Utilizing Deep Learning architectures for Early Detection of Lung Diseases in Chest X-ray Images

by
Sarder Tanvir Ahmed
19101153
Mohammad Fairoz Mahdin
19101192
Shomtirtha Barua
19101218
Fahim Shariar Abir
19101545

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

Department of Computer Science and Engineering
Brac University
Summer 2023

Decleration

It is hereby declared that

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

Student's Full Name and Signature:

Sarder Tanvir Ahmed
19101153

Mohammad Fairoz Mahdin
19101192

Shomtirtha Barua
19101218

Fahim Shariar Abir
19101545

Approval

The thesis titled "Utilizing Deep Learning architectures for Early Detection of Lung Diseases in CXR Images" submitted by

1. Sarder Tanvir Ahmed(19101153)
2. Mohammad Fairoz Mahdin (19101192)
3. Shomtirtha Barua (19101218)
4. Fahim Shariar Abir (19101545)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for thedegree of B.Sc. in Computer Science on September 17, 2023.

Examining Committee:

Supervisor:
(Member)

Amitabha Chakrabarty, 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

Abstract

COVID-19 and viral Pneumonia are types of lung disease that are highly infectious, and have a tendency of causing an outbreak, that then puts stress on the healthcare sector. When the COVID-19 pandemic first began there were still some working-class people who could not work remotely and so, had to expose themselves to the virus. The COVID-19 pandemic has presented the research community with a complicated problem because of the significant human and financial costs. Consequently, a suitable system is required, that will try to predict if a patient is infected or not by analyzing their X-Ray report. Before the final result of their lung disease is provided to the doctors, patients who are at risk and, having other lung related issues might benefit from having this component of patient evaluation as the results from the tests can be sent through the internet to reduce the danger of infection while traveling to get tested. This thesis has trained 5 models (ResNet18, VGG16, Alexnet, Inception-v3 and Densenet169) to detect COVID-19 and viral Pneumonia in chest X-ray images from the COVID-QU-Ex dataset which yielded 90-95% prediction accuracy. Afterwards, we created an ensemble model using model-level fusion, comprising of pre-trained CNNs ResNet18, VGG16, and Inception-v3 which were the top-performing models. Afterwards, the performance of the model is evaluated by using multiple metrics such as precision, recall, and F1-score was generated. The ensemble yields a prediction accuracy of 95.002% with a recall of 94.543% and main focus of the ensemble model is to utilize it as a supplementary tool by medical professionals for early detection, while ensuring its credibility through explanation generation. This explanation generation is created using AI tools, the code generates a prediction accuracy score, a prediction mask, and a saliency map in an effort to better explain the predictions made by our ensemble model. The saliency map, which highlights the areas in an image that have the greatest impact on a machine learning model's prediction, is created by further processing the predicted mask. The severity and location of lung infections inside the X-ray image are shown on this map, which also illustrates how confident the model is in its predictions.

Keywords: COVID-19, Pneumonia, Chest X-rays (CXR), Image Processing, Data Mining, Deep learning, Convolutional Neural Network (CNN), ensemble;

Acknowledgement

First of all, we would want to thank Almighty Allah for keeping us strong enough—both physically and mentally—to finish our thesis titled” Utilizing Deep Learning architectures to create a Smart Health Monitoring System for COVID-19 Patients” within the deadline. As well as for giving us the information, abilities, and willpower to finish our thesis.

We are thankful for our supervisor, Dr. Amitabha Chakrabarty, Professor of Computer Science and Engineering. His invaluable advice, support, supervision, motivation, and suggestions throughout the thesis-writing process was instrumental. We want to express that his unwavering support allowed us to fully process our work and ideas.

The results of this study was successful not only as a result of individual effort but also as a result of the combined efforts of numerous other people. This research would not have been possible without the overall assistance of our Department of Computer Science. We would want to thank our parents for never giving up on us and continuing to support us while we pursue our jobs since without them, we might not have been able to finish our thesis work.

Table of Contents

List of Figures

List of Tables

Nomenclature

Several symbols and abbreviations that will be used later in the document's body are described in the following list.

AI Artificial Intelligence

CNN Convolutional Neural Network

CXR Chest X-Ray

RT-PCR Reverse Transcription-Polymerase Chain Reaction

SARS-CoV-2 Severe Acute Respiratory Syndrome CoronaVirus-2

Chapter 1

Introduction

ResNet-18, VGG-16, DenseNet-169, AlexNet and Inception-V3 are CNNs that, in previous research papers have been capable of highly accurate image classification tasks. In our study, we propose to use these CNNs to create an ensemble that can detect COVID-19 and viral Pneumonia in CXR images, give the percentage of the area of lung infected and provide explanations to their prediction using AI tools. The first step in our approach is to collect a dataset of CXRs images consisting of normal, COVID-19 and viral Pneumonia infected patients and their respective lung segmentation masks. Next, we will train each CNN on the pre-processed images. We will use the transfer learning approach, which means that we will start with a pre-trained model that has been trained on a large dataset of images. We will then fine-tune the model on our dataset of chest X-rays. Once the CNNs are trained, we will evaluate their performance on a test set of images. Lastly We will use a variety of metrics to evaluate the performance, including accuracy, f1-score and recall. This system could be used to help doctors diagnose COVID-19 and viral Pneumonia in patients get the treatment they need through early detection.

1.1 Motivation

Our intention with this kind of lung disease prediction using AI is to be used as a supplementary tool for healthcare professionals and while they cannot and should not replace clinical diagnosis, they can assist in screening and treating patients. Especially in situations where resources are limited or during another outbreak. If our efforts were acknowledged and supported by mentors and institutions, we could continue to our research and add more accessibility options via various platforms such as a mobile app or a website. This accessibility will help extend the awareness of infectious lung diseases and help contribute to the growing body of knowledge in the field of medical AI.

1.2 Research Problem

The world is now entering 2023 with the knowledge and wisdom of tackling Covid-19, which is still having another wave of infections. Specifically the BF.7 is one of the alarming sub-variants of Omicron that is responsible for the present deadly Covid-19 outbreak in China and it is said to be two or three times more infectious than the original Omicron [6]. It is very evident that with last year's winter being one of the coldest due to global warming, people have been getting sick at a significantly higher rate and very few people have access to the healthcare they would need to get a proper medical diagnosis.

Therefore it is imperative that the world search for a quick and efficient method for a large-scale population test to detect the existence of SARS-CoV-2 that produces COVID-19. Usually people might say, “What is the significance of finding a quicker detection method?”, when improving vaccines and treatment could benefit society more. To that we say, working class people are still doing their responsibilities with the burden of getting sick and most people in developing countries do not have access to rapid antigen test or reverse transcription-polymerase chain reaction (RT-PCR) for COVID-19. Additionally, if a large population of these people becomes severely sick and become hospitalized, there needs to be efficient ways to detect the symptoms of COVID-19 to quickly treat their respiratory diseases.

The affordable alternative to a quicker detection of COVID-19 is “Chest X-rays”, which need fewer resources and have a lower radiation dose, making it possible to repeat the test in order to track the development of the condition. However chest X-rays were not recommended as primary practice for COVID-19 diagnosis, but the World Health Organization (WHO) has advised using chest X-rays if RT-PCR was unavailable or if the findings are delayed [16]. How exactly will the patients who might have regular jobs get notified of the progression of their sickness after taking tests? Or, what are the actions that can be taken and provide us automated updates about the progression of their sickness in the lungs?

First, interpretability is a challenge we have to overcome. It is difficult to comprehend what led an algorithm to make a particular prediction, hence constraining the use of DL for critical life and death situations. Moreover, results in [5] show that patients trust human doctors more than results based on AI. Additionally, Generalization metrics/ability can limit the usage of DL. Due to the natural variation in lung size and other factors, generalization is crucial in medical image analysis. The capacity to extrapolate to patients of all heights, weight and structure is a requirement for efficient intelligence systems.

Furthermore, as the volume of data increases, Deep Learning’s performance gets better. For better performance, Deep Learning currently relies on huge datasets. It is difficult to collect data for several variants quickly, because there is never enough dataset information for lung diseases as its variants keep on changing. In a pandemic response scenario where speedy action is essential, this is problematic. Massive datasets are challenging to produce for various healthcare applications, such as CXR analysis or precise diagnosis. Deep learning methods often employ a small labeled dataset and a large unlabeled dataset.

Finally, data security can be a barrier in the implementation of DL in the medical field. Performance of DL can be enhanced by examining a vast database of patient EHRs, genomes, blood test results and numerous other medical histories which many patients would not perceive secure disclosing this private information to a centralized database that would be vulnerable to hacking. However, chest radiographs by themselves are not much of a problem, but metadata related to it or recorded in electronic health records can be.

1.3 Research Objectives

Thus, it is important to identify the infected people as soon as possible so that we can bring them under quarantine and start the treatment procedure almost immediately. Deep Learning integrated with medical image processing has been investigated to identify COVID-19 and pneumonia using techniques like Convolutional Neural Networks (CNNs) but faces challenges in interpretability, limited data for lung disease variants, and data security concerns in the medical field. So, we are hoping to take advantage of the Deep Learning CNN architectures which can help us in analyzing through image processing such as X-Ray by applying appropriate algorithms. It may help us in detecting the very early stage when there is little amount of particles in the upper respiratory tract and save us from the false negative RT-PCR results. Additionally, X-ray imaging is becoming accessible to the mass of people due to the low cost of facilities. Therefore, our objective is to build a deep learning model trained on a dataset comprised of CXR images and lung masks to see if this is a serviceable substitute to this RT-PCR test for healthcare professionals to utilize.

We have implemented some research objectives as to better explain our goals and more importantly show the milestones we need to surpass. These objectives are directly relevant to conduct our research. Those objectives are:

- Seek to develop a conceptual framework to facilitate COVID-19 and viral Pneumonia identification using deep learning models for image classification.
- Finding the best deep learning model that can be used with the data currently available for COVID-19 viral Pneumonia identification from CXR images.
- Our research will demonstrate how deep learning can be used to identify COVID-19 viral Pneumonia, which may be used as a supplementary tool to help medical practitioners choose the most appropriate course of treatment.

Chapter 2

Literature Review

Deep learning (DL) is a subset of machine learning. To extract higher-level features from the raw input; it consists of many neural network layers that, in higher layers, may recognize more specialized characteristics whereas lower layers may distinguish edges and lines. Each layer uses its predecessor's output as its input in a sequential fashion [7]. Over 766 million individuals have COVID-19 infection, and there have been close to 6.9 million fatalities thus far (as of 22nd May, 2023). A critical technique for properly managing this pandemic is early identification, patient isolation, and care[1]. In a final effort to stop the COVID-19 pandemic, research has been started on scientific experiments in all fields and Deep Learning integrated with medical image processing has also been carefully investigated to come to a conclusive conclusion. Pneumonia is a rapid-onset lung infection characterised by inflammation of the air sacs and fluid accumulation in the lungs. It can be caused by various microorganisms such as bacteria, viruses, or fungi. Each year, pneumonia impacts around 7.7% of the global population. Pneumonia is a significant cause of mortality, particularly among children under the age of five, accounting for more than 15% of deaths in this age group.[27] It is prevalent in underdeveloped and developing nations, where factors like overcrowding, pollution, and poor sanitation contribute to its high incidence. Due to limited healthcare resources in these regions, early detection and effective management are crucial in preventing severe complications and fatalities. Diagnostic imaging techniques such as computed tomography (CT), magnetic resonance imaging (MRI), and X-rays are commonly employed to examine the lungs and facilitate accurate diagnosis. One of the top priority research fields is disease categorization using medical image analysis. An Artificial Intelligence-aided diagnosis system can be developed to accurately interpret COVID-19 cases from the input X-ray image with the assistance of qualified radiologists and based on the properties of X-ray images discussed above. With this goal in mind, several studies have been carried out that will help with the speedy screening of COVID-19 using artificial intelligence and deep learning. Data scientists have been focusing their efforts on enhancing the identification, analysis, and predictive capabilities of diseases by utilizing CXR image datasets and various data classifiers. These studies are centred on using CNNs, a kind of deep learning technique that processes the CXR images. It is a method that involves leveraging a pre-trained model with its original architecture and learned weights intact. In this phase, the CNN algorithm is utilized to process a collection of images of a different kind and extract features using the knowledge it has gained about feature extraction from its initial training. Therefore, the extracted features are then incorporated into a new network that performs the desired categorization task. This approach is frequently used to avoid the high computational costs associated with training very deep networks from scratch or to keep the relevant feature extractors developed throughout the first phase [8]. When conducting research using a pre-processed dataset, it is critical to undertake a thorough literature analysis of previous research articles that have used the same dataset. We acquire vital insights into past researchers' approaches, models, and techniques by analysing the current literature. This enables us to build on their work, identify any gaps or restrictions, and contribute to the field's current knowledge. Furthermore, reviewing the findings and conclusions of past studies provides us with a baseline against which to evaluate the performance and generalization of our own approach. It allows us to compare our findings to those of other researchers, identify areas for improvement, and evaluate the robustness of our results.

In the recent past, CNN architectures have succeeded to gain similar levels of human expertise in a variety of composite visual categorizations including image evaluation and recognition. Since the very first successful CNN architectures have come under recognition which is LeNet, created by Yann LeCun in 1998, was utilized for handwritten digit recognition; since then, a variety of CNN architectures have emerged throughout the past two decades [9]. Though compared to current advanced CNN models, LeNet is considered to be a mere architecture containing three convolutional, two average pooling and two fully connected layers. The CNN architectures engaged in the reviewed studies, as well as their application and COVID-19 detection outcomes are briefly described in the following subsections.

2.1 Deep Learning Classifiers

Medical picture identification, classification and diagnosis techniques based on artificial intelligence (AI) have been introduced and applied. Recent AI invention has substantially enhanced COVID-19 judgments and vaccination, performing in excellent scaling and prompt response, dependable and effective issues and occasionally surpassing humans in specific healthcare conditioning. AI solutions are now being used to address a wide variety of biological issues and complications, including brain tumor diagnosis, lung illness detection and other oncological emergencies [4].

The two most well-known artificial intelligence (AI) fields are machine learning and deep learning. Deep learning is a set of machine learning methods that mostly focus on automatically extracting and classifying picture information. The application of DL in current scientific research has surged due to their ability to learn and adapt from context. By finding and detecting the traits needed to make clinical choices, DL can manage complicated data representations and resemble qualified medical professionals. Deep learning approaches can be classified into three main methods: supervised, semi-supervised, and unsupervised. In supervised learning, known input values are used to predict the desired output.

This approach is commonly employed in classification tasks such as traffic signal recognition, speech-to-text conversion, and facial recognition. Semi-supervised learning utilizes both labelled and unlabelled data for training, with the unlabelled data significantly enhancing learning accuracy when combined with a small portion of labelled data. On the other hand, unsupervised learning aims to uncover patterns or connections among data points without any predefined labels, allowing for categorization or anomaly detection. In the domain of security, unsupervised learning is frequently employed to identify unusual or anomalous behavior.

Recent investigations for COVID-19 detection, evaluation, and categorization have used DL-based techniques. These methods include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). CNNs, in particular, have demonstrated exceptional effectiveness in a variety of medical images segmentation applications [2][3]. In order to identify pediatric pneumonia from Chest X-Ray images, CNN can enhance imaging quality in low-light images, which is crucial for the deep learning algorithm's accuracy. However, there are few instances when DL might fall short and can be problematic for its application in identifying COVID-19.

AlexNet: The necessity to enhance the outcomes of the ImageNet challenge led to the creation of AlexNet [10]. With an accuracy of 84.7 % this was one of the first Deep Convolutional Networks to do well on the 2012 ImageNet LSVRC-2012 challenge. The authors used data augmentation in AlexNet and a dropout regularization technique with a view to solving the overfitting issue.

DenseNet: A CNN network with a densely linked topology named as DenseNet was proposed by Huand et al. in 2017.DenseNet, a type of CNN network that utilizes dense connection between layers with the presence of Dense Blocks [11]. DenseNet was developed with a view to having smooth and intact interaction of data with input and output layers. DenseNet consists of transition layer, dense block, convolutional layer and fully connected layer.

ResNet: This is very interesting to know that since 2012, when after AlexNet winning the Image net competition we have seen a pattern that every new subsequent architecture has used more layers in their deep neural network with a view to dropping the error rate [12]. But this increasing number of layers help for only a few layers. As the layers increase in number, a problem arises known as Vanishing gradient. This problem makes the gradient too large or makes the gradient to zero.

VGGNet: Researchers from the University of Oxford introduced the CNN architecture known as VGGNet (Visual Geometry Group) in 2014 [13].It is a typical deep CNN design that has several layers. The VGG16 algorithm, known for its impressive accuracy rate of 92.7%, demonstrates excellent performance in identifying and classifying 1000 photos across a wide range of 1000 distinct categories. Leveraging transfer learning, this technique has gained popularity as an effective and user-friendly method for image classification.

Inception: The researchers propose a deep convolutional neural network architecture called ”Inception,” which achieved state-of-the-art results in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014) for classification and detection tasks. The distinctive feature of this architecture is its efficient utilization of computing resources within the network. This was accomplished by carefully designing the network to increase its depth and width while maintaining a constant computational budget. One specific version of the architecture used in their submission for ILSVRC 2014 is referred to as GoogLeNet, which is a 22-layer deep network [28].

MobileNetV2: Designed by Google, is optimized for mobile and embedded vision applications. Its lightweight architecture is beneficial for this classification task due to its efficiency in terms of computation and memory usage. It's a practical choice when resource constraints are a concern [29].

