii) Extract the features from the Global average pooling 2D Layer from each CNN model. Fig 3.11 shows the process of how we extract the features.

(iii)As shown in Fig 3.11, the two sets of features obtained in this way will be joined to form the combined features of the two CNN models.

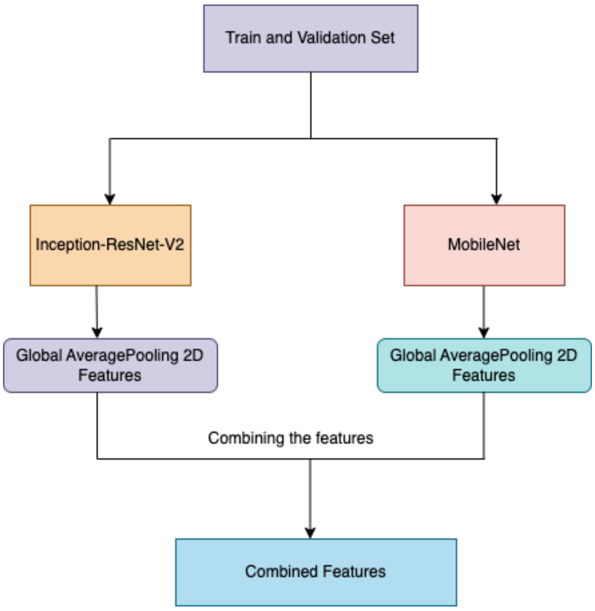


Figure 3.11: Extracting and Combining the features from two CNN models

3.4.2 Using KNN Classifier

The K-Nearest Neighbors (KNN) algorithm is utilized to classify a given data point based on the classes of its neighboring data points. The algorithm assigns the class label that occurs most frequently among the neighboring data points as the class label for the target data point.

For KNN predictions, data points are represented as vectors of the features of the data point each data point in the training and validation data has corresponding labels. The values in these vectors are used to determine the distance of any target data point from any other data point in the training data. Throughout the training stage, the K-Nearest Neighbors (KNN) algorithm retains the training dataset, which comprises the feature vectors and their corresponding labels. The K-Nearest Neighbors (KNN) algorithm exhibits a rapid training process due to its simplistic nature. Specifically, the training phase solely entails the retention of annotated data without any additional computations. During the prediction phase, the KNN model calculates the distance (Euclidean distance, Manhattan distance, or cosine similarity) of the target data point from all other data points in the training data set. Then those distances are sorted, and the classes of the K nearest neighbors are considered. And the most frequently occurring label is assigned to the target Data point. The prediction process slows as the training dataset grows because each data point's distance from the desired data point must be calculated. Fig 3.12 shows the process of giving the extracted features to the KNN, and Fig 3.12 shows the Process of Predicting using KNN.

3.4.3 Using CatBoost Classifier

CatBoost has the benefit of being capable of handling categorical features automatically. CatBoost is quick and scalable in addition to the above-mentioned advantage. CatBoost makes Predictions based on the predictions of other models. Here initially, weaker models are trained, and then they are improved upon iteratively. These are then combined into a stronger ensemble model to make predictions.

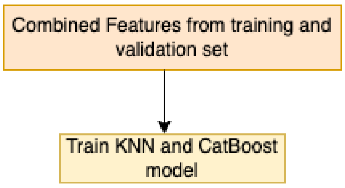

Combined Features from training and validation set
Train KNN and CatBoost model

Figure 3.12: Train the KNN and CatBoost Model by the Training and Val Features

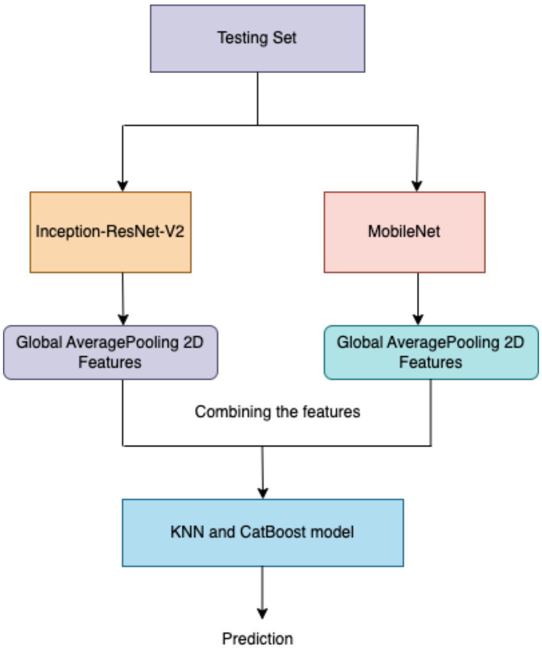


Figure 3.13: Process of Prediction using KNN and CatBoost Model

Fig 3.12 shows the process of giving the extracted features to the KNN, and Fig 3.13 shows the Process of Predicting using KNN.