EfficientNet: EfficientNet, developed by Google, introduces a compound scaling method that optimizes model depth, width, and resolution simultaneously. It provides a balance between accuracy and computational efficiency, making it valuable for tasks where both high performance and resource constraints are important [30].

SqueezeNet: SqueezeNet is a lightweight neural network architecture designed for real-time object recognition. Its compact size and efficient design make it useful for deployment on edge devices or in scenarios where computational resources are limited. In the context of chest X-ray classification, it can provide reasonably accurate predictions while keeping computational demands low [31].

Xception: Xception is another architecture from Google that extends the inception concept by separating the cross-channel and spatial correlations. This design is useful in chest X-ray classification tasks as it allows the model to capture complex spatial relationships and patterns within the images [32].

2.2 Related Works:

Razzak et al. used AlexNet to categorize COVID-19 instances in a binary and multiple manner. In the binary classification task of distinguishing COVID-19 from healthy cases, they achieved an accuracy of 97.04% on the test set. Furthermore, for the multiclass classification task encompassing COVID-19, healthy, bacterial pneumonia, and viral pneumonia, their accuracy reached 63.27% [13].

Hilmizen et al. used a combination technique to categorize CXR images by combining two different transfer learning models. The authors divided 2,500 CXR images into two groups: normal and COVID-19 using the following algorithms: Densenet-121, Mobile Net, X-ception, Inception-v3, ResNet-50, and VGG-16. According to the outcomes of the experiments, the ResNet-50 and VGG-16 obtained the greatest accuracy of 99.87% [14].

P. K. Sethy et al. retrieved the features from various pre-trained architecture of CNN using chest X-ray pictures. The ResNet50 classifier combined with the Support Vector Machine (SVM) classifier to create 50 samples, of which 25 were normal and the remaining 25 were COVID-19 instances. This combination produced the highest accuracy (95.38%) [15].

Demir, F. et al. proposed an efficient approach for the classification of lung diseases using Convolutional Neural Networks (CNNs). Their study, published in Health Information Science and Systems, presents a promising method for automated lung disease diagnosis, showcasing the potential of deep learning in medical image analysis. The prediction accuracy for the first and second proposed methods were 65.5% and 63.09%, respectively [33] .

S. H. Karaddi and L. D. Sharma employed eight pre-trained CNNs: Alexnet, Darknet-19, Darknet-53, Densenet-201, Googlenet, InceptionResnetV2, MobilenetV2, and Resnet-18. Densenet-201 outperformed the others, achieving a peak accuracy of 97.49% for K=10, along with high sensitivity (95.57%) and specificity (97.96%). These results demonstrate the method's superiority over existing techniques and its potential to aid clinicians in diagnosing lung conditions promptly, enabling timely treatment decisions. [34]

From Table 1, we can see information about researcher papers that we reviewed, and the models they implemented with their prediction accuracy.

Table 1: Research papers on COVID-19 Detection using CXR images

#	Authors	CNN Model	Ref	Accuracy
1	Szegedy, C. et al.	Inception	[28]	97.72%
2	Razzak, I. et al.	K-fold	[13]	99.25%
3	Kuzinkovas, D. et al.	DenseNet	[11]	98.82%
4	Sethy, P.K. et al.	ResNet50, ResNet101, Inception$_{R}esNetv2$	[15]	95%
5	Makris, A., et al.	VGG16, VGG19, MobileNetV2 and DenseNet201	[16]	95%
6	Apostolopoulos, J. D. et al.	VGG16, VGG19, COVID-Net	[8]	96.78%
7	Schraut, J.X. et al.	U-Net	[19]	97.72%
8	Sanida, T. et al.	Hybrid DCNN	[20]	99.25%
9	Kuzinkovas, D. et al.	Ensemble(ResNet50, VGG19, VGG16)	[21]	98.82%
10	Nafisah, S.I. et al.	EfficientNetB7	[22]	99.82%
11	Ibrokhimov, B. et al.	VGG19, ResNet50	[23]	96.6%
12	Anas, M.T. et al.	VGG19, U-Net, U-Net++, and Feature Pyramid Networks	[24]	99%
13	Mutyambizi, M. N. et al.	ResNet50	[25]	94.14%
14	Marios, C. et al.	ResNet50, ResNet101, DenseNet121, DenseNet169, InceptionV3	[26]	93%
15	Demir, F. et al	ResNet50, Alexnet, VGG-16	[33]	62%
16	Karaddi, S. H. et al	lexnet, Darknet-19, Darknet-53, Densenet-201, Googlenet, InceptionResnetV2, MobilenetV2, and Resnet-18.	[34]	97.54%
17	Ghosh, K. K. et al	InceptionResnetV3.	[35]	99.34%

During the literature review, we delved into various research papers focusing on deep learning techniques for lung disease categorization using medical image analysis. The utilization of CNNs in processing chest X-ray (CXR) images has been a prominent area of research, especially in the context of lung disease detection. We found that CNN architectures, such as AlexNet, DenseNet, ResNet, VGGNet, and InceptionV3, have shown promising results in accurately classifying lung diseases from CXR images. Additionally, we stated some related works where researchers employed CNN models for COVID-19 detection using CXR images and these models leveraged transfer learning to extract features from the images and achieved high accuracy. In our study, we aimed to contribute to the existing body of research by exploring the potential benefits of integrating AI-driven model prediction explanations and we believe that our work offers a valuable perspective on enhancing the comprehensibility of AI-driven models. Lastly in our proposed methodology, we investigated and described the layer architectures of the various CNNs.

Chapter 3

Dataset

3.1 Description:

We are utilizing 33,920 CXR images and their respective lung-masks as our data, which we collected from "COVID-QU-Ex", from the data science platform, Kaggle. This dataset has been recognized and awarded by the Kaggle Machine Learning and Data Science community. [4] [18] The dataset includes 11,956 COVID-19 pictures, 11,263 non-COVID infections, and 10,701 normal cases, these are the data points. The collection is notable for including ground-truth lung segmentation masks, making it the most comprehensive lung mask dataset ever constructed. Therefore,the features are the pixel values of the CXR images and the lung segmentation masks and the number of features would depend to the dimension of the image, where 299x299 would mean each image carries 89,401 features.

The model will learn to associate certain patterns and combinations of pixel values with these classes during training. These pixel values are considered as quantitative as each pixel's value represents the intensity or brightness of light at a specific location in the image. These values are typically continuous and can range from 0 (black) to 255 (white) in grayscale images. This dataset will be utilized for classification, as we will attempt to train our models to be able detect and localize COVID-19 and viral Pneumonia infections in chest X-ray images. Additionally, COVID, normal, and viral pneumonia, are the labels that the deep learning model aims to predict based on the learned features from the CXR images. Lastly, in the COVID-QU-Ex dataset, all the classes have an equal number of instances and a data point would correspond to a single chest X-ray (CXR) image along with its associated information which are:

- The actual CXR image data, which consists of pixel values representing the X-ray image.
- Ground-truth lung segmentation masks, which indicate the boundary or region of the lungs in the image.
- Information about the class of the image, such as whether it's a COVID-19 image, a normal image, or an image showing viral pneumonia.

3.2 Dataset Pre-processing

In our endeavour to effectively analyse X-ray images for the detection of lung infections, we encountered two significant challenges that compelled us to employ a series of innovative solutions. Firstly, the dimensions of the X-ray images provided in our dataset were substantial, exceeding the manageable scale for our available computational resources. Training deep learning models on these large images would have been computationally intensive and time-consuming. Additionally, we were keen on diversifying our dataset to ensure robust model performance. Despite having access to a dataset comprising 33,000 instances, we recognized the need for greater image variation to enhance the model's ability to generalize. To tackle these issues head-on, we implemented a comprehensive pre-processing pipeline that significantly enriched our dataset while addressing the computational constraints.

Our pre-processing pipeline, depicted in the provided code snippet, entails a series of meticulous transformations. Firstly, we resized the images to a more manageable scale of 299x299 pixels, which not only reduced the computational burden but also standardized the image dimensions for consistency. Moreover, we harnessed the power of data augmentation techniques to inject diversity into our dataset. Random horizontal and vertical flips, random rotation up to 30 degrees, and controlled color jitter, including adjustments in brightness, contrast, saturation, and hue, were applied as shown in Table 2. These augmentations not only introduced variations in the images but also augmented the dataset with each epoch during training.

Table 2: Preprocessing Pipeline

Data Augmentation	**Parameters**
Resize	Size: (299, 299)
Random Horizontal Flip	-
Random Vertical Flip	-
Random Rotation	Degree: 30
Color Jitter	Brightness: 0.4 Contrast: 0.4 Saturation: 0.4 Hue: 0.1
To Tensor	-
Normalize	Mean: [0.485, 0.456, 0.406] Std: [0.229, 0.224, 0.225]

The dataset has been further separated using Stratified splitting, where training, testing, and validation sets are made to support successful training and evaluation of the CNN models. The CXR images are divided in an 80:20 ratios, guaranteeing that 80% of the images are used for training and 20% for testing and validation, as shown in figure 1.

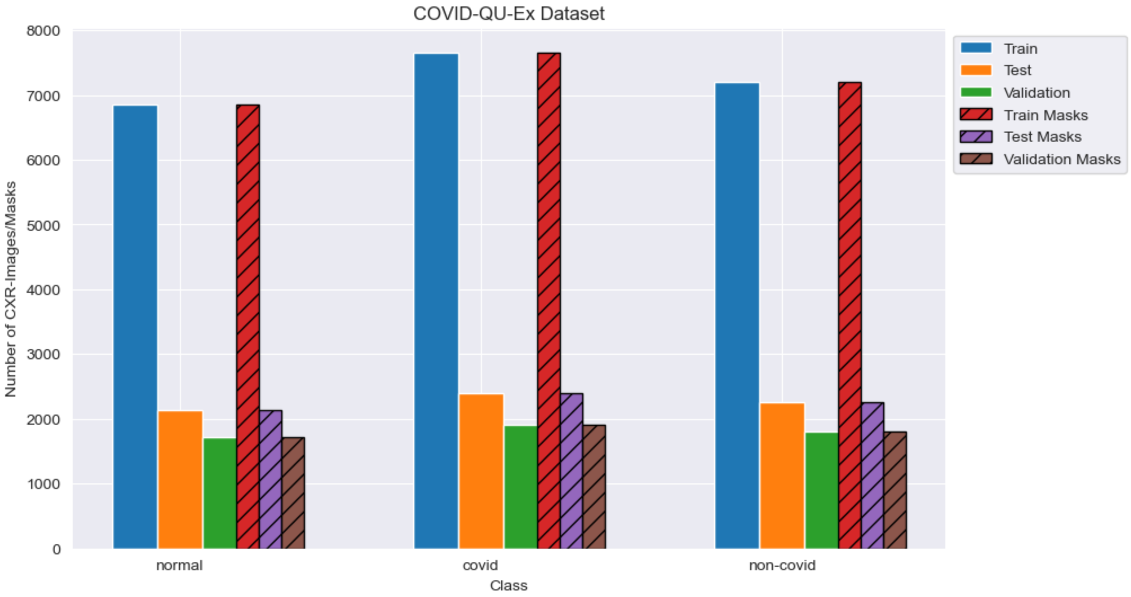


Figure 1: COVID-QU-Ex Dataset Infograpgic

We can see the data has been batched for more efficient processing during model training and evaluation. In the training phase, there are a total of 5429 batches, while the testing and validation phases have 1697 and 1355 batches, respectively. This batch-wise organization enhances the computational efficiency of the model training and testing procedures. This meticulous dataset separation, alongside stratified splitting, serves as a strong foundation for the subsequent CNN model training and evaluation, ensuring that the models are robust and capable of accurate classification tasks for lung diseases as seen in Table 3.

Table 3: Dataset Summary

Class	**Train**	**Test**	**Validation**
Covid	7658	2395	1903
Non Covid	7208	2253	1802
Normal	6849	2140	1712

Chapter 4

Methodology

4.1 Proposed Methodology:

4.1.1 DenseNet-169:

DenseNet-169 is CNN with a depth of 169 layers. It incorporates convolutional layers, batch normalization layers and ReLU activation functions. The convolutional layers are responsible for extracting features from the input image, while batch normalization layers ensure the normalization of feature maps and the ReLU activation functions introduce non-linearity to enhance the network's learning capability. By leveraging these components, DenseNet-169 can effectively capture and represent complex features in image processing. DenseNet169 can learn more complex features from the input image, is less vulnerable to overfitting, and is easier to train than other CNN models. However, it is computationally expensive to train and it can be difficult to tune its hyper parameters. From Figure 2, we can see our interpretation of the layer architecture of DenseNet-169, made with latex which we took reference from the original researcher.

$$X_{l+1} = H_l\left([X_0, X_1, X_2, \ldots, X_l]\right) \tag{1}$$

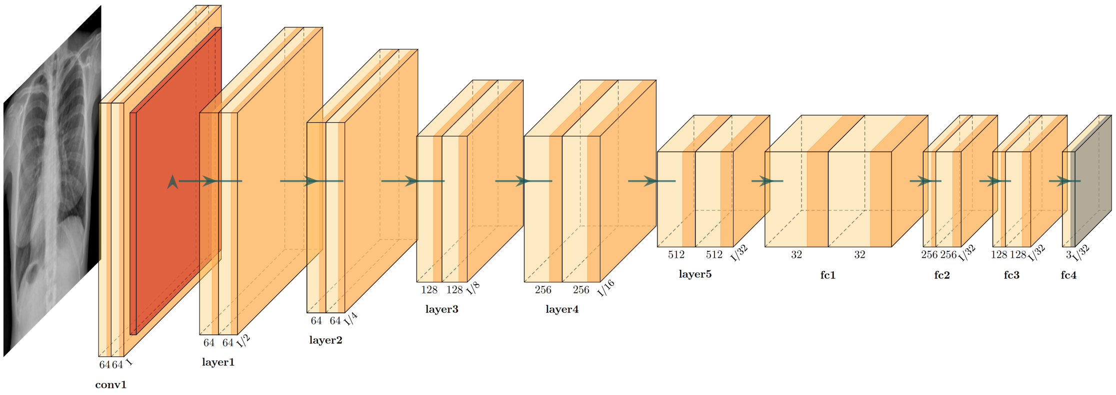


Figure 2: Layer Visualization of Densenet-169 Model

4.1.2 RezNet-18:

By utilizing residual blocks, ResNet-18 enables the network to learn residual mappings and fixes the issue of gradients diminishing during training. Comprising 18 layers, each layer of ResNet-18 consists of a convolutional layer followed by batch normalization and ReLU activation. The initial two layers employ a 7x7 kernel size, while the subsequent layers use a 3x3 kernel size. Moreover, the architecture includes two max pooling layers that downsample the feature maps by a factor of 2. From Figure 3, we can see our interpretation of the layer architecture of Resnet-18, made with latex which we took reference from the original researcher paper.

$$y = F(x, \{W_i\}) + x \tag{2}$$

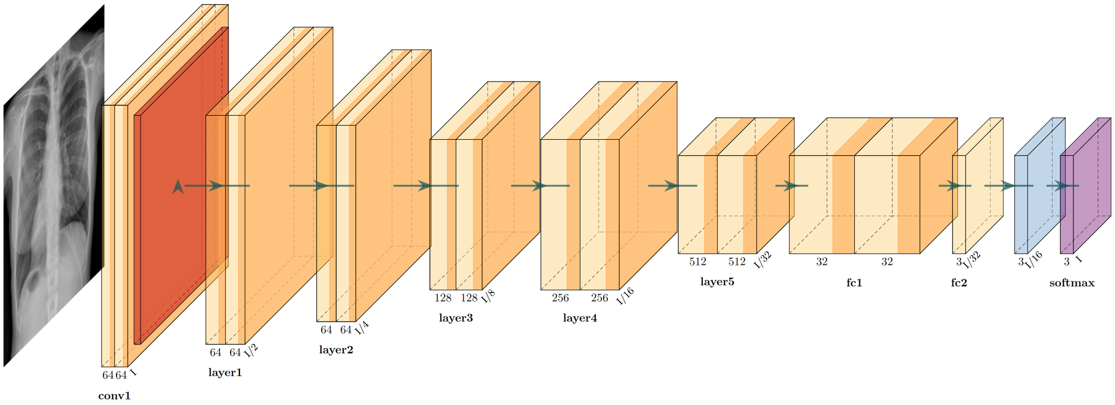


Figure 3: Layer Visualization of ResNet-18 Model

4.1.3 VGG-16:

VGG-16 is a deep CNN with 16 layers, consisting of convolutional layers, max pooling layers, and fully connected layers. The convolutional layers play a vital role in extracting informative features from the input image. The max pooling layers help downsample the extracted feature maps, reducing their spatial dimensions while retaining important features. Finally, the fully connected layers utilize the extracted features to classify the input image into various categories. From Figure 4, we can see our interpretation of the layer architecture of VGG-16, made with latex which we took reference from the original researcher paper.