3.4.4 LIME Explainability

For explainable artificial intelligence (XAI), one prominent library is LIME (Local Interpretable Model-Agnostic Explanations). Its goal is to help people comprehend and make sense of the decisions made by machine learning models by providing explanations that are independent of the specific models used to make those decisions. LIME's primary intent is to produce locally faithful justifications for specific predictions. The goal is to explain why the model produced that result in that circumstance. Since the inner workings of complicated models like deep neural networks are often considered black boxes, this is very helpful when working with such models.

LIME is effective because it approximates the model's behavior close to a given prediction. To do this, it introduces noise into the input features of the instance and tracks how the model's predictions evolve as a result. A fresh dataset with modified instances and associated predictions is produced by this method. To explain the connection between the perturbed instances and their predictions, LIME then fits an interpretable, local surrogate model, like linear regression or decision tree. The surrogate model learns a set of coefficients or rules that are used to construct the explanatory text. LIME's ability to generate explanations for any black-box model, irrespective of its architecture or training algorithm, is a critical feature. It does not need access to the model's internal parameters in order to operate on the model's output probabilities or scores. LIME's adaptability and generalization are enhanced by the fact that it is model-agnostic.

Explainable AI, of which LIME is an example, aspires to make machine learning models more open and comprehensible to the general public. This is especially important in contexts where regulatory compliance, fairness, accountability, or user trust necessitate knowledge of the rationale underlying model decisions. The logic behind a model's predictions can be understood with the help of XAI techniques like LIME.

Chapter 4

Result Analysis and Discussion

We have shown the result of each model with the precision, recall, f1-score and final accuracy. Depending on these parameters, we are understanding the efficiency of our models. The precision value mainly indicates the proportion of positive identification which were actually correct. If any model gives a precision value of 1.00, then it means there are no false positive predictions by the model. In equation (4.1), the formula of precision is shown.

On the other hand, recall value indicates the proportion of actual positives which are correctly classified. If any model gives a recall value of 1.00 for any class, then it means that for that class, the model didn't give any false negative prediction.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

In equation (4.2), the formula of recall is shown. By combining these precision and recall values, if the model is perfect for any class, then it shows 1.00 as the f1-score. The formula of the f1-score is also given in equation (4.3)

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.3}$$

We also used the confusion matrix and explainable AI to analyze our results and the learnings of the models.

4.1 Inception-ResNet-v2 Implementation and Results

Inception-ResNet-v2, a convolutional neural network, was used in our initial test. The learning rate for this particular model was set to be $1e^{-4}$, and the batch size was determined to be 132. After loading the model's ImageNet-trained weights into the Keras library, the model was run for 100 iterations. Both the model's accuracy during training and validation and its losses over training and validation are depicted in Fig 4.1. The best model was then utilized to evaluate the testing set after being saved using Keras Checkpoint with the highest validation accuracy. The model's accuracy on the testing set was 88.47%. Table 4.1 presents the Precision, Recall, and F1 Scores for every class in the testing set of the model. In Figure 4.2, where our four diseases—cardiomegaly, infiltration, nodule, and pneumonia—are displayed, we have also demonstrated the confusion matrix on the testing set for each of the classes.

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	0.99	1.00
Infiltration	0.76	0.80	0.78
Nodule	0.74	0.69	0.72
Pneumonia	1.00	1.00	1.00

Table 4.1: Inception-ResNet-v2 Precision, Recall and F1-Score

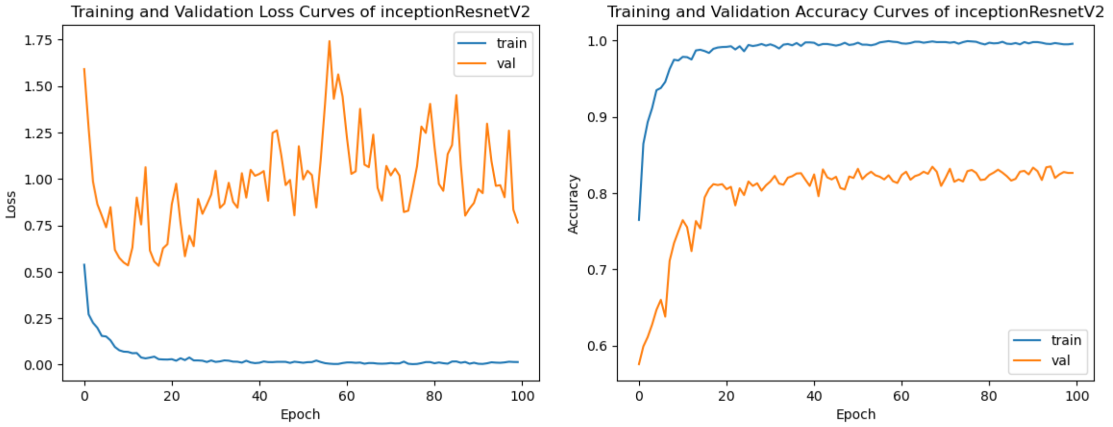


Figure 4.1: Training and Validation Loss & Accuracy Curves of Inception-ResNet-v2