$$Y = \sigma \left(W * X + b \right) \tag{3}$$

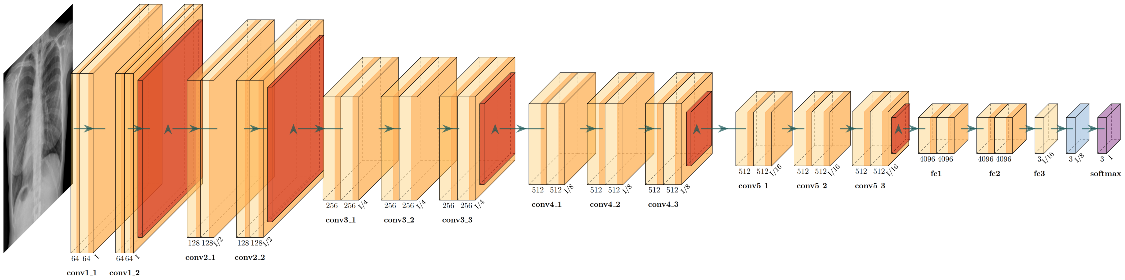


Figure 4: Layer Visualization of VGG-16 Model

4.1.4 Alexnet:

The aim to fine-tune Alexnet for a specific classification task while leveraging the pre-trained weights learned from ImageNet, alongside newer models allows us to showcase the contrast between the performance of older and more recent CNN architectures, highlighting the progress made in the field and providing valuable insights into the evolution of deep learning techniques. From Figure 5, we can see our interpretation of the layer architecture of Alexnet, made with latex which we took reference from the original researcher paper.

$$Y = f\left(\sum_{i=1}^{n} W_i * X_i + b\right) \tag{4}$$

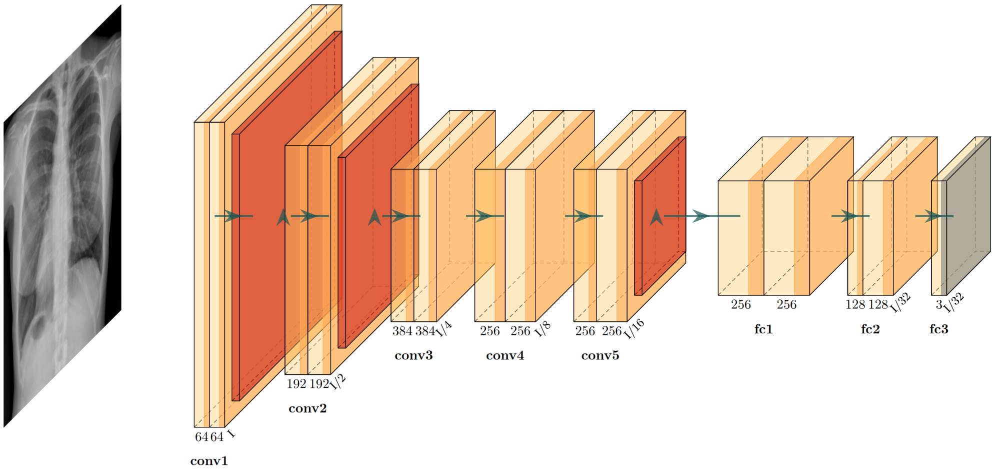


Figure 5: Layer Visualization of AlexNet Model

4.1.5 Inception V3:

The primary innovation of Inception-v3 lies in its utilization of ”inception” modules, which are designed to capture features at different scales by using various kernel sizes (1x1, 3x3, 5x5) within the same layer. These inception modules allow the network to efficiently capture complex patterns and hierarchies in images, enabling more accurate and robust classification. From Figure 6, we can see our interpretation of the layer architecture of Inception-v3, made with latex.

$$Inception(X) = [Concat(Branch_1(X), Branch_2(X), Branch_3(X), Branch_4(X))] \tag{5}$$

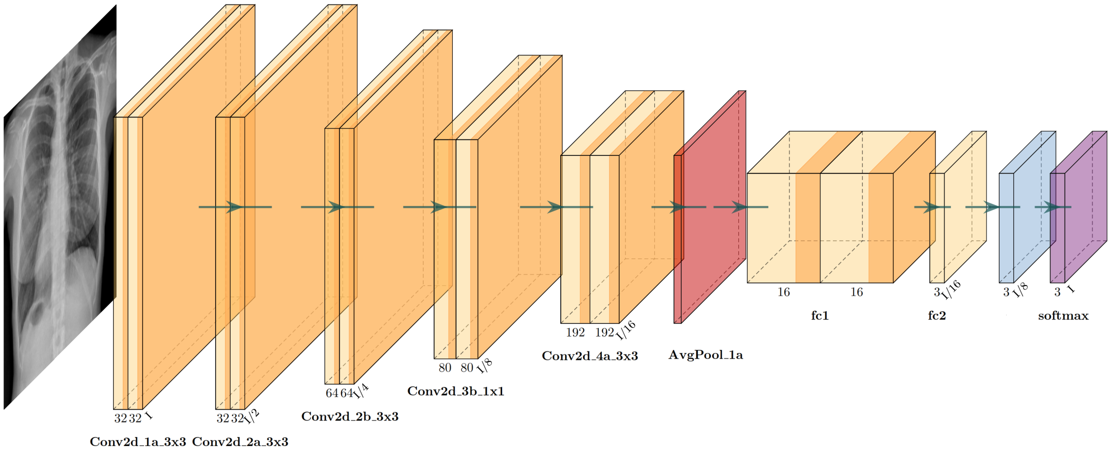


Figure 6: Layer Visualization of Inception-V3 Model

4.2 Working Plan:

After going through a lot of datasets we found a proper dataset with help of our thesis instructor. That data set consists of COVID-19, pneumonia and normal CXR images and their lung masks as shown in Figure 7. Then we used pre-processing techniques such as data augmentation to expand the dataset's diversity and enhance the model's ability to apply its knowledge to new or unfamiliar data. Techniques like rotation, scaling, flipping, and adding random noise were used to generate new variations of the original images, enabling the model to learn more generalized and invariant features from different variations of the original images.

Additionally, we went through a lot of research articles about deep learning algorithms and their results. Then we chose Inception-v3, VGG-16, and ResNet-18 for our ensemble model because of their strong performance. DenseNet-169 was selected for its enhanced feature and reuse capabilities through dense connections also allowing efficient utilization of features from preceding layers. It offers parameter efficiency and capturing fine-grained details which has shown promising results in similar tasks. VGG-16 was chosen for its simplicity and interpretability. It's straightforward architecture of stacked convolutional and pooling layers which makes it easy to understand and analyze. VGG-16 has demonstrated good generalization and has been widely used for various image classification tasks.

ResNet-18 was preferred for its deep architecture and introduction of residual connections. These connections reduce the vanishing gradient problem and enable successful training of deeper networks. ResNet-18 has shown strong performance, high accuracy and also suitable for transfer learning with pre-trained models. InceptionV3 was chosen for its innovative inception modules, which utilize multiple filter sizes within the same layer. This enables the network to capture features at different scales, making it particularly effective in handling both fine-grained and large-scale patterns in the data. The inception modules facilitate a more efficient use of computational resources by reducing the number of parameters compared to other deep architectures. We also selected AlexnetV3 to provide us with more context with how different models perform for this classification task. After training these models and the ensemble on the train dataset, we used the test dataset to generate the models' metrics and produce explanations for their predictions.

4.3 Transfer Learning:

Transfer learning is a highly valuable technique in the field of deep learning that has proven to be particularly effective in medical image analysis, including the critical task of COVID-19 detection from CXR images. In this context, transfer learning involves leveraging pre-existing models that have been trained on extensive datasets, such as ImageNet, which contains millions of labeled images across various categories. These pre-trained models have learned intricate and discriminative features from their training data, enabling them to capture essential visual patterns.

Firstly, transfer learning offers several benefits. Firstly, pretrained models are already trained on large-scale datasets such as ImageNet, which contain diverse images from various categories. Second, transfer learning helps us get over the drawback of having little training data. For specialized tasks like lung disease identification, a large amount of labeled data may not be easily accessible when training deep learning models from start. We can take use of their learnt representations by starting with pretrained models and hone them on the relatively smaller CXR dataset. By

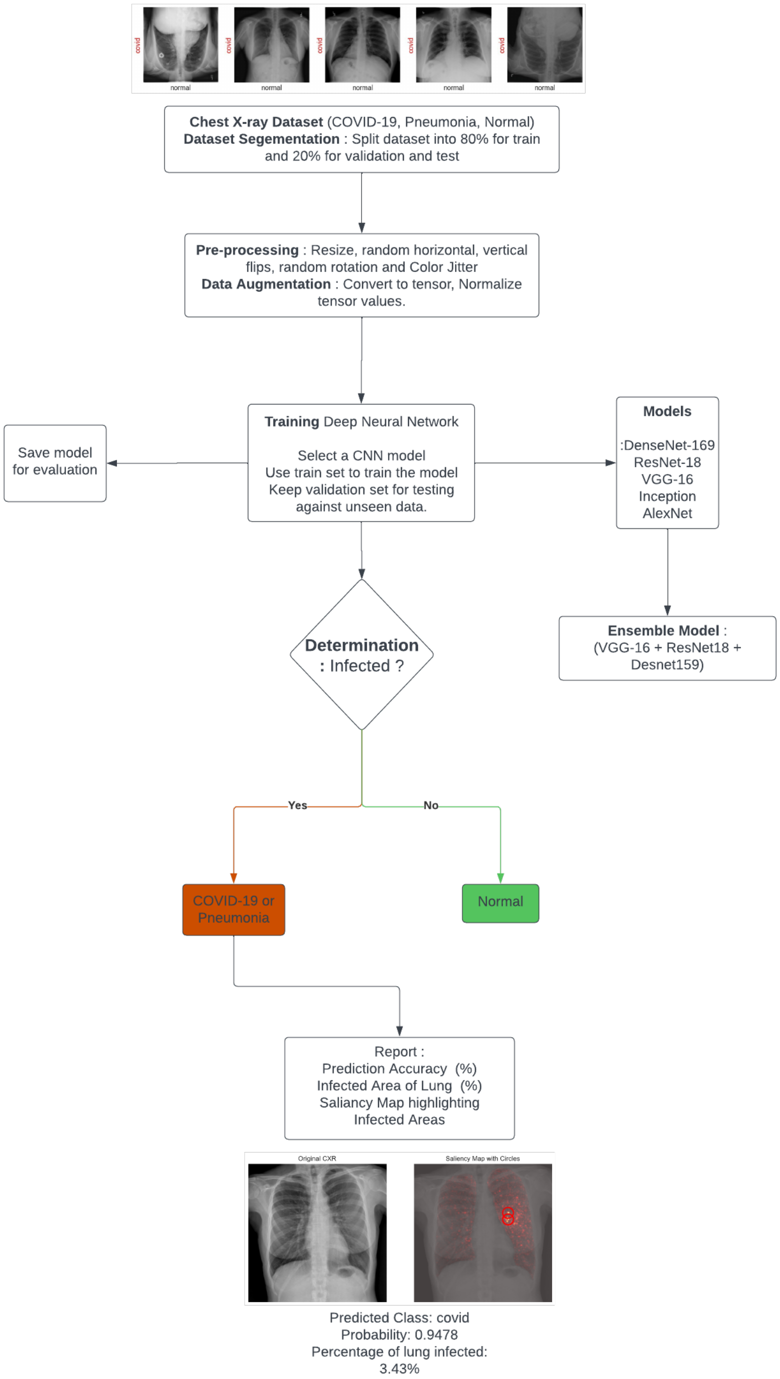

covid
normal
covid
normal
covid
normal
covid
normal
covid
normal
Chest X-ray Dataset (COVID-19, Pneumonia, Normal)
Dataset Segementation : Split dataset into 80% for train and 20% for validation and test
Pre-processing : Resize, random horizontal, vertical flips, random rotation and Color Jitter
Data Augmentation : Convert to tensor, Normalize tensor values.
Training Deep Neural Network
Select a CNN model
Use train set to train the model
Keep validation set for testing against unseen data.
Save model for evaluation
Models
:DenseNet-169
ResNet-18
VGG-16
Inception
AlexNet
Ensemble Model : (VGG-16 + ResNet18 + Desnet159)
Determination : Infected ?
Yes
No
COVID-19 or Pneumonia
Normal
Report :
Prediction Accuracy (%)
Infected Area of Lung (%)
Saliancy Map highlighting Infected Areas
Original CXR
Saliency Map with Circles
Predicted Class: covid
Probability: 0.9478
Percentage of lung infected: 3.43%

Figure 7: Flowchart of Working Plan

using this strategy, the model can better perform by applying its learnt features to the current task.

By using pretrained models such as DenseNet-169, Inception-v3, Alexnet-v3, VGG-16, and ResNet-18, we leveraged the knowledge and representations learned from large-scale image datasets. In the described implementation, the pretrained models are initialised with their weights and then fine-tuned by freezing all layers except the last one. Freezing the initial layers ensures that the models retain their learned representations and prevents them from being modified during training, while the last layer is trained specifically for the COVID-19 detection task. We hoped to fine-tune the models' last layer to better suit our goal of detecting COVID-19 in CXRs by replacing it with a different configuration.

4.4 Evaluation and Visualization:

The model's state dictionary is loaded from the specified path, and the model is initialized. The test dataset is loaded using a Data Loader object. The model is set to evaluation mode using the eval() method. Afterwards it generates predictions based on the test images as it cycles through the test loader. The predicted labels and true labels are collected, using the collected labels, the confusion matrix and classification report are computed utilizing scikit-learn's features using the NumPy library [17].

The confusion matrix heat map provides insights into the model's performance by revealing the occurrence of correct and incorrect predictions, enabling a comprehensive understanding of the model's accuracy and error rates. The classification report includes metrics such as precision, recall, F1-score, and support is printed to provide a detailed summary of the model's evaluation metrics. The performance of a classification model at various classification thresholds is graphically represented by a ROC curve, also known as the receiver operating characteristic curve. The loss and accuracy curves for training and validation are plotted against the epoch number to visualize the model's learning progress.

Saliency heat maps are used to identify regions in the CXR images where deep learning models focused the most during COVID-19 or pneumonia detection. Additionally, this provides insights into the process for making predictions of the model and specific areas that contributed significantly to the detection are marked with a circle. The lung masks act as the boundary of the lungs for the detection process and to calculate the area of infection.

Chapter 5

Implementation

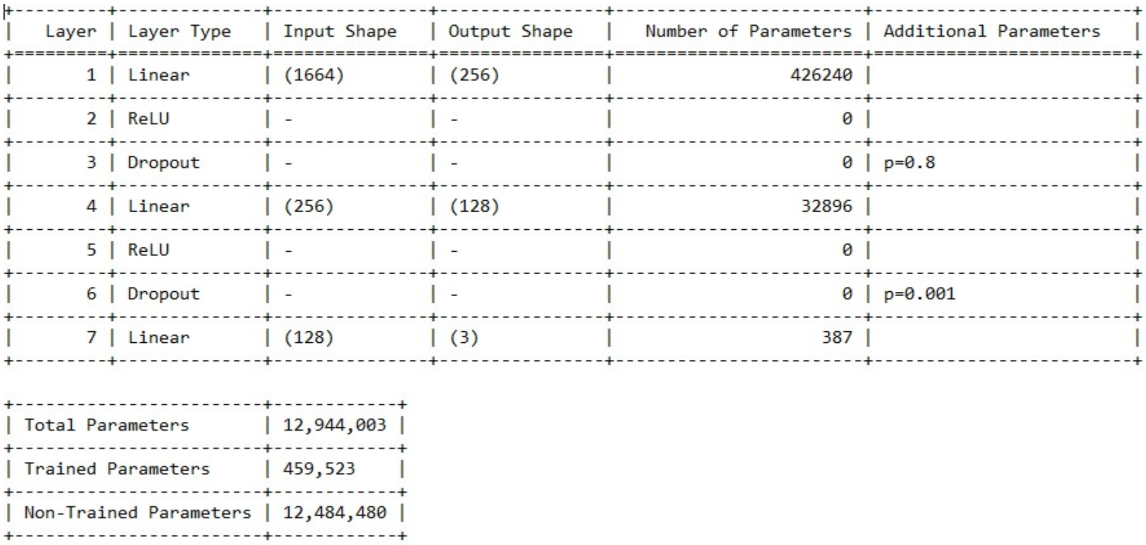
We have developed a CNN ensemble integrated with DenseNet169, VGG-16 (Visual Geometry Group 16), and ResNet-18 (Residual Network-18). The Implementation includes comparing the performance of the 5 pre-trained models and the ensemble model, to identify which model achieves the highest accuracy and effectiveness in detecting COVID-19 and Pneumonia from a dataset of CXRs consisting COVID-19 infected, Pneumonia (viral) and normal. Additionally, the model will also be able to generate a saliency map depicting areas of infection in the lung, with the percentage of lung area infected by utilizing the lung mask of the CXR image. During training, the CXR images and lung masks are fed into the models in the form of data-loader which are initialized from the torchvision library and loads the pretrained weights, and the network parameters are fine-tuned. Furthermore, to fine-tune the model, all layers except the last one are frozen to prevent their weights from being updated during training. The chosen loss function for optimization is cross-entropy, paired with the utilization of the Adam optimizer with a learning rate of xe-5, where 'x' is the multiplier for the specific model. Lastly the models learn to distinguish between COVID and normal images and adjust their weights accordingly. This process of training multiple complex deep learning models which involved processing numerous layers and operations taken from high-dimensional photogenic input data required us to use a GPU (Graphics Processing Unit). A GPU can handle computationally heavy processing of models with shorter overall training enabling us to evaluate the trained models faster and add adjustments in the last layer depending on its metrics. As shown in Table 4, these metrics aid in understanding the model's strengths and weaknesses and can help guide us to make further improvements or adjustments to the model. In the final stages of implementing deep learning CNN models for COVID-19 and pneumonia detection in CXR images, we implement explanation methods, such as saliency heat maps, that are utilized to provide insights into the models' predictions. These explanations help to highlight the regions of interest in the images that influenced the model's decision, enhancing interpret ability and maintaining credibility in the model's diagnostic capabilities.

Table 4: Model Performance

Model	Precision	Recall	F1-Score
Ensemble	0.95	0.95	0.95
VGG16	0.95	0.95	0.95
InceptionV3	0.95	0.95	0.95
ResNet-18	0.94	0.94	0.94
DenseNet-169	0.84	0.83	0.83
AlexNet	0.79	0.79	0.79

5.1 Hyper-parameter Optimization:

5.1.1 DenseNet169



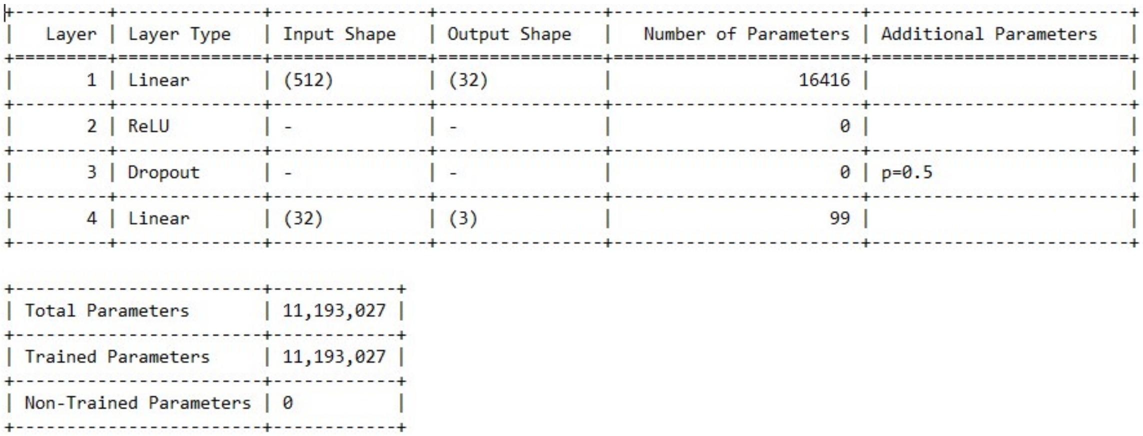
```
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|   Layer | Layer Type  | Input Shape  | Output Shape  |   Number of Parameters | Additional Parameters |
+=========+=============+==============+===============+========================+=======================+
|       1 | Linear      | (1664)       | (256)         |                 426240 |                       |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       2 | ReLU        | -            | -             |                      0 |                       |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       3 | Dropout     | -            | -             |                      0 | p=0.8                 |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       4 | Linear      | (256)        | (128)         |                  32896 |                       |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       5 | ReLU        | -            | -             |                      0 |                       |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       6 | Dropout     | -            | -             |                      0 | p=0.001               |
+---------+-------------+--------------+---------------+------------------------+-----------------------+
|       7 | Linear      | (128)        | (3)           |                    387 |                       |
+---------+-------------+--------------+---------------+------------------------+-----------------------+

+------------------------+------------+
| Total Parameters       | 12,944,003 |
+------------------------+------------+
| Trained Parameters     | 459,523    |
+------------------------+------------+
| Non-Trained Parameters | 12,484,480 |
+------------------------+------------+
```

Figure 8: Layer Summary of DenseNet169 Model

In Figure 8, we can see the modified layer decreases the input size by dividing the original number of features by a factor 4, employs ReLU activation for non-linearity, and employs dropout with a probability of 0.8 to prevent overfitting. Another fully connected layer with 128 units is added, with regularization and enhanced generalization achieved by ReLU (Rectified Linear Unit) activation and dropout with a probability of 0.001. The number of classes is represented by the final fully connected layer, which has two units: Covid and normal. The model is trained utilizing the Adam optimization algorithm and employs the Cross-Entropy Loss function, with a learning rate of 0.05e-5. Finally, we were able to acquire the 271,505 trainable parameters that the model utilized to train.

This combined strategy classified COVID-19 on CXRs with an accuracy of 0.9014 and a data loss of 0.2369 for this method shown in Figure 13.

5.1.2 Reznet18



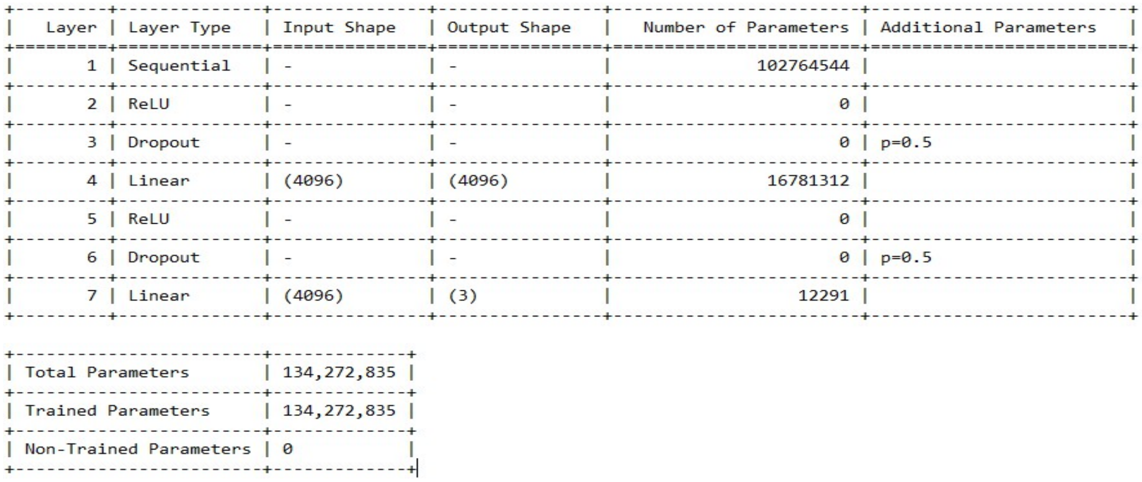
```
+---------+--------------+----------------+----------------+------------------------+------------------------+
|   Layer | Layer Type   | Input Shape    | Output Shape   |   Number of Parameters | Additional Parameters  |
+=========+==============+================+================+========================+========================+
|       1 | Linear       | (512)          | (32)           |                  16416 |                        |
+---------+--------------+----------------+----------------+------------------------+------------------------+
|       2 | ReLU         | -              | -              |                      0 |                        |
+---------+--------------+----------------+----------------+------------------------+------------------------+
|       3 | Dropout      | -              | -              |                      0 | p=0.5                  |
+---------+--------------+----------------+----------------+------------------------+------------------------+
|       4 | Linear       | (32)           | (3)            |                     99 |                        |
+---------+--------------+----------------+----------------+------------------------+------------------------+

+------------------------+------------+
| Total Parameters       | 11,193,027 |
+------------------------+------------+
| Trained Parameters     | 11,193,027 |
+------------------------+------------+
| Non-Trained Parameters | 0          |
+------------------------+------------+
```

Figure 9: Layer Summary of Reznet18 Model

In Figure 9, we can see the modification included reducing the amount of hidden units in the linear layer from the original number of feature to 32. Subsequently, a Rectified Linear Unit (ReLU) activation function was applied to introduce non-linearity, enhancing the model's representational capacity. To prevent overfitting, the dropout rate was set to 0.5 using a dropout layer. Lastly, the final layer of the model, responsible for classification, was retained with 2 units corresponding to our classification needs. The model is trained utilizing the Adam optimization algorithm and employs the Cross-Entropy Loss function, with a learning rate of 0.25e-5. Finally, we were able to acquire the 11,192,994 trainable parameters that the model utilized to train.

This combined strategy classified COVID-19 on CXRs with an accuracy of 0.9755 and a data loss of 0.0766 for this method shown in Figure 16.

5.1.3 Vgg16



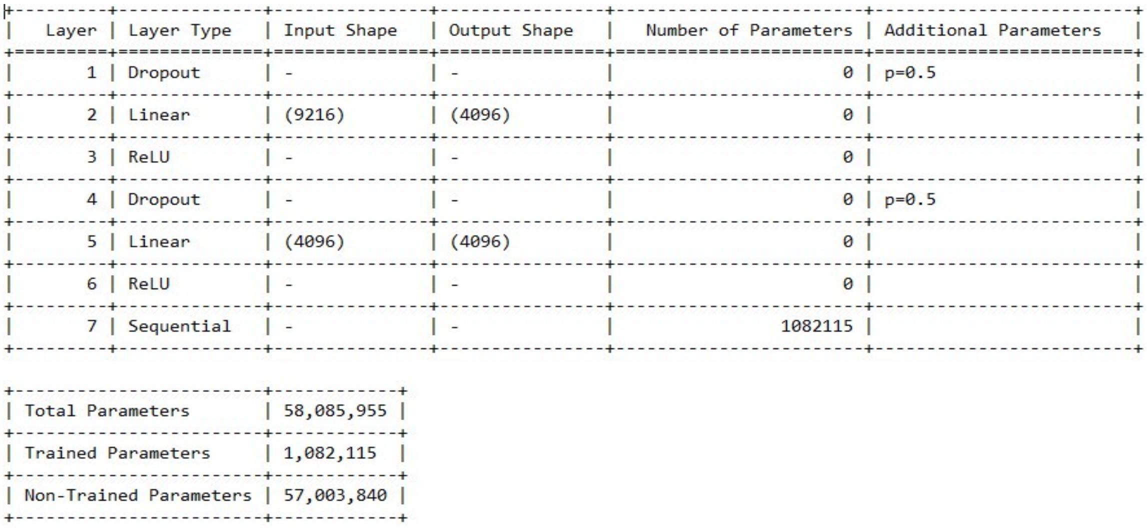
```
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|  Layer | Layer Type   | Input Shape    | Output Shape   |   Number of Parameters | Additional Parameters   |
+========+==============+================+================+==========================+=========================+
|      1 | Sequential   | -              | -              |                102764544 |                         |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      2 | ReLU         | -              | -              |                        0 |                         |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      3 | Dropout      | -              | -              |                        0 | p=0.5                   |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      4 | Linear       | (4096)         | (4096)         |                 16781312 |                         |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      5 | ReLU         | -              | -              |                        0 |                         |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      6 | Dropout      | -              | -              |                        0 | p=0.5                   |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+
|      7 | Linear       | (4096)         | (3)            |                    12291 |                         |
+--------+--------------+----------------+----------------+--------------------------+-------------------------+

+------------------------+-------------+
| Total Parameters       | 134,272,835 |
+------------------------+-------------+
| Trained Parameters     | 134,272,835 |
+------------------------+-------------+
| Non-Trained Parameters | 0           |
+------------------------+-------------+
```

Figure 10: Layer Summary of Vgg16 Model

In Figure 10, we can see the modifications are made to the final completely connected layer to have 2 output features, adapting the model for a binary classification task. After the first completely is connected layer, a dropout layer with a probability of 0.8 is added, improving generalization by reducing overfitting. Another dropout layer with a probability of 0.2 is added after the second fully connected layer, further enhancing regularization and preventing overfitting. The model is trained utilizing the Adam optimization algorithm and employs the Cross-Entropy Loss function, with a learning rate of 0.15e-5. Finally, we were able to acquire the 134,268,738 trainable parameters that the model utilized to train.

This combined strategy classified COVID-19 on CXRs with an accuracy of 0.9819 and a data loss of 0.0481 for this method shown in Figure 19.

5.1.4 AlexNet



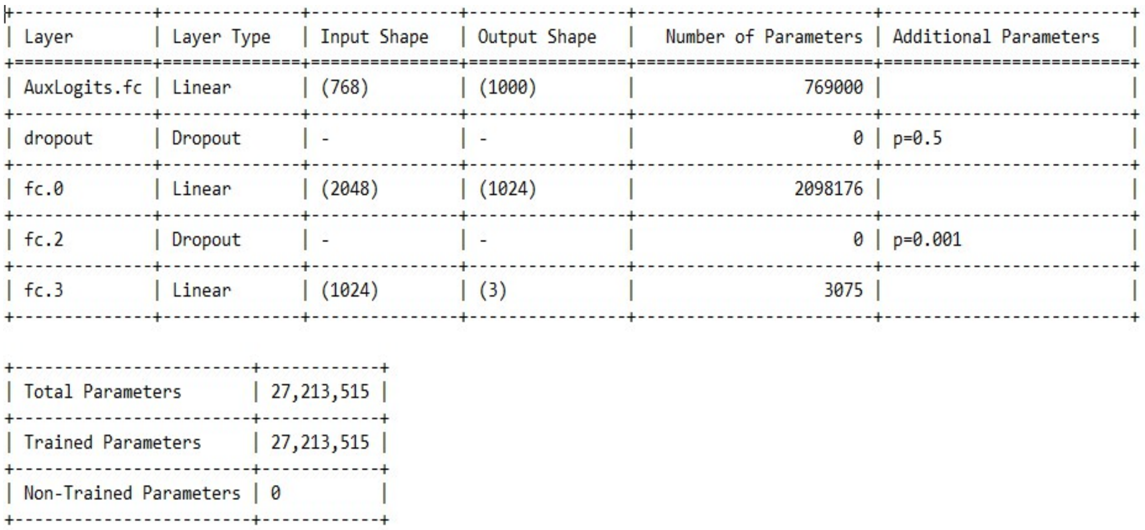
```
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|  Layer | Layer Type   | Input Shape  | Output Shape  |  Number of Parameters | Additional Parameters  |
+========+==============+==============+===============+=======================+========================+
|      1 | Dropout      | -            | -             |                     0 | p=0.5                  |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      2 | Linear       | (9216)       | (4096)        |                     0 |                        |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      3 | ReLU         | -            | -             |                     0 |                        |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      4 | Dropout      | -            | -             |                     0 | p=0.5                  |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      5 | Linear       | (4096)       | (4096)        |                     0 |                        |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      6 | ReLU         | -            | -             |                     0 |                        |
+--------+--------------+--------------+---------------+-----------------------+------------------------+
|      7 | Sequential   | -            | -             |               1082115 |                        |
+--------+--------------+--------------+---------------+-----------------------+------------------------+

+-----------------------+------------+
| Total Parameters      | 58,085,955 |
+-----------------------+------------+
| Trained Parameters    | 1,082,115  |
+-----------------------+------------+
| Non-Trained Parameters | 57,003,840 |
+-----------------------+------------+
```

Figure 11: Layer Summary of AlexNet Model

In Figure 11, Alexnet architecture starts with a linear layer with 256 output features, followed by a ReLU activation function to introduce non-linearity. A dropout layer with a probability of 0.8 is added to prevent overfitting during training. Next, another linear layer with 128 output features is added, followed by another ReLU activation. Finally, we replaced the last linear layer with one that has 3 output features, corresponding to the number of classes in the task we're working on. The model is trained utilizing the Adam optimization algorithm and employs the Cross-Entropy Loss function, with a learning rate of 0.15e-5. Finally, we were able to acquire the 1,002,1 trainable parameters that the model utilized to train.

This combined strategy classified COVID-19 on CXRs with an accuracy of 0.9714 and a data loss of 0.1425 for this method shown in Figure 22.

5.1.5 Inception V3



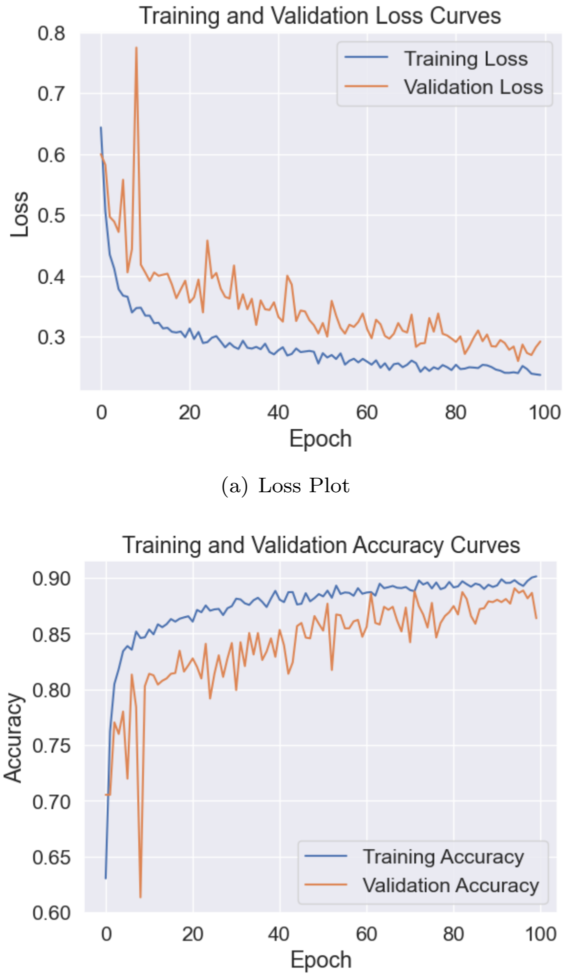
```
+-------------+-------------+--------------+---------------+-----------------------+------------------------+
| Layer       | Layer Type  | Input Shape  | Output Shape  |  Number of Parameters | Additional Parameters  |
+=============+=============+==============+===============+=======================+========================+
| AuxLogits.fc| Linear      | (768)        | (1000)        |                769000 |                        |
+-------------+-------------+--------------+---------------+-----------------------+------------------------+
| dropout     | Dropout     | -            | -             |                     0 | p=0.5                  |
+-------------+-------------+--------------+---------------+-----------------------+------------------------+
| fc.0        | Linear      | (2048)       | (1024)        |               2098176 |                        |
+-------------+-------------+--------------+---------------+-----------------------+------------------------+
| fc.2        | Dropout     | -            | -             |                     0 | p=0.001                |
+-------------+-------------+--------------+---------------+-----------------------+------------------------+
| fc.3        | Linear      | (1024)       | (3)           |                  3075 |                        |
+-------------+-------------+--------------+---------------+-----------------------+------------------------+

+-----------------------+------------+
| Total Parameters      | 27,213,515 |
+-----------------------+------------+
| Trained Parameters    | 27,213,515 |
+-----------------------+------------+
| Non-Trained Parameters | 0         |
+-----------------------+------------+
```

Figure 12: Layer Summary of Inception Model

In Figure 12, we can see the modifications are made to the final completely connected layer to have 3 output features, adapting the model for multi-classification task. In our customization, we modified the fully connected layer to have half the number of hidden units and changed the output size to 3 for our specific classification task, effectively reducing the model's complexity. Additionally, we removed the final softmax activation from the auxiliary classifiers. The model is trained utilizing the Adam optimization algorithm and employs the Cross-Entropy Loss function, with a learning rate of 1e-5. Finally, we were able to acquire the 27,213,515 trainable parameters that the model utilized to train.