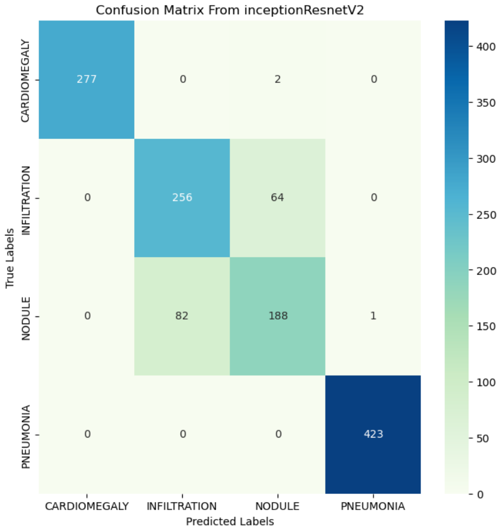


Figure 4.2: Confusion Matrix of Inception-ResNet-v2

4.2 MobileNet Implementation and Results

We used 100 iterations and $1e^{-4}$ as a rate of learning on a testing set of 132 samples to assess the model's performance. After that, we chose to stick with the model that had the best overall validation accuracy. Fig 4.3 shows the model's training and validation losses and accuracy. The maximum testing accuracy for the model was 89.86%. Table 4.2 shows each class's Precision, Recall, and F1 Scores on the model's tests. Once more, Fig 4.4 displays the confusion matrix for the testing for each class.

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	1.00	1.00
Infiltration	0.85	0.72	0.78
Nodule	0.72	0.85	0.78
Pneumonia	1.00	1.00	1.00

Table 4.2: MobileNet Precision, Recall and F1-Score

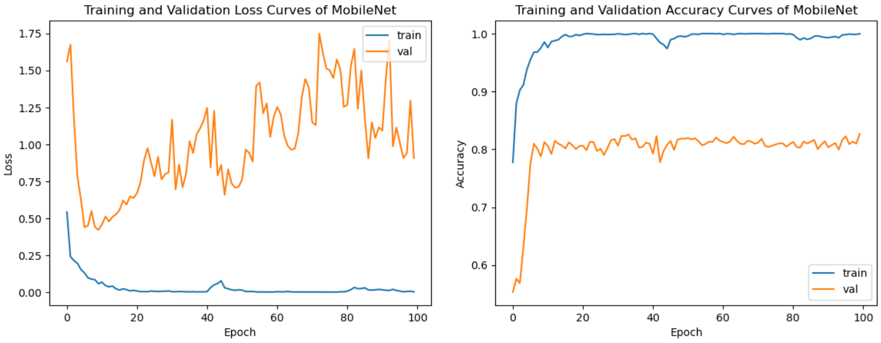


Figure 4.3: Training and Validation Loss & Accuracy Curves of MobileNet

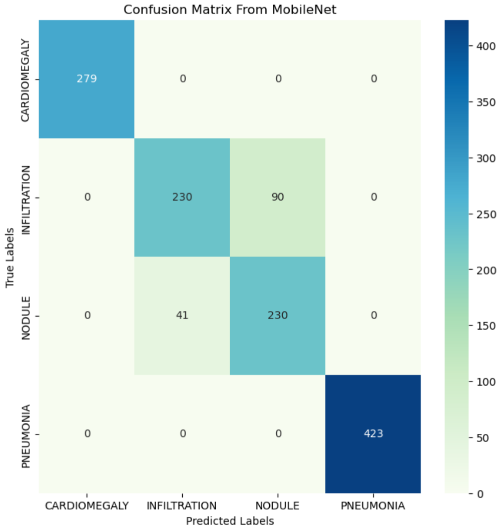


Figure 4.4: Confusion Matrix of MobileNet

4.3 VGG-16 Implementation and Results

VGG-16 was the third convolutional neural network model we used in our research. With batch size set to 132, learning rate set to $1e^{-4}$, and run for 100 epochs, the model was run using the same hyperparameters as our earlier tests with CNN architectures. Testing accuracy for the top model was 87.70%. Figures 4.5 show model losses and accuracy during training and validation. Figure 4.6 shows the confusion matrix for the testing set, while Table 4.3 shows each class's precision, recall, and F1 scores.

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	1.00	1.00
Infiltration	0.88	0.58	0.70
Nodule	0.65	0.91	0.76
Pneumonia	1.00	1.00	1.00

Table 4.3: VGG-16 Precision, Recall and F1-Score

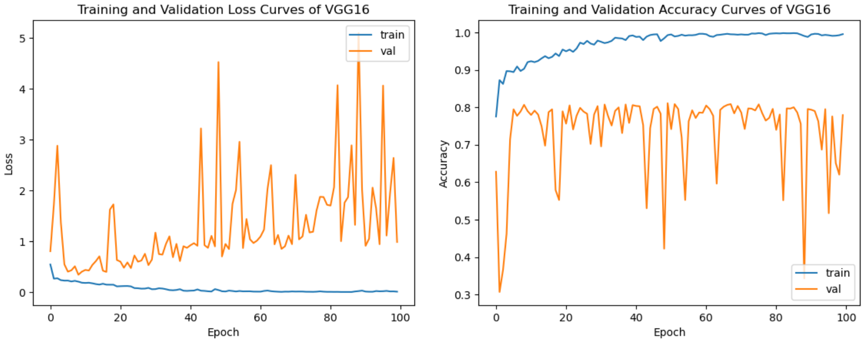


Figure 4.5: Training and Validation Loss & Accuracy Curves of VGG-16

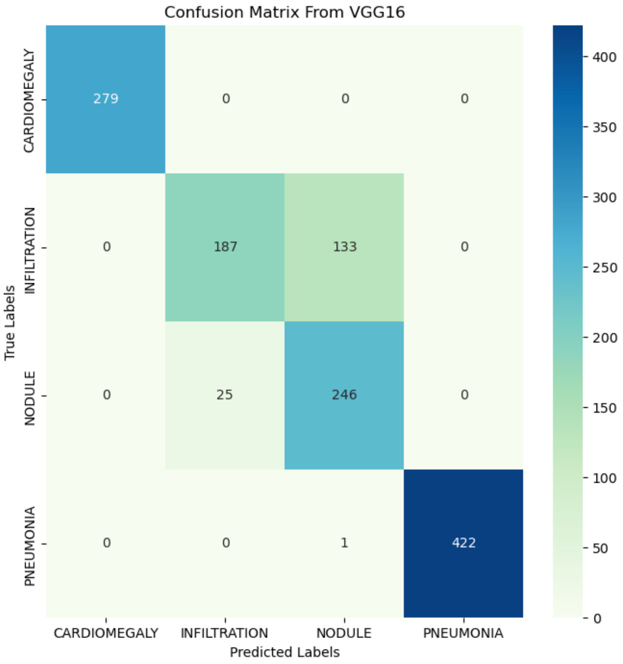


Figure 4.6: Confusion Matrix of VGG-16

4.4 DenseNet-121 Implementation and Results

DenseNet121 was the final CNN model we tested in our experiments. We repeated the tests that we had previously conducted and ran the model with the same hyperparameters. The learning rate was maintained at $1e^{-4}$, and the batch size remained at 132. During the evaluation phase, the models were assessed using a uniform approach, whereby the model with the highest validation accuracy was selected after undergoing one hundred epochs. DenseNet121 had the same accuracy as MobileNet, which is 89.86%. Figure 4.7 shows training and testing loss and accuracy curves. Figure 4.8 depicts the DenseNet121 testing set confusion matrix, while Table 4.4 illustrates the DenseNet121 precision, recall, and F1 scores.

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	1.00	1.00
Infiltration	0.79	0.80	0.80
Nodule	0.76	0.76	0.76
Pneumonia	1.00	1.00	1.00