This combined strategy classified COVID-19 on CXRs with an accuracy of 0.9819 and a data loss of 0.0481 for this method shown in Figure 25.

5.2 Deep Learning Model Summary



(a) Loss Plot

(b) Accuracy Plot

Figure 13: DenseNet169 Model

5.2.1 DenseNet-169

The metrics shown in Figure 14, suggest that out of 2257 samples labeled as ”normal,” the model accurately predicted 1993 (true negatives) while misclassifying 81 as ”Covid-19” (false positives) and 121 as ”pneumonia” (false positives). For the 2325 samples labeled as ”Covid-19,” the model correctly predicted 1758 (true positives) but misclassified 345 as ”normal” (false negatives) and 169 as ”pneumonia” (false positives). Lastly, out of 2204 samples labeled as ”pneumonia,” the model correctly predicted 1902 (true negatives) and misclassified 292 as ”normal” (false negatives) and

127 as ”Covid-19” (false negatives). The classification report from Table 5indicates that the model achieved precision (0.84 for ”covid19”, 0.89 for ”pneumonia” and 0.87 for ”normal”).

Class	Precision	Recall	F1-Score	Support
Normal	0.84	0.90	0.87	2259
COVID-19	0.89	0.86	0.85	2278
Pneumonia	0.87	0.83	0.88	2251
Accuracy			0.82	6788
Macro Avg	0.85	0.84	0.85	6788
Weighted Avg	0.85	0.84	0.85	6788

Table 5: DenseNet-169 Classification Report

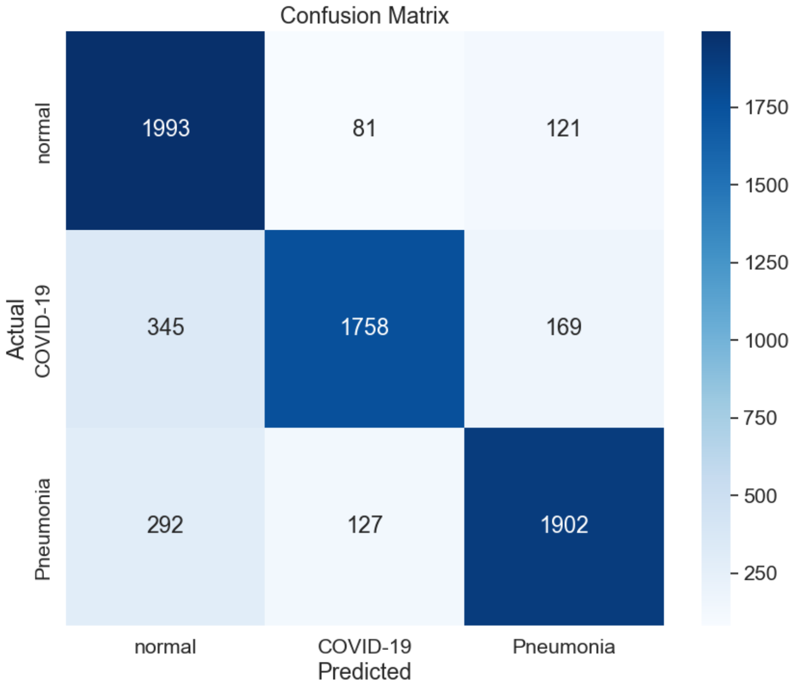


Figure 14: Confusion Matrix of DenseNet169 Model

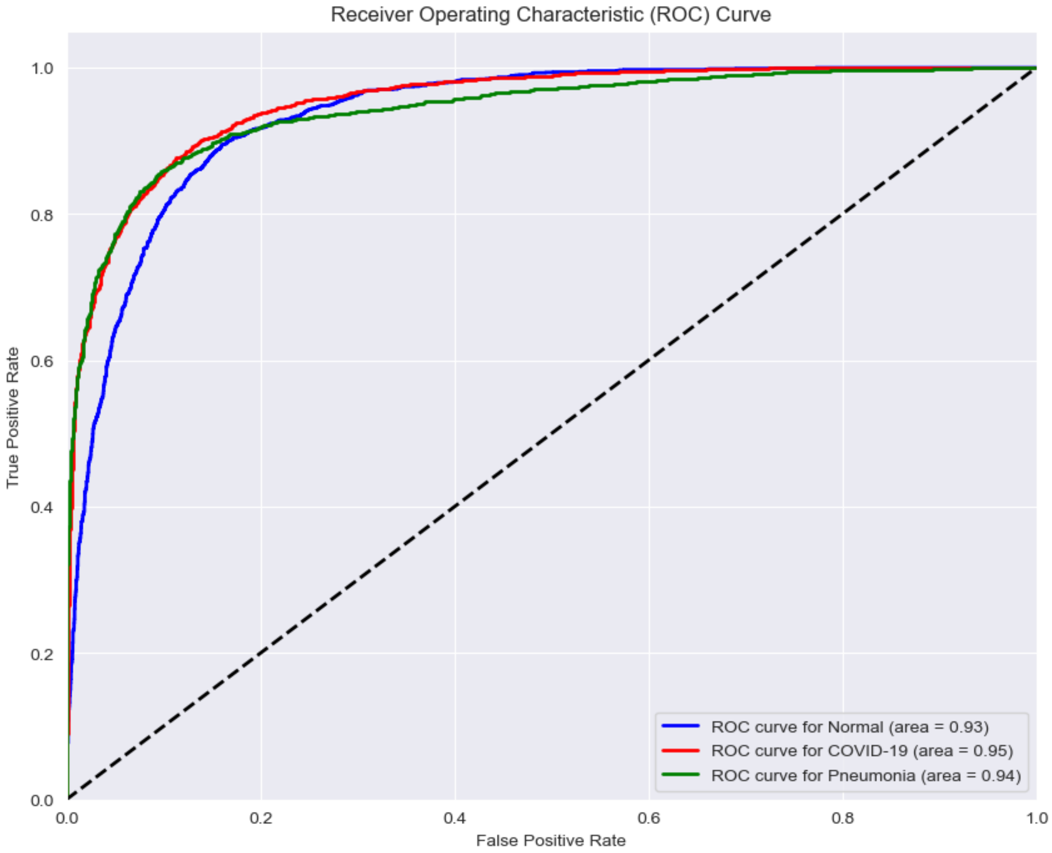
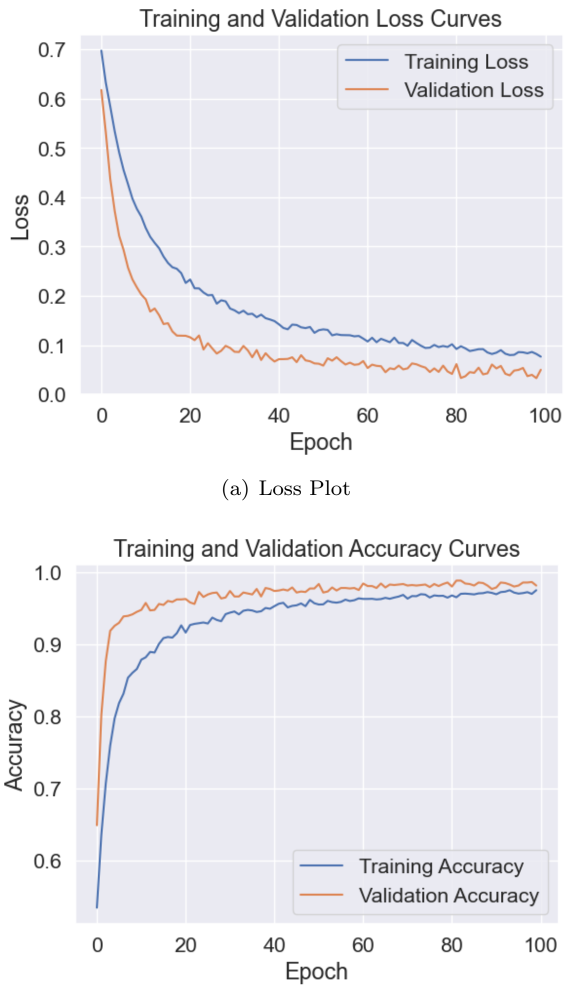


Figure 15: ROC of DenseNet169 Model



(a) Loss Plot

(b) Accuracy Plot

Figure 16: Reznet18 Model

5.2.2 ResNet-18

The metrics shown in Figure 17 suggest that out of 2257 samples labeled as ”normal,” the model accurately predicted 2047 (true negatives) while misclassifying 35 as ”Covid-19” (false positives) and 78 as ”pneumonia” (false positives). For the 2325 samples labeled as ”Covid-19,” the model correctly predicted 2180 (true positives) but misclassified 76 as ”normal” (false negatives) and 41 as ”pneumonia” (false positives). Lastly, out of 2204 samples labeled as ”pneumonia,” the model correctly predicted 2140 (true negatives) and misclassified 150 as ”normal” (false negatives) and 41 as ”Covid-19” (false negatives). The classification report from Table 6 indicates that the model achieved high precision (0.86 for ”covid19”, 0.95 for ”pneumonia” and 0.93 for ”normal”).

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Normal	0.86	0.93	0.90	2259
COVID-19	0.95	0.95	0.95	2278
Pneumonia	0.93	0.86	0.90	2251
Accuracy			0.92	6788
Macro Avg	0.92	0.92	0.92	6788
Weighted Avg	0.92	0.92	0.92	6788

Table 6: ResNet-18 Classification Report

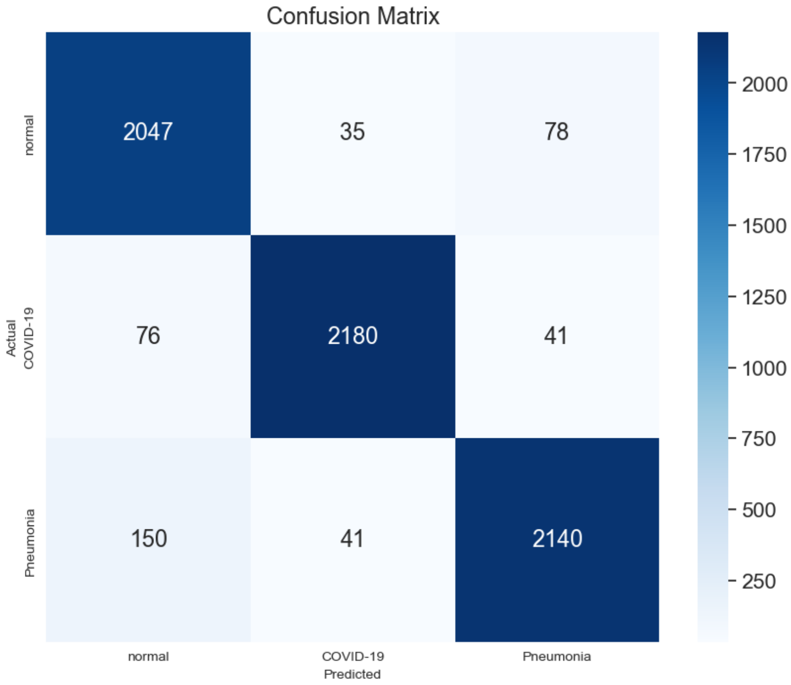


Figure 17: Confusion Matrix of ResNet18 Model

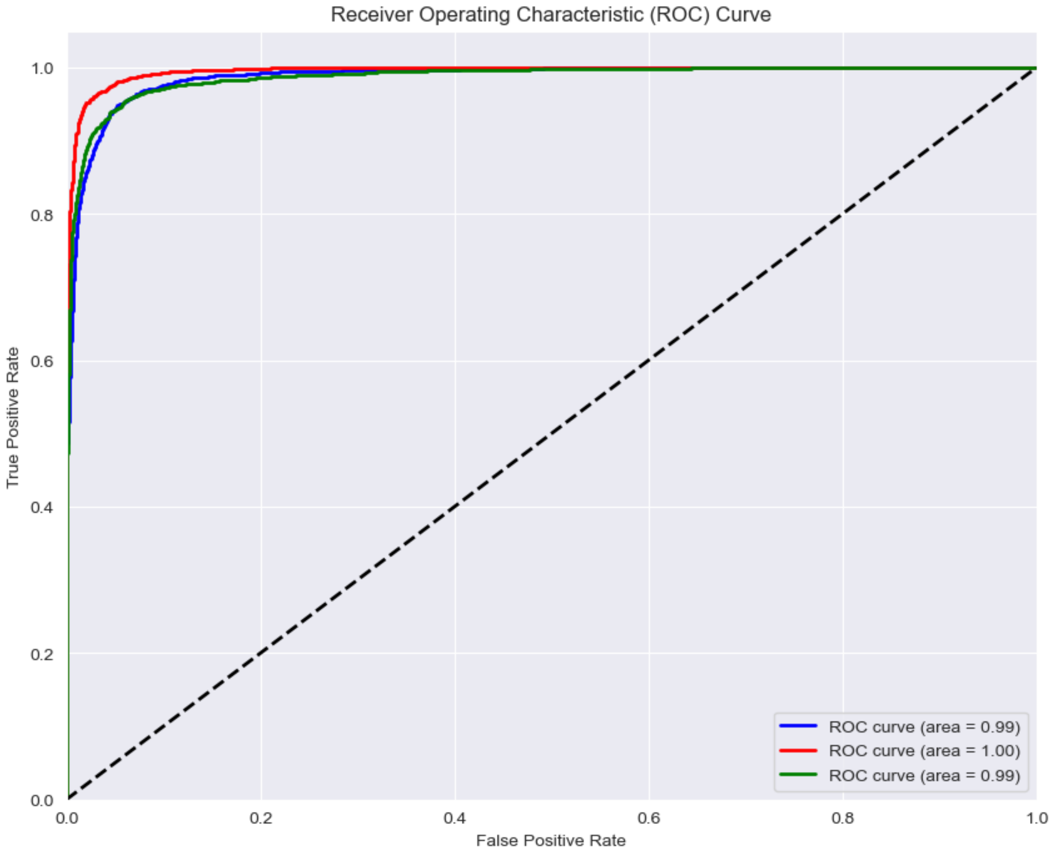
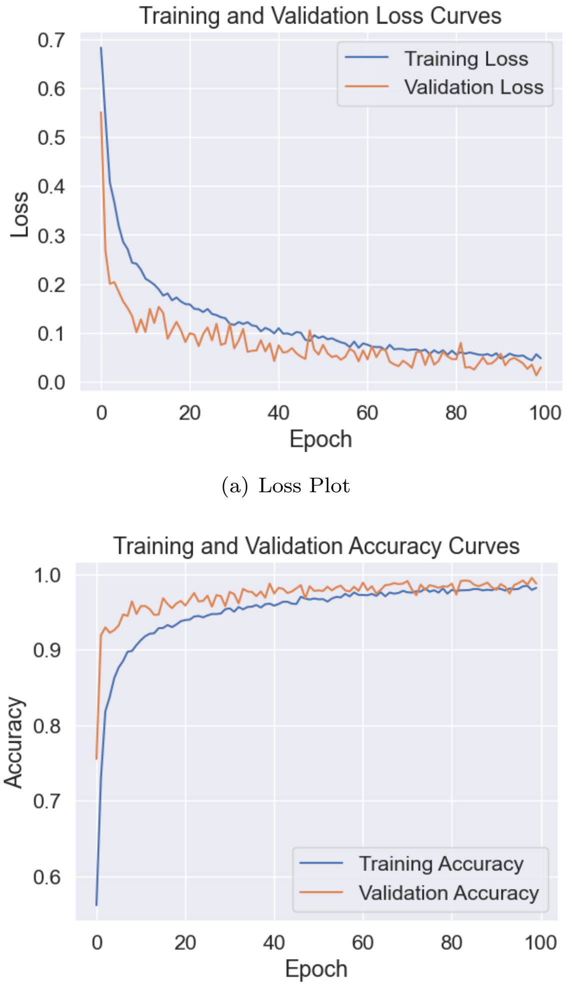


Figure 18: ROC of ResNet18 Model



(a) Loss Plot

(b) Accuracy Plot

Figure 19: Vgg16 Model

5.2.3 VGG-16

The metrics shown in Figure 20 suggest that out of 2257 samples labeled as ”normal,” the model accurately predicted 2085 (true negatives) while misclassifying 16 as ”Covid-19” (false positives) and 169 as ”pneumonia” (false positives). For the 2325 samples labeled as ”Covid-19,” the model correctly predicted 2119 (true positives) but misclassified 36 as ”normal” (false negatives) and 53 as ”pneumonia” (false positives). Lastly, out of 2204 samples labeled as ”pneumonia,” the model correctly predicted 2211 (true negatives) and misclassified 82 as ”normal” (false negatives) and 17 as ”Covid-19” (false negatives). The classification report from Table 7 indicates that the model achieved high precision (0.92 for ”covid19”, 0.98 for ”pneumonia” and 0.89 for ”normal”).