Table 4.4: DenseNet-121 Precision, Recall and F1-Score

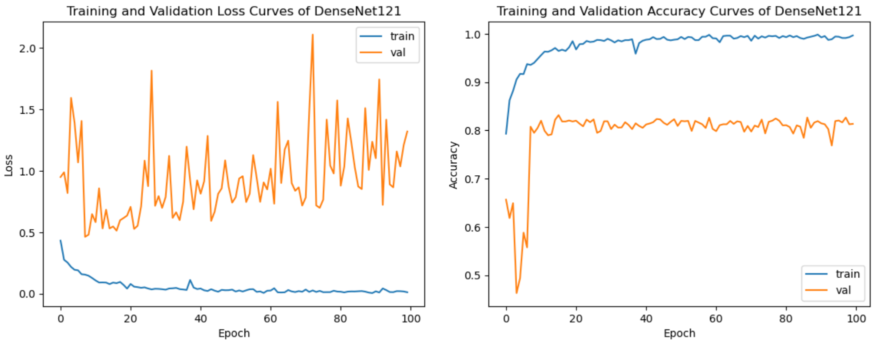


Figure 4.7: Training and Validation Loss & Accuracy Curves of DenseNet-121

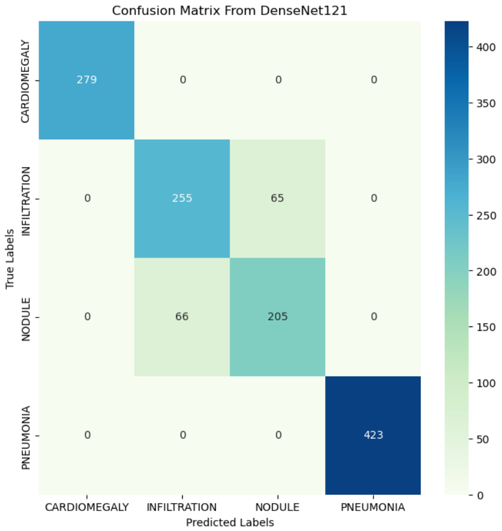


Figure 4.8: Confusion Matrix of DenseNet-121

4.5 Fusion Model Implementation and Results

4.5.1 Combining Features from Two CNN Models

First, we train our Inception-ResNet-V2 and MobileNet models with the training data of our desired dataset. After training both target models, we discard their final hidden layer and reveal their 2D global average. Once the models are exposed to the bottom-most global average pooling layer, they can be used as a feature extractor. Then we extracted the features from each model, which were trained with the training and validation data. At this point, we concatenated the features, and with these features, the ML classifier KNN and CatBoost are trained.

4.5.2 KNN Implementation and Results

Grid Search was used to determine the optimal k in the combined models utilizing the annotated training and validation data, and then KNN was applied to the extracted features. The range in which we've run the grid search is 1 to 200 neighbors, and we got the best k value as 81. So, by giving this value of k in the KNN classifier, we are getting an overall accuracy of 90.56%. Table 4.5 displays the KNN-obtained Precision, Recall, and F1 Scores, while Fig 4.9 shows the confusion matrix.

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	1.00	1.00
Infiltration	0.81	0.81	0.81
Nodule	0.77	0.77	0.77
Pneumonia	1.00	1.00	1.00

Table 4.5: Precision, Recall and F1-Score After Implementing KNN

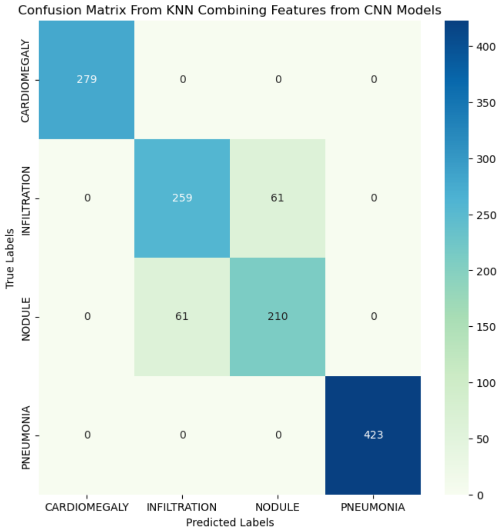


Figure 4.9: Confusion Matrix of KNN Combining Features from CNN Models

4.5.3 CatBoost Implementation and Results

The implementation of CatBoost is similar to that of KNN. After that, instead of training a KNN classifier with the concatenated features, train a CatBoost classifier with the concatenated features obtained from the training and validation data. So, we are giving max_depth as 10 and random_strength as 10, n_estimators as 1000, and the learning rate is 0.1.

So, by using this parameter, we are getting an overall accuracy of 90.33%. The Precision, Recall, and F1 Scores we've achieved through KNN are shown in Table 4.6, and the confusion matrix is also shown in Fig 4.10

Disease Name	Precision	Recall	F1-Score
Cardiomegaly	1.00	1.00	1.00
Infiltration	0.78	0.82	0.80
Nodule	0.80	0.75	0.78
Pneumonia	1.00	1.00	1.00

Table 4.6: Precision, Recall and F1-Score After Implementing CatBoost

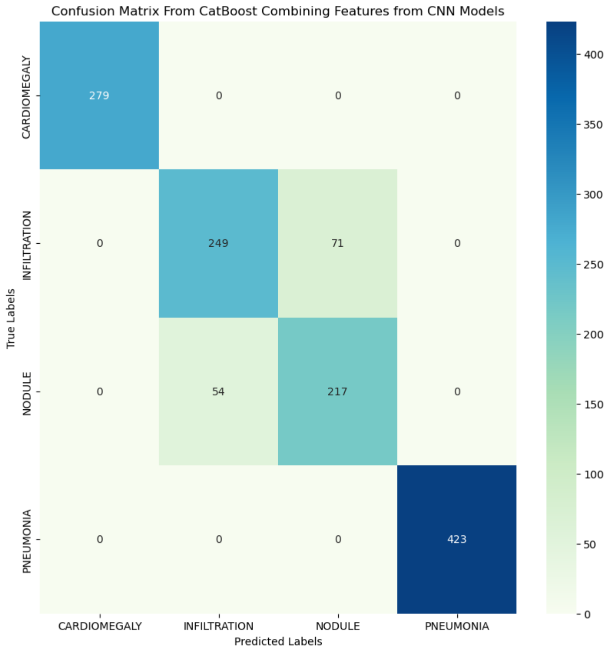


Figure 4.10: Confusion Matrix of CatBoost Combining Features from CNN Models

4.6 Comparative Study

Using the benchmark accuracy and f1-score from another paper, the efficacy of our respective disease detection models for the thorax will be compared (Inception-ResNet-v2, MobileNet, VGG-16, DenseNet-121) to our own. We'll also talk about how the Inception-ResNet-V2 and MobileNet fusion model with the KNN classifier performed.

The overall accuracy of Inception-ResNet-v2 was 88.47%, with improvements in the precision, recall, and F1-score for most thorax diseases. Cardiomegaly, infiltration, and pneumonia were all areas in which it excelled, while nodule detection was an area in which it struggled.

Cardiomegaly and pneumonia were areas where VGG-16 performed particularly well, while infiltration and nodule detection were areas where it struggled. Overall, it was accurate 87.70% of the time.

MobileNet: Precision, recall, and F1-score were all quite good for Mobile Net, contributing to its overall accuracy of 89.86% across numerous diseases. It was perfect at detecting cardiomegaly and pneumonia but not as good at detecting infiltration or nodules.

Cardiomegaly and pneumonia were areas where VGG-16 performed particularly well, while infiltration and nodule detection were areas where it struggled. Overall, it was accurate 87.70% of the time.

DenseNet-121: DenseNet-121 had an overall accuracy of 89.86% and very high precision, recall, and F1-score for most diseases. It did well in detecting cardiomegaly, infiltration, and pneumonia, but not as well in detecting nodules.

Outcomes of the Fusion Model:

Combining the KNN classifier with Inception-ResNet-v2 and MobileNet: When compared to the individual models, the fusion model performed better, with an accuracy of 90.56% being achieved. It scored perfectly in detecting cardiomegaly and pneumonia and did quite well in detecting infiltration and nodules. This demonstrates how well the KNN classifier works when features from different models are combined.

Accuracy Compared to a Benchmark and Another Paper:

The dataset authority established accuracy benchmarks of 0.8141 for cardiomegaly, 0.6128 for infiltration, 0.7164 for nodules, and 0.6333 for pneumonia. In every disease category, your models outperform the state of the art.

The dataset authority established accuracy benchmarks of 0.8141 for cardiomegaly, 0.6128 for infiltration, 0.7164 for nodules, and 0.6333 for pneumonia. In every disease category, our models outperform modern technology.

Identifying cardiomegaly was more accurate (0.925); infiltration detection (0.78), nodule detection (0.735), and pneumonia detection (0.768) were all more accurate than the particular models.

When compared to the gold standard of accuracy, our fusion model performed admirably, with improved precision for the majority of diseases. The fusion model outperformed both the Inception-ResNet-V2 and MobileNet models separately, with an overall accuracy of 90.56%. Therefore, we recommend for this model because it serves as a standard against which future models can be measured. As we didn't find any paper specifically working with the 4 diseases we've worked on, we just took these 4 diseases' accuracy from the dataset benchmark and from a benchmark paper and showed the comparison in Table 4.7, Comparative Study of our proposed model with other models.

Diseases Name	Dataset Benchmark (2017)	ChestXNet (2017)	Proposed Fusion Model
Cardiomegaly	0.8147	0.9248	1.00
Infiltration	0.6128	0.7802	0.81
Nodule	0.7164	0.7802	0.77
Pneumonia	0.6333	0.7680	1.00

Table 4.7: Comparing Our Model to Others

However, the findings of the other paper point to the possibility of further enhancing our model's performance.