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Normal	0.92	0.89	0.91	2259
COVID-19	0.98	0.96	0.97	2278
Pneumonia	0.89	0.94	0.91	2251
Accuracy			0.93	6788
Macro Avg	0.93	0.93	0.93	6788
Weighted Avg	0.93	0.93	0.93	6788

Table 7: VGG-16 Classification Report

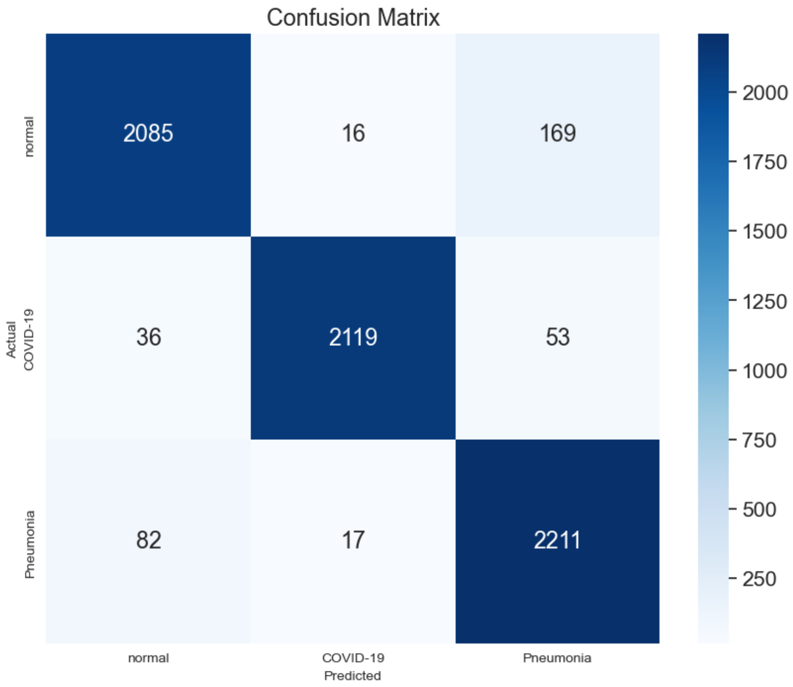


Figure 20: Confusion Matrix of VGG16 Model

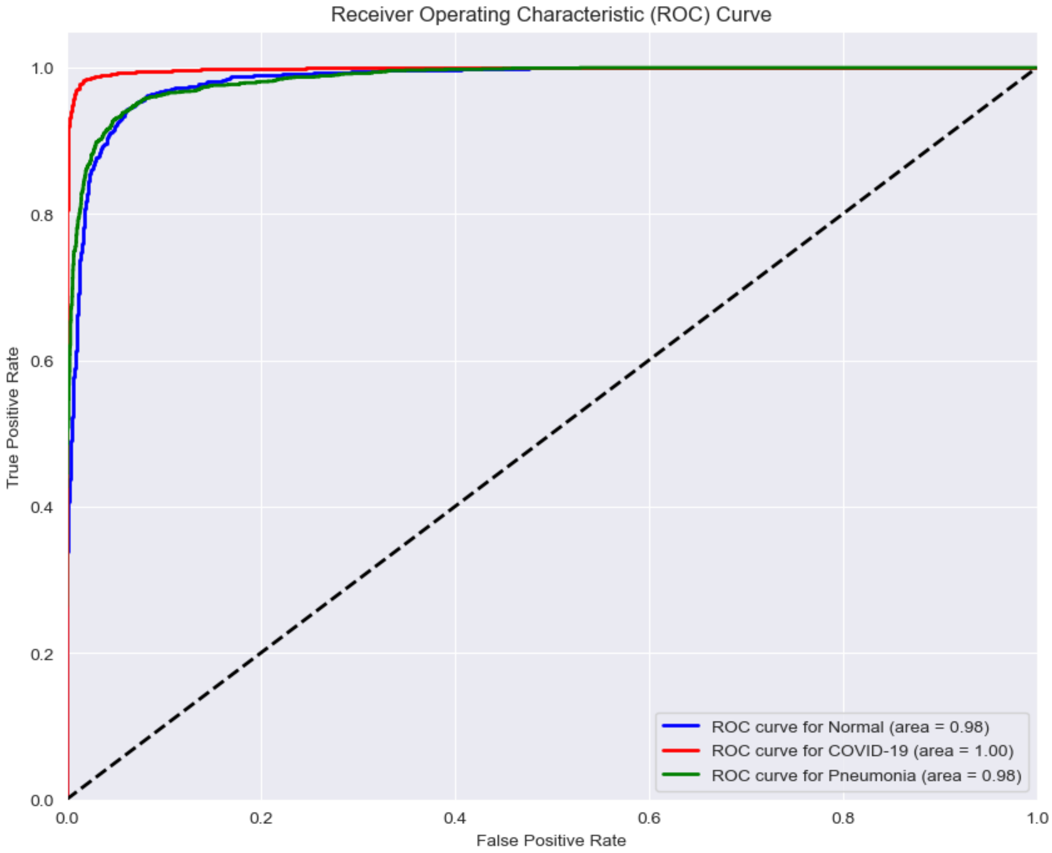
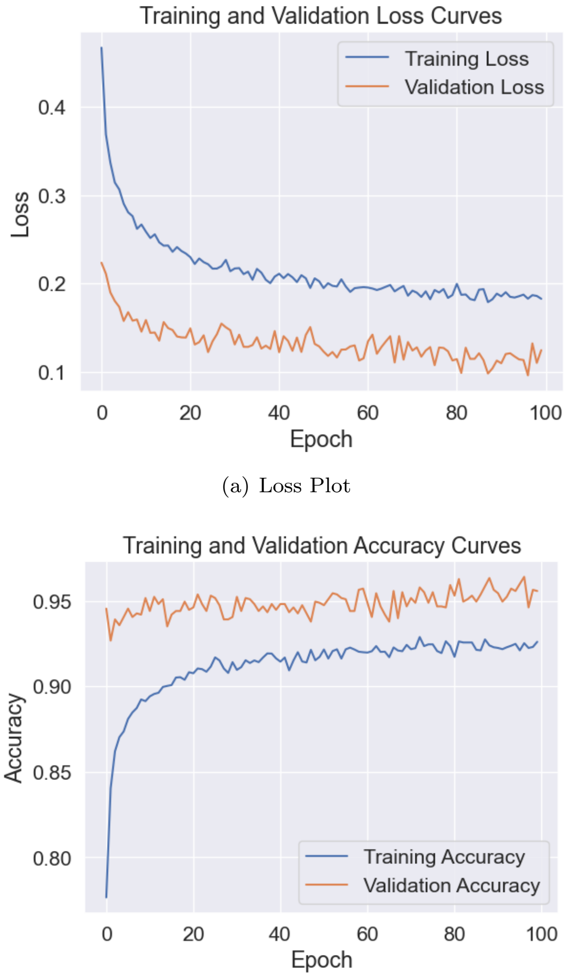


Figure 21: ROC of VGG16 Model



(a) Loss Plot

(b) Accuracy Plot

Figure 22: AlexNet Model

5.2.4 AlexNet

The metrics shown in Figure 23 suggest that out of 2257 samples labeled as ”normal,” the model accurately predicted 1933 (true negatives) while misclassifying 130 as ”Covid-19” (false positives) and 231 as ”pneumonia” (false positives). For the 2325 samples labeled as ”Covid-19,” the model correctly predicted 1583 (true positives) but misclassified 381 as ”normal” (false negatives) and 193 as ”pneumonia” (false positives). Lastly, out of 2204 samples labeled as ”pneumonia,” the model correctly predicted 1727 (true negatives) and misclassified 272 as ”normal” (false negatives) and 238 as ”Covid-19” (false negatives). The classification report from Table 8 indicates that the model achieved precision (0.72 for ”covid19”, 0.81 for ”pneumonia” and 0.84 for ”normal”).

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Normal	0.72	0.84	0.77	2259
COVID-19	0.81	0.75	0.78	2278
Pneumonia	0.84	0.77	0.80	2251
Accuracy			0.78	6788
Macro Avg	0.79	0.78	0.79	6788
Weighted Avg	0.79	0.78	0.79	6788

Table 8: AlexNet Classification Report

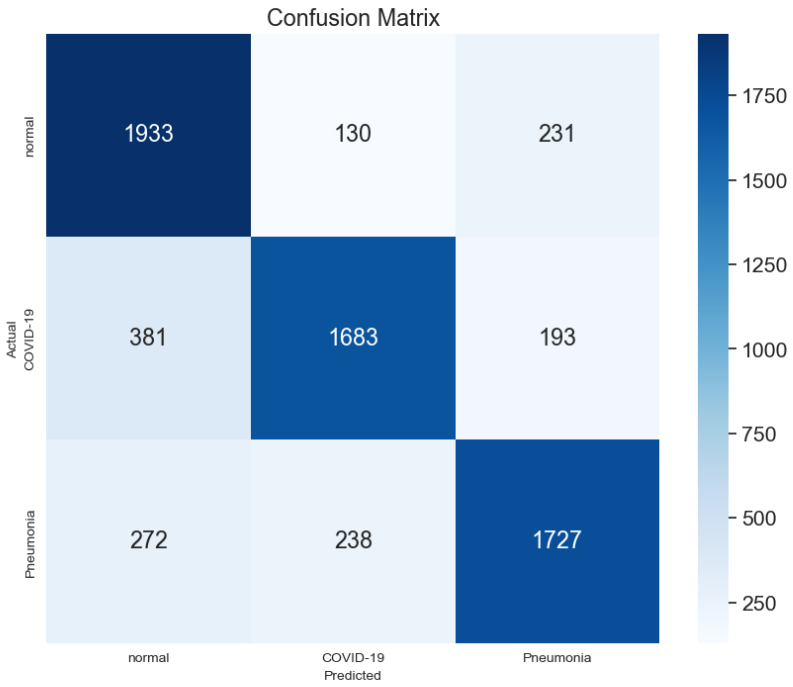


Figure 23: Confusion Matrix of AlexNet Model

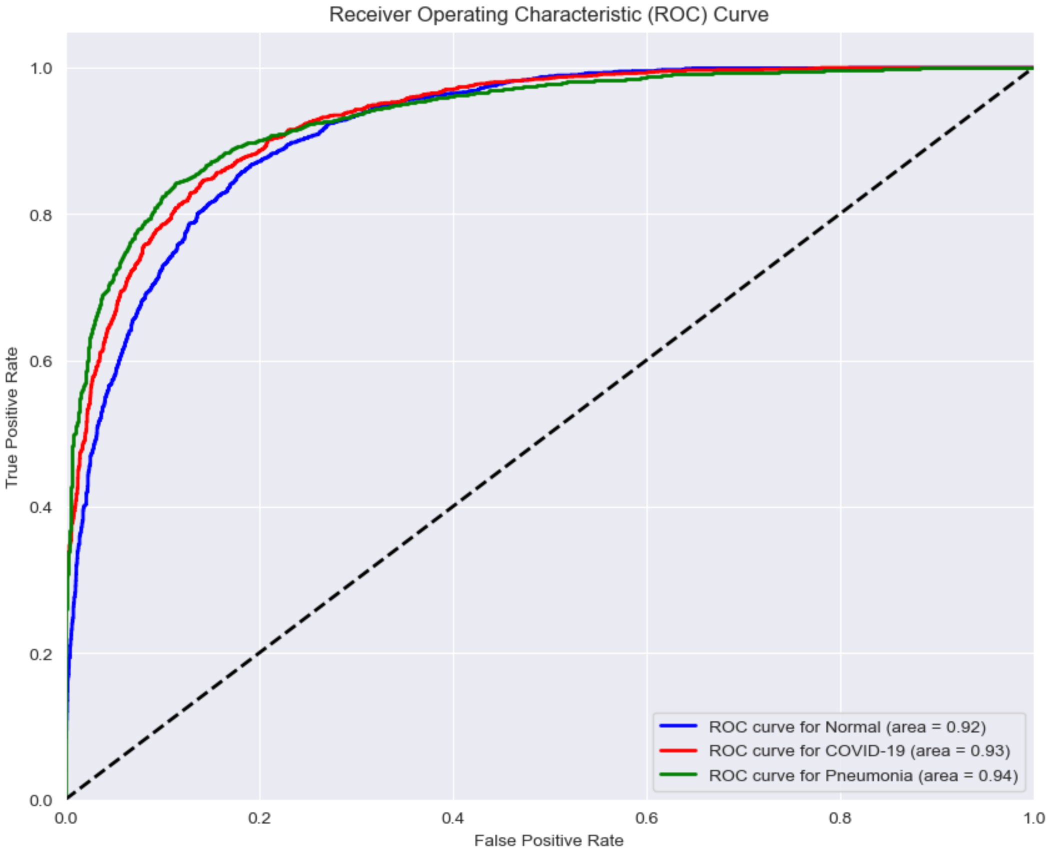
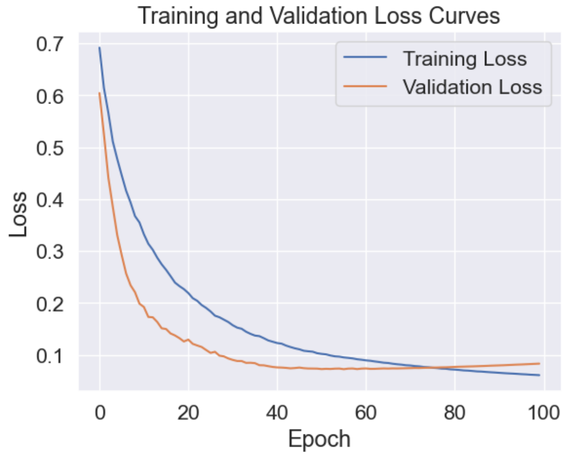
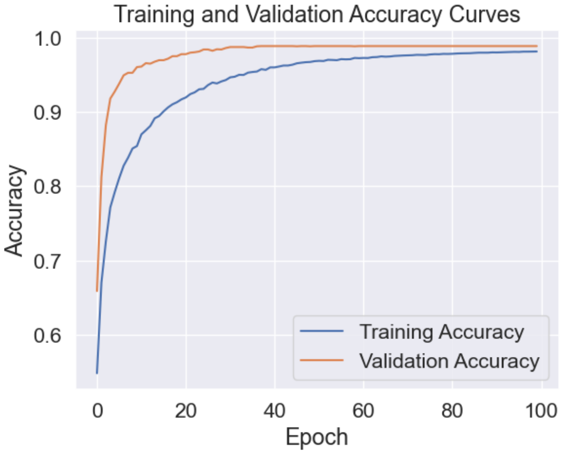


Figure 24: ROC of AlexNet Model



(a) Loss Plot



(b) Accuracy Plot

Figure 25: Inception Model

5.2.5 Inception

The metrics shown in Figure 26 suggest that out of 2257 samples labeled as "normal," the model accurately predicted 2079 (true negatives) while misclassifying 21 as "Covid-19" (false positives) and 157 as "pneumonia" (false positives). For the 2325 samples labeled as "Covid-19," the model correctly predicted 2264 (true positives) but misclassified 45 as "normal" (false negatives) and 16 as "pneumonia" (false positives). Lastly, out of 2204 samples labeled as "pneumonia," the model correctly predicted 2090 (true negatives) and misclassified 78 as "normal" (false negatives) and 36 as "Covid-19" (false negatives). The classification report indicates from Table 9 that the model achieved high precision (0.97 for "covid19", 0.92 for "pneumonia" and 0.94 for "normal").

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Normal	0.92	0.95	0.90	2259
COVID-19	0.99	0.97	0.98	2278
Pneumonia	0.88	0.94	0.91	2251
Accuracy			0.93	6788
Macro Avg	0.93	0.93	0.93	6788
Weighted Avg	0.93	0.93	0.93	6788

Table 9: Inception V3 Classification Report

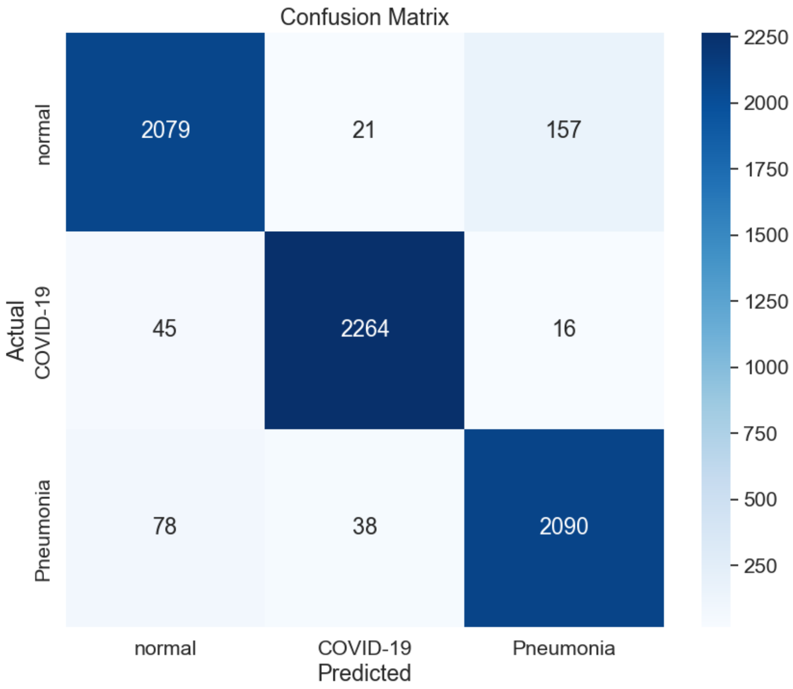


Figure 26: Confusion Matrix of Inception Model

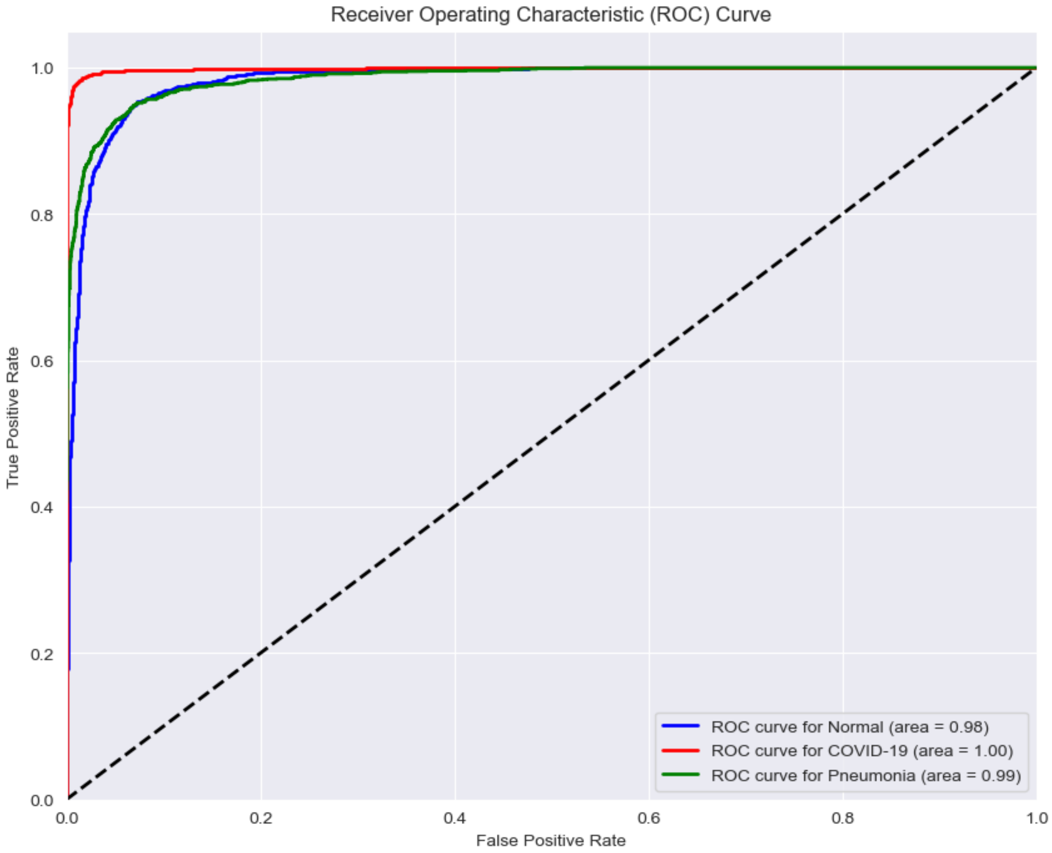


Figure 27: ROC of IncetionV3 Model

5.3 Ensemble Model Summary:

5.3.1 Explanation

Three pre-trained deep learning models, ResNet-18, VGG16, and Inception-v3, are combined to form a hybrid ensemble in this ensemble model's architecture. In order to provide the appropriate number of classes (num-classes =3) for categorizing COVID-19, viral pneumonia, and normal, each model is adjusted by swapping out the final completely connected layer. ResNet-18's final fully connected layer is replaced with nn.Linear(512, num-classes), VGG16's final classifier layer with nn.Linear(4096, num-classes), and Inception-v3's last fully connected layer with nn.Linear(2048, num-classes). The ensemble then takes the average of the output probabilities from the three models to produce the final prediction. The hyper parameters of the ensemble are set to use the CrossEntropyLoss as the loss function, Adam optimizer with a learning rate of 0.05e-5, and weight decay of 5e-5 for training the ensemble. Our understanding of model-level fusion in this ensemble architecture is that, as it combines complementary information and representations from different pre-trained models (ResNet-18, VGG16, and Inception-v3) to enhance the overall predictive power and increase the diversity of feature extraction, leading to improved classification performance as shown by the ensemble's results in Figure 28.

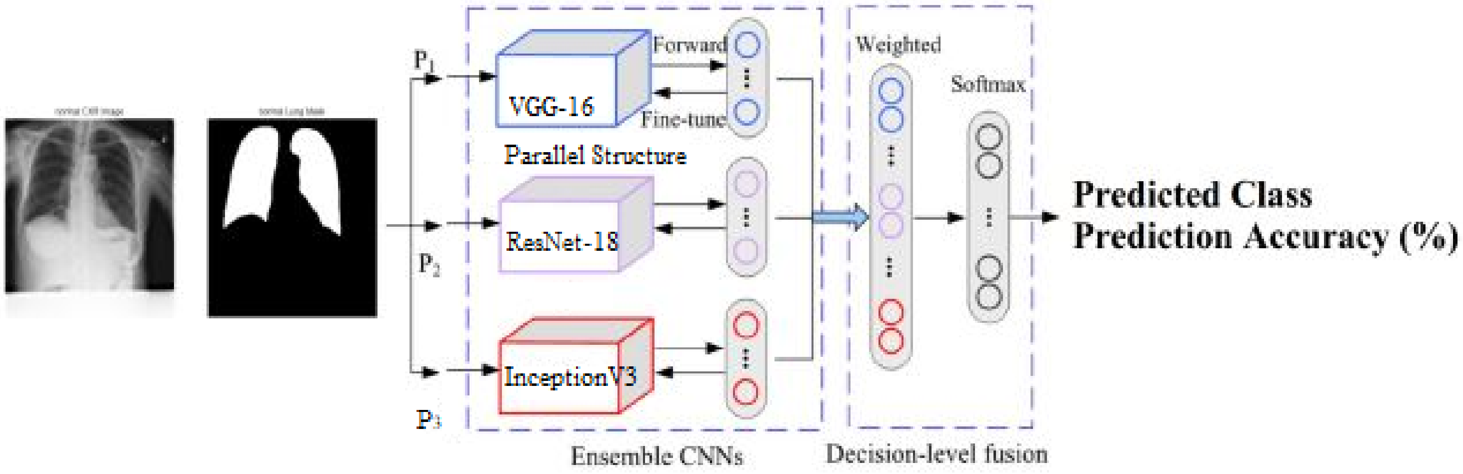
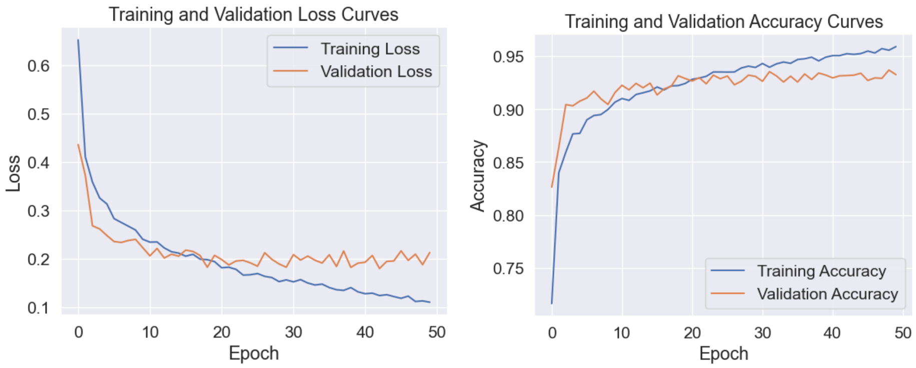


Figure 28: Ensemble Fusion Diagram

5.3.2 Ensemble Results

From Figure 29, 30 and 31 we can see the ensemble models exhibiting remarkable performance by combining diverse algorithms, yielding impressive metrics such as high accuracy and low loss during training. Moreover, their ability to consistently generalize well to unseen data is a testament to their robustness and effectiveness in generating accurate predictions.



(a) Loss Plot

(b) Accuracy Plot

Figure 29: Ensemble Model

Class	Precision	Recall	F1-Score	Support
Normal	0.96	0.90	0.93	2259
COVID-19	0.96	0.99	0.98	2278
Pneumonia	0.93	0.95	0.94	2251
Accuracy			0.95	6788
Macro Avg	0.95	0.95	0.95	6788
Weighted Avg	0.95	0.95	0.95	6788

Table 10: Ensemble Classification Report

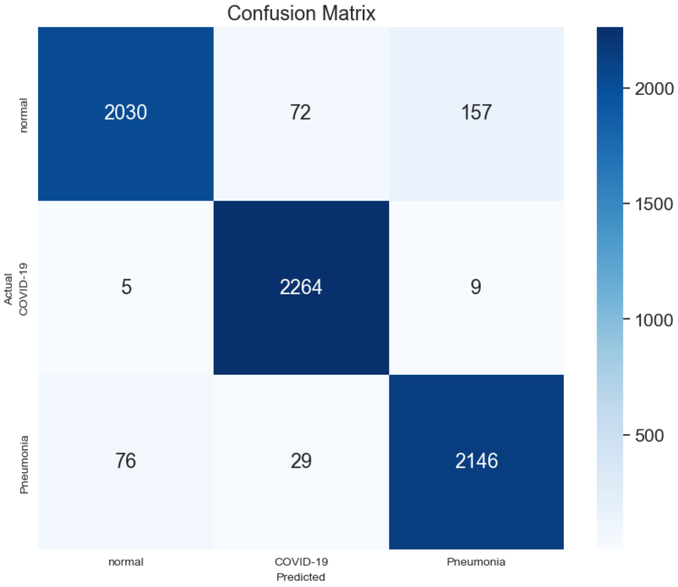


Figure 30: Confusion Matrix of Ensemble Model

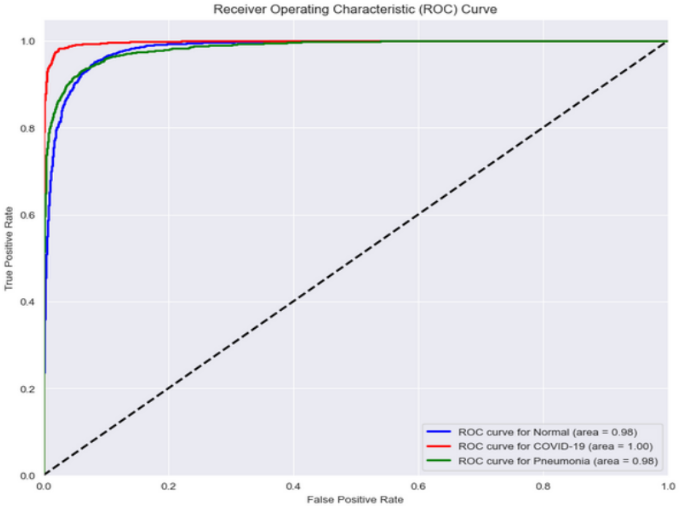


Figure 31: ROC of Ensemble Model

5.3.3 Comparison and Analysis

In this X-ray image saliency analysis algorithm, we first load and preprocess the X-ray image and the trained deep learning model. We then pass the preprocessed image through the model to obtain predictions. Additionally, we load and preprocess a lung mask to focus on specific regions of interest. Next, we generate a saliency map to highlight the areas of the image that have the most influence on the model's prediction. We normalize this saliency map to make it visually interpretable. Afterward, we apply the lung mask to the normalized saliency map to focus on relevant regions. To quantify the extent of infection, we calculate the percentage of infected lung tissue based on the masked saliency map. Finally, we display the original image, the saliency map with highlighted regions, and the calculated infection percentage to aid in the interpretation of the X-ray image.

Algorithm 1 X-ray Image Saliency Analysis

```
$image \gets \textsc{LoadImage}(image\_path)$
$preprocessed\_image \gets \textsc{PreprocessImage}(image, \text{Size: } (299, 299))$
$model \gets \textsc{LoadModel}(model\_path)$
$output \gets \textsc{ModelForwardPass}(model, preprocessed\_image)$
$predicted\_class, probability \gets \textsc{ProcessModelOutput}(output)$
$mask \gets \textsc{LoadMask}(mask\_path)$
$preprocessed\_mask \gets \textsc{PreprocessMask}(mask, \text{Size: } (299, 299))$
$saliency\_threshold \gets 0.6$
$saliency\_map \gets \textsc{GenerateSaliencyMap}(model, preprocessed\_image, predicted\_class, \text{Saliency Threshold: } saliency\_threshold)$
$normalized\_saliency\_map \gets \textsc{NormalizeSaliencyMap}(saliency\_map)$
$masked\_saliency\_map \gets \textsc{ApplyMask}(normalized\_saliency\_map, preprocessed\_mask)$
$circle\_threshold \gets 0.6$
$infected\_percentage \gets \textsc{CalculateInfectionPercentage}(masked\_saliency\_map, preprocessed\_mask, \text{Circle Threshold: } circle\_threshold)$
$\textsc{DisplayOriginalImage}(image)$
$\textsc{DisplaySaliencyMapWithCircles}(image, masked\_saliency\_map, circle\_threshold)$
$\textsc{DisplayInfectionPercentage}(infected\_percentage)$
```

As shown in Figure 32, the implementation of generating this explanation loads a CXR image, lung mask image and the ensemble model, passes the image through the model to predict its class and probability, and then generates a saliency map. The idea behind the ensemble is to leverage the diverse capabilities of these models and concatenate their predictions along 1 dimension to form a combined output tensor. This tensor represents the ensemble's final prediction, which incorporates the diverse features learned by the three base models. Moreover, saliency map highlights most influence on the ensemble model's prediction for the inputted image. The code applies a lung segmentation mask to the saliency map to focus on the lung area only. It calculates the percentage of the lung area infected and visualizes the original image and saliency map side by side. The saliency map is shown with a heatmap, darker or more intense areas in a saliency heat map indicate regions that had a stronger impact on the model's output and regions of high saliency are circled.

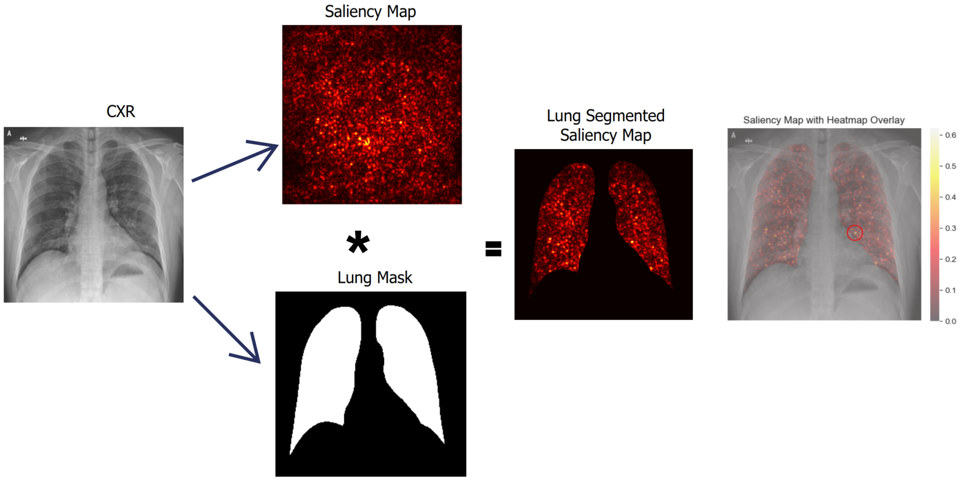

Saliency Map
CXR
*
Lung Mask
=
Lung Segmented Saliency Map
Saliency Map with Heatmap Overlay
0.6
0.5
0.4
0.3
0.2
0.1
0.0

Figure 32: Explanation through AI tools

The prediction accuracy of our ensemble model has shown to be high, with 96% which is higher than all the single models, shown in Figure 33. This further proves to us that the explanation of the ensemble's prediction comes from a robust and complementary fusion of the models Resnet-18, VGG-16 and Inception-V3. As shown in table 3, the ensemble's superior performance reflects the effective combination of diverse individual models' decision boundaries and features. Utilizing AI explanation tools on our ensemble model can provide insights into the specific contributions and patterns identified by each constituent model, enhancing our understanding of the complex decision-making process that leads to such high accuracy.

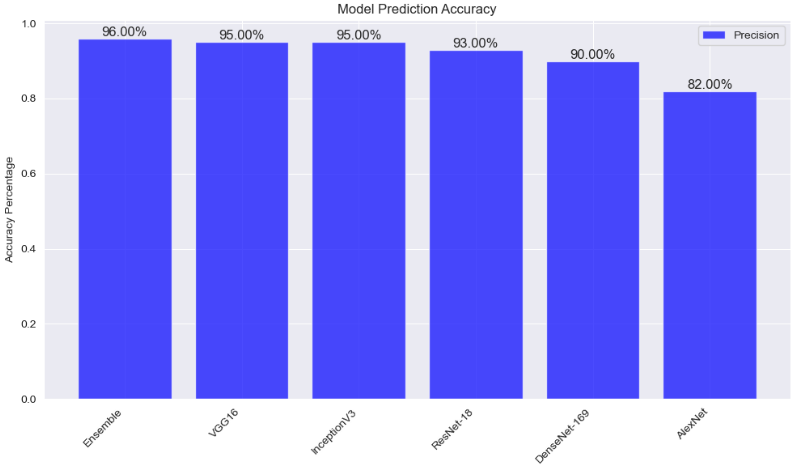


Figure 33: Overall Precision Accuraccy

Here in Figure 34, are three examples of explanations generated for the predictions of COVID-19 and viral pneumonia-infected CXR images from our ensemble model.