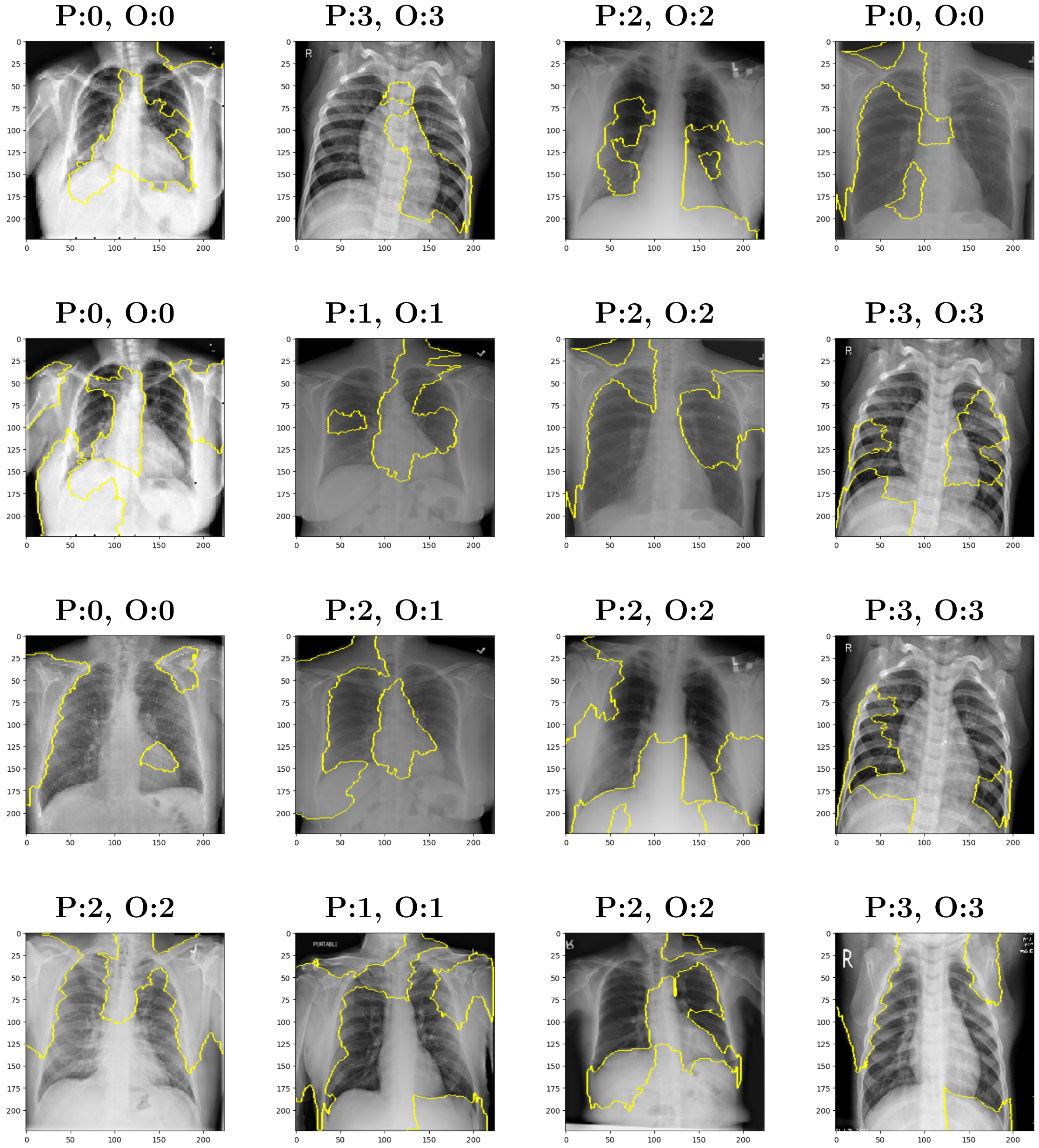
In order to improve the precision of our models, we think about implementing strategies like data augmentation, fine-tuning, and experimenting with alternative architectures and classifiers. CatBoost was also used, but it did not outperform KNN in terms of the f1-score.

Our work shows the value of fusion models and improves thorax disease detection by achieving high accuracy and f1-scores. Even better outcomes and advancements in this field are possible with more research and experimentation.

Please keep in mind that this evaluation relies on the data you shared about the outcomes of your models. When performing a comprehensive comparative study, it is crucial to take into account the particular dataset, evaluation metrics, and other factors.

4.7 Explainable AI

We used the Lime library for Explainable AI for each of the four CNN models that we used in this experiment. Fig 4.11 shows the output of explainable AI respectively for Inception-ResNet-v2, MobileNet, VGG-16, and DenseNet-121 where the yellow marked regions are the features the CNN models used to make the predictions, and P refers to the model's predicted disease, and O refers to the original disease of that X-ray image. The numbers 0,1,2,3, respectively, refer to Cardiomegaly, Infiltration, Nodule, and Pneumonia disease.



P:0, O:0 P:3, O:3 P:2, O:2 P:0, O:0

P:0, O:0 P:1, O:1 P:2, O:2 P:3, O:3

P:0, O:0 P:2, O:1 P:2, O:2 P:3, O:3

P:2, O:2 P:1, O:1 P:2, O:2 P:3, O:3

Figure 4.11: Explainable AI for each CNN Model we used

Chapter 5

Conclusion

It is a highly laborious and lengthy process to diagnose diseases by inspecting X-ray images produced by medical equipment. In order to diagnose anomalies or disorders, you will need the assistance of an experienced radiologist. Both Inception-ResNet-v2 and MobileNet are well-known deep learning models that are utilized for computer vision applications, specifically image classification. These models have been utilized by us for the classification of four distinct diseases, and the results have been compelling. In later iterations of our fusion model, we were able to reach higher levels of accuracy in disease identification.

This kind of work might be valuable to the field of medical science. Because it has the ability to diagnose diseases with not only accuracy that is high but also great precision. It is capable of early detection of diseases. In the future, this might be utilized in the creation of an automated diagnosing system. This kind of work has the potential to provide solutions to screening issues on a large scale. Transfer learning methods and statistical learning methods are making enormous leaps forward on a daily basis. In the not-too-distant future, diagnostic techniques that are similar to our fusion model may be capable of producing significantly improved results.