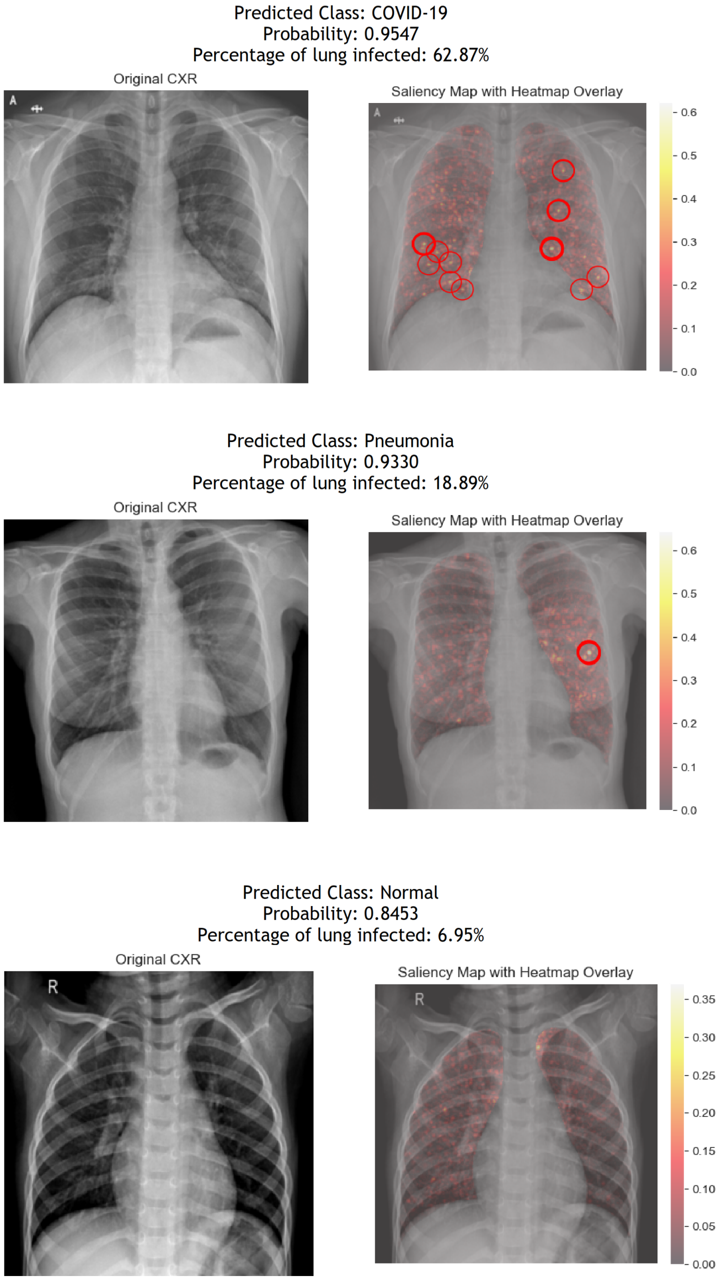
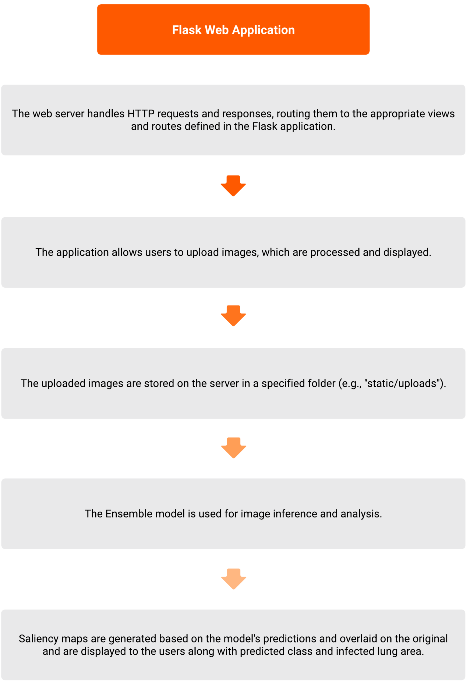

Predicted Class: COVID-19
Probability: 0.9547
Percentage of lung infected: 62.87%
Original CXR
A
Saliency Map with Heatmap Overlay
A
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Predicted Class: Pneumonia
Probability: 0.9330
Percentage of lung infected: 18.89%
Original CXR
Saliency Map with Heatmap Overlay
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Predicted Class: Normal
Probability: 0.8453
Percentage of lung infected: 6.95%
Original CXR
R
Saliency Map with Heatmap Overlay
R
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00

Figure 34: Prediction Explanation

Chapter 6

Website Deployment

The architecture of this web application utilizes Flask, a Python web framework, to establish a web server that manages HTTP requests and directs them to specific views and routes. Users are enabled to upload images, which are subsequently processed and presented. These uploaded images find storage on the server within a designated folder. The application incorporates the ensemble model, for conducting image inference and analysis tasks. The outcome of this analysis includes the generation of saliency maps, which are created based on the model's predictions and superimposed onto the original images. Finally, the processed images, complete with their saliency maps, predicted class and infected lung area are made accessible to users. From Figure 35, we can see the workflow of this implementation.



The web server handles HTTP requests and responses, routing them to the appropriate views and routes defined in the Flask application.

The application allows users to upload images, which are processed and displayed.

The uploaded images are stored on the server in a specified folder (e.g., "static/uploads").

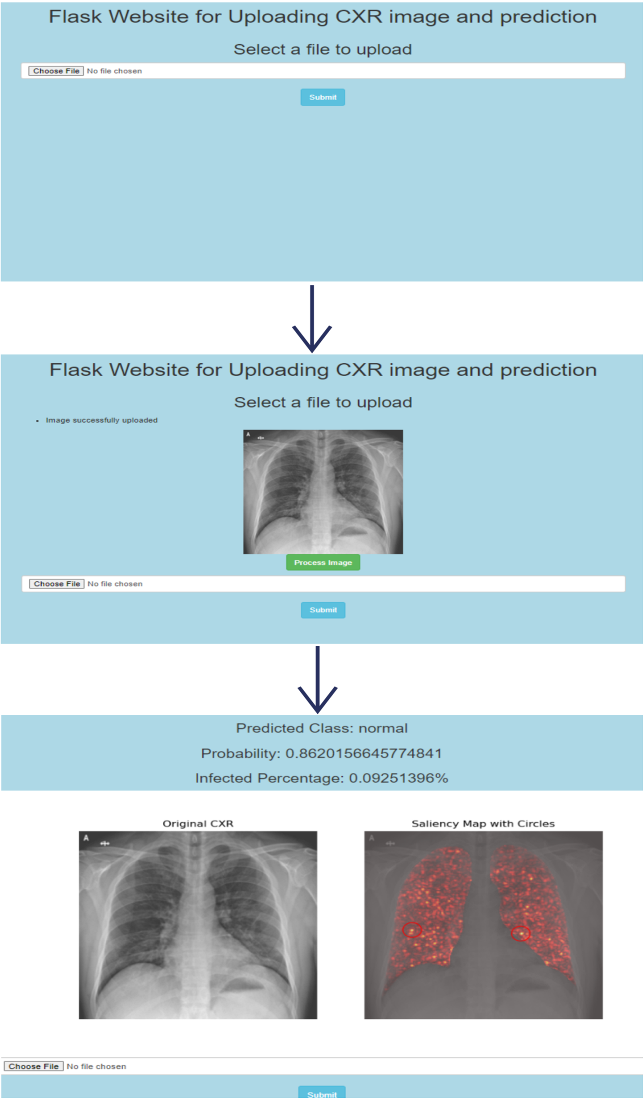
The Ensemble model is used for image inference and analysis.

Saliency maps are generated based on the model's predictions and overlaid on the original and are displayed to the users along with predicted class and infected lung area.

Figure 35: Workflow of Flask Website

6.1 Limitation

In most circumstances, it would be difficult to implement such a website in remote areas with lack of access to a high computational GPU. Future research can focus on utilizing high computational GPU edge devices and cloud services to improve the performance of the model and user experience, which would we very helpful during a Pandemic. The Figure 36 demonstrates the implementation of the image classification model through the website.



Flask Website for Uploading CXR image and prediction

Select a file to upload

Flask Website for Uploading CXR image and prediction

Select a file to upload

- Image successfully uploaded

Predicted Class: normal

Probability: 0.8620156645774841

Infected Percentage: 0.09251396%

Figure 36: Implementation of Flask Website

Chapter 7

Conclusion

7.1 Conclusion

Our research on these deep learning models have shown to have the capacity to predict whether a person is infected or not by analyzing their CXR report. Additionally, as shown in Figure 26 Inception-V3, ResNet-18 and VGG-16 have demonstrated be to detecting COVID-19 with a 94-98% accuracy. Moreover, both Inception-V3 and ResNet-18 shows similar performance, with outstanding evaluation metrics for both classes, as well as high overall accuracy. However, Inception-V3 obtains slightly better evaluation metrics for both classes compared to ResNet-18. VGG-16 has an overall prediction accuracy of 0.89, which is lower than the 0.95 accuracy achieved by both Inception-V3 and ResNet-18. Therefore, based on these metrics, we are confident that our ensemble model with its overall prediction accuracy of 96%, which combines our top 3 performing models achieved higher predictive accuracy and improved robustness by leveraging the diverse strengths of each model. This approach helps mitigate the risk of overfitting, enhances generalization, and provides more reliable predictions of COVID-19 and Viral Pneumonia. Lastly we attempt to provide better explanations for our ensemble model's prediction by generating a prediction accuracy score, a prediction mask and a saliency map. The predicted mask is further processed to create a saliency map which highlight the regions in an image that have the most influence on a machine learning model's prediction. This map is a visualization of the model's confidence in its predictions and provides information on how severe and where in the CXR image lung infections are present.

7.2 Future Work

Due of the vital necessity of healthcare decisions, the existing ensemble model may not match the demanding requirements for medical diagnosis. When considering real-world deployment in medical diagnosis, improving prediction accuracy is critical. Furthermore, addressing the dataset's limited variance, particularly for distinct strains of COVID-19 such as Omicron, Delta, Alpha, and so on, may improve the model's capacity to distinguish specific disease sub-types more correctly. To further improve the model, it might be beneficial to incorporate the expertise of medical professionals who possess extensive experience in diagnosing lung diseases from Chest X-rays. Integrating their insights during model development could significantly enhance its capabilities.

Bibliography

[1] World Health Organization (WHO), "COVID-19 Weekly Epidemiological Update Edition 143," Retrieved from https://reliefweb.int/report/world/covid-19-weekly-epidemiological -update-edition-143, 2023,

[2] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights into Imaging, vol. 9, no. 4, pp. 611-629, 2018, doi: 10.1007/s13244-018-0639-9.

[3] M. Mazurowski, M. Buda, A. Saha, and M. Bashir, "Deep learning in radiology: an overview of the concepts and a survey of the state of the art," arXiv.org, 2018, doi: 10.48550/arXiv.1802.08717

[4] M. E. H. Chowdhury et al., "Can AI Help in Screening Viral and COVID-19 Pneumonia?," in IEEE Access, vol. 8, pp. 132665-132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

[5] Juravle, G., Boudouraki, A., Terziyska, M., Rezlescu, C. . "Trust in artificial intelligence for medical diagnoses". Progress in brain research, 253, 263-282, 2020, doi: 10.1016/bs.pbr.2020.06.006

[6] K. M. Akran, "Covid BF.7 subvariant: What we know and what to do," The Daily Star, January 1, 2023, Retrieved from https://www.thedailystar.net/health/disease/coronavirus/ news/covid-bf7-subvariant-what-we-know-and-what-do-3210846.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[8] J. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," Physical and Engineering Sciences in Medicine, vol. 43, pp. 635–640, 2020, doi: 10.1007/s13246-020-00865-4.

[9] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998, doi: 10.1109/5.726791.

[10] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks Alex," Proc. 25th Int. Conf. Neural Inf. Process. Syst., vol. 1, pp. 1097-1105, 2012.

[11] G. Huang, Z. Liu, V. Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016, doi: 10.48550/arXiv.1608.06993

[12] He, K., Zhang, X., Ren, S., Sun, J. . "Deep residual learning for image recognition". In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016. doi: 10.48550/arXiv.1512.03385

[13] I. Razzak, S. Naz, A. Rehman, A. Khan and A. Zaib, "Improving Coronavirus (COVID-19) diagnosis using deep transfer learning," medRxiv, 2020, doi: 10.1101/2020.04.11.20054643

[14] N. Hilmizen, A. Bustamam and D. Sarwinda, "The multimodel deep learning for diagnosing COVID-19 pneumonia from chest CT-scan and X-ray images," Proc. 3rd Int. Seminar Res. Inf. Technol. Intell. Syst. (ISRITI), pp.26-31, 2020. doi: 10.1109/ISRITI51436.2020.9315478

[15] P. K. Sethy and S. K. Behera, "Detection of coronavirus disease (COVID-19) based on deep features", 2020. doi: 10.20944/preprints202003.0300.v1.

[16] A. Makris, et al "COVID-19 detection from chest X-Ray imagesusing Deep Learning and Convolutional Neural Networks", 2020. doi: 10.1145/3411408.3411416.

[17] C. R. Harris et al., "Array programming with NumPy," Nature, vol. 585, no. 7825, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.

[18] T. Rahman, A. Khandakar, Y. Qiblawey, A. Tahir, S. Kiranyaz, S. B. A. Kashem, M. T. Islam, S. Al Maadeed, S. M. Zughaier, M. S. Khan, and others, "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images," Computers in Biology and Medicine, vol. 132, p. 104319, 2021, doi: 10.1016/j.compbiomed.2021.104319

[19] J. X. Schraut, L. Liu, J. Gong et al., "A multi-output network with U-net enhanced class activation map and robust classification performance for medical imaging analysis," Discov Artif Intell 3, 1, 2023, doi: 10.1007/s44163-022-00045-1.

[20] T. Sanida, I-M. Tabakis, M.V. Sanida, A. Sideris, and M. Dasygenis, "A Robust Hybrid Deep Convolutional Neural Network for COVID-19 Disease Identification from Chest X-ray Images," Information, vol. 14, no. 6, pp. 310, 2023, doi: 10.3390/info14060310.

[21] D. Kuzinkovas and S. Clement, "The Detection of COVID-19 in Chest X-Rays Using Ensemble CNN Techniques," medRxiv, 2020 doi: 10.1101/2022.11.29.22282856.

[22] S. I. Nafisah, G. Muhammad, M. S. Hossain, and S. A. AlQahtani, "A Comparative Evaluation between Convolutional Neural Networks and Vision Transformers for COVID-19 Detection," Mathematics, vol. 11, no. 6, pp. 1489, 2023, doi: 10.3390/math11061489.

[23] B. Ibrokhimov and J-Y. Kang, "Deep Learning Model for COVID-19-Infected Pneumonia Diagnosis Using Chest Radiography Images," BioMedInformatics, vol. 2, no. 4, pp. 654-670, 2022, doi: 10.3390/biomedinformatics2040043.

[24] A. M. Tahir et al., "COVID-19 infection localization and severity grading from chest X-ray images," Computers in Biology and Medicine, vol. 139, 2021, doi: 10.1016/j.compbiomed.2021.105002.

[25] M. M. Nyasha, X. Zhu, S. Rusike, and E. Khondowe, "Automated Detection of Covid-19 from Chest X-ray Images Based on FlexMatch with ResNet50," International Research Journal of Advanced Engineering and Science, vol. 8, no. 1, pp. 293-298, 2023.

[26] M. Constantinou, T. Exarchos, A. G. Vrahatis, and P. Vlamos, "COVID-19 Classification on Chest X-ray Images Using Deep Learning Methods," International Journal of Environmental Research and Public Health, vol. 20, no. 3, pp. 2035, 2023, doi: 10.3390/ijerph20032035.

[27] World Health Organization (WHO), "Revised WHO classification and treatment of childhood pneumonia at health facilities", 2023.

[28] Szegedy, C., et al. "Going deeper with convolutions". Computer vision and pattern recognition (pp. 1-9), 2015. doi:10.48550/arXiv.1409.4842.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, ”Mobilenetv2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[30] M. Tan and Q. Le, ”Efficientnet: Rethinking model scaling for convolutional neural networks,” in International conference on machine learning, pp. 6105-6114, PMLR, doi: /10.48550/arXiv.1905.11946.

[31] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, ”SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5 MB model size”, 2019, doi: 10.48550/arXiv.1602.07360.

[32] F. Chollet, ”Xception: Deep learning with depthwise separable convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258, 2017, doi: /10.48550/arXiv.1610.02357.

[33] F. Demir, A. Sengur, and V. Bajaj, ”Convolutional neural networks based efficient approach for classification of lung diseases,” Health information science and systems, vol. 8, pp. 1-8, 2019, doi: /10.1007/s13755-019-0091-3.

[34] S. H. Karaddi and L. D. Sharma, ”Automated multi-class classification of lung diseases from CXR-images using pre-trained convolutional neural networks,” Expert Systems with Applications, vol. 211, pp. 118650, 2023, ISSN 0957-4174, doi: 10.1016/j.eswa.2022.118650.

[35] K. Kumar Ghosh, M. F. Ul Islam, A. A. Efaz, A. Chakrabarty and S. Hossain, ”Real-Time Mastitis Detection in Livestock using Deep Learning and Machine Learning Leveraging Edge Devices,” 2023 IEEE 17th International Symposium on Medical Information and Communication Technology (ISMICT), Lincoln, NE, USA, 2023, pp. 01-06, doi: 10.1109/ISMICT58261.2023.10152110.