5.1 Limitations

Our model's accuracy has suffered as a result of a lack of available data. We can achieve much higher precision by adding vaster and many datasets. Overfitting problems have emerged due to a lack of data. We can successfully reduce the amount of loss by using additional data in the training and validation stages. This strategy has the potential to drastically reduce loss during training and validation, making the model more effective. To overcome the current obstacles and improve the model's precision as a whole, it is clear that more and better datasets must be utilized. We can train more accurate and resilient models if we collect data from a wider variety of sources that cover more ground in the space where the problem exists. By highlighting the importance of dataset richness and abundance in optimizing the accuracy of machine learning models, this study contributes to future research and development in this area.

5.2 Recommendation and Future Work

Our future effort will be focused on improving our model's robustness greatly. To do this, we want to investigate new approaches, such as finding better datasets and broadening the classification classes. We expect our model's accuracy to increase significantly when we expand the classes in our dataset. We are also eager to improve the accuracy of our model by combining various classifiers. By utilizing complementing capabilities from various classifier classes, this tactic helps our model perform better overall. We aim to continuously create and improve our fusion model through these iterative phases, increasing the accuracy of the model. The complex issues and changing nature of the problem domain must be addressed by these suggested improvements. We expect our model and its applications in numerous sectors to develop significantly by utilizing this multidimensional approach. This study makes a significant contribution to the discipline, laying the groundwork for more research and providing opportunities for improvement.

Bibliography

[1] H. Yoo, S. Han, and K. Chung, "Diagnosis support model of cardiomegaly based on cnn using resnet and explainable feature map," *IEEE Access*, vol. 9, pp. 55 802–55 813, 2021.

[2] R. Jain, P. Nagrath, G. Kataria, V. S. Kaushik, and D. J. Hemanth, "Pneumonia detection in chest x-ray images using convolutional neural networks and transfer learning," *Measurement*, vol. 165, p. 108 046, 2020.

[3] I. Chamveha, T. Promwiset, T. Tongdee, P. Saiviroonporn, and W. Chaisangmongkon, "Automated cardiothoracic ratio calculation and cardiomegaly detection using deep learning approach," *arXiv preprint arXiv:2002.07468*, 2020.

[4] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman, "Learning to diagnose from scratch by exploiting dependencies among labels," *arXiv preprint arXiv:1710.10501*, 2017.

[5] L. Yao, J. Prosky, E. Poblenz, B. Covington, and K. Lyman, "Weakly supervised medical diagnosis and localization from multiple resolutions," *arXiv preprint arXiv:1803.07703*, 2018.

[6] P. Rajpurkar, J. Irvin, K. Zhu, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[7] Z. Li, C. Wang, M. Han, *et al.*, "Thoracic disease identification and localization with limited supervision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8290–8299.

[8] E. Sogancioglu, K. Murphy, E. Calli, E. T. Scholten, S. Schalekamp, and B. Van Ginneken, "Cardiomegaly detection on chest radiographs: Segmentation versus classification," *IEEE Access*, vol. 8, pp. 94 631–94 642, 2020.

[9] Q. Que, Z. Tang, R. Wang, *et al.*, "Cardioxnet: Automated detection for cardiomegaly based on deep learning," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2018, pp. 612–615.

[10] B. Bercean, S. Iarca, A. Tenescu, C. Avramescu, and S. Fuicu, "Assisting radiologists through automatic cardiothoracic ratio calculation," in *2020 IEEE 14th international symposium on applied computational intelligence and informatics (SACI)*, IEEE, 2020, pp. 000 173–000 178.

[11] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Computer methods and programs in biomedicine*, vol. 187, p. 104 964, 2020.

[12] M. Elkamouny and M. Ghantous, “Pneumonia classification for covid-19 based on machine learning,” in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, IEEE, 2022, pp. 135–140.

[13] K. Kc, Z. Yin, M. Wu, and Z. Wu, “Evaluation of deep learning-based approaches for covid-19 classification based on chest x-ray images,” *Signal, image and video processing*, vol. 15, pp. 959–966, 2021.

[14] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2097–2106.

[15] D. Kermany, K. Zhang, M. Goldbaum, *et al.*, “Labeled optical coherence tomography (oct) and chest x-ray images for classification,” *Mendeley data*, vol. 2, no. 2, p. 651, 2018.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[17] M. Gurucharan, *Basic cnn architecture: Explaining 5 layers of convolutional neural network. up grad blog*, 2020.

[18] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, 2017.

[19] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, “Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery,” *Remote Sensing*, vol. 10, no. 7, p. 1119, 2018.

[20] A. G. Howard, M. Zhu, B. Chen, *et al.*, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[21] A. Pujara, *Image classification with mobilenet*, Jul. 2020. [Online]. Available: https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470%5C%7D.

[22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[23] V. Khandelwal, “The architecture and implementation of vgg-16,” *Towards AI*, vol. 17, no. 8, 2020.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[25] N. Radwan, “Leveraging sparse and dense features for reliable state estimation in urban environments,” Ph.D. dissertation, University of Freiburg, Freiburg im Breisgau, Germany, 2019.