A PROJECT REPORT

PNEUMONIA DETECTION USING TRANSFER LEARNING

Submitted by

AKASH R
(19MIS1084)

in partial fulfillment for the award of the degree of

Master of Technology

in

Software Engineering (5 Year Integrated Programme)

School of Computer Science and Engineering
Vellore Institute of Technology
Vandalur - Kelambakkam Road, Chennai - 600 127

November - 2023

School of Computer Science and Engineering

DECLARATION

I hereby declare that the project entitled Your Pneumonia Detection using Transfer Learning submitted by me to the School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600 127, in partial fulfillment of the requirements of the award of the degree of Master of Technology in Software Engineering (5 year Integrated Programme) and as part of SWE3004 – Software Design and Development Project is a bona-fide record of the work carried out by me under the supervision of Prof. Dr. Karthikeyan N . I further declare that the work reported in this project, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Place: Chennai

Date:

Signature of Candidate

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled Pneumonia Detection using Transfer Learning is prepared and submitted by Akash R (Reg. No. 19MIS1084) to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of Master of Technology in Software Engineering (5 year Integrated Programme) and as part of SWE3004 – Software Design and Development Project is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission.

Guide/Supervisor

Name:
Date:

HoD

Name: Dr. Nisha V M
Date:

Examiner

Name:
Date:

Examiner

Name:
Date:

(Seal of SCOPE)

Acknowledgement

I obliged to give my appreciation to a number of people without whom I could not have completed this thesis successfully.

I would like to place on record my deep sense of gratitude and thanks to my internal guide Prof. Dr.Karthikeyan N , School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, whose esteemed support and immense guidance encouraged me to complete the project successfully.

I would like to thank our HoD Dr. Nisha V M, School of Computer Science and Engineering (SCOPE) and Project Coordinator Dr.Kanchana Devi V, Vellore Institute of Technology, Chennai, for their valuable support and encouragement to take up and complete this thesis.

Special mention to our Dean Dr. Ganesan R, Associate Deans Dr. Parvathi R and Dr. S. Geetha, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Chennai, for motivating us in every aspect of software engineering.

I thank our management of Vellore Institute of Technology, Chennai, for permitting me to use the library and laboratory resources. I also thank all the faculty members for giving me the courage and the strength that I needed to complete my goal. This acknowledgment would be incomplete without expressing the whole hearted thanks to my family and friends who motivated me during the course of my work.

AKASH R
19MIS1084

Abstract

The purpose of this study is to evaluate the performance of transfer learning models from the EfficientNetV2 family for the identification of Pneumonia. Specifically, the study aims to compare the performance of EfficientNetV2M, EfficientNetV2L, EfficientNetV2B1, EfficientNetV2B2, and EfficientNetV2B3 with the older transfer learning models VGG 16 and Xception. The study also seeks to assess the effectiveness of two optimizers, Adam optimizer and RMSProp optimizer. The comparison will be based on metrics such as Accuracy, Precision, Recall, F1-Score, and Area Under the Curve Score. To avoid overfitting, the study will apply 5-fold cross-validation and fine-tuning. Additionally, the study will create an ensemble model from the top three models to optimize performance further. Based on the findings of the comparative research, various transfer learning models for pneumonia detection using 5-Fold cross-validation and fine-tuning, it was observed that EfficientNetV2M with RMSprop optimizer achieved the top performance among all the models considered. Additionally, an ensemble model consisting of the top three performing models was created, which outperformed all individual models in terms of the performance metrics for this study. As a result, it is possible that employing models from the EfficientNetV2 family might be a useful strategy for pneumonia detection, and ensemble models can further improve the performance of individual models.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

RSV	Respiratory Syncytial Virus
CXR	Chest X-Rays
WHO	World Health Organization
UNICEF	United Nations Children's Fund
CT Scan	Computed Tomography Scan
CNN	Convolutional Nueral Networks
SVM	Support Vector Machine
RF	Random Forest Classifier
KNN	K-Nearest Neighbours
DCGAN	Deep Convolutional Generative Adversarial Network
SGD	Stochastic Gradient Descent
RNN	Recurrent Neural Networks
Bi-GRU	Bidirectional Gated Recurrent Unit
CAD	Computer-Aided Design
TB	Tuberculosis
CLAHE	Contrast Limited Adaptive Histogram Equalization
AHE	Adaptive Histogram Equalization
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
NAS	Neural Architecture Search

1. Introduction

1.1 INTRODUCTION

Pneumonia is an infectious respiratory disease that affects the human lungs. The Alveoli part of the lungs is affected by Pneumonia. In the alveoli, oxygen and carbon dioxide are exchanged. For an average healthy person, the air is filled in the alveoli when he/she breathes, but for a pneumonia-infected person, the air sacs would be filled with fluid or pus, which causes phlegm or pus, and the patient will have breathing difficulties. Pneumonia is commonly caused because of bacteria, viruses, and fungi. For adults, bacterias are the common cause of Pneumonia. The most common bacteria which causes Pneumonia is called Streptococcus Pneumonia, and the second most common bacteria causing Pneumonia is Haemophilus influenzae type b (Hib)[1]. The influenza virus and the rhinovirus are the most frequent causes of viral Pneumonia in adults. RSV, or respiratory syncytial virus, is the most frequent cause of pneumonia in young infants. Common Pneumonia symptoms are chest pain, cough, breathing difficulties, low oxygen level in your blood, etc.

The doctor would initially do a physical examination test for a Pneumonia infected person. The doctor would use a stethoscope to listen to your lungs and check for abnormal sounds. If so, he/she will recommend some tests. Chest X-rays (CXR) are just one of the several diagnostics that may be used to detect pneumonia.Fig. 1(b) illustrates a lung CT scan, a lung needle biopsy, and an MRI scan [2]. The gold standard for identifying pneumonia is a chest X-ray [3]. Because CT imaging typically takes longer than X-ray imaging and there aren't enough high-quality CT scanning facilities in many developing countries, X-ray imaging is preferable to CT imaging [4]. Antibiotics can cure bacteria-infected Pneumonia, whereas for viral Pneumonia, antibiotics won't help, and our immune system has to develop antibodies to fight against the virus.

The World Health Organisation(WHO) claims that, for the year 2019, 2.5 million deaths were recorded because of Pneumonia, and most of the victims were below the age of 5. In the year 2018, around 800,000 children died because of Pneumonia. This infection is mostly found in Africa and South Asia [4]. Deaths due to Pneumonia are highest in Africa. Many African countries are poverty-stricken, and adequate healthcare facilities are in deficit. In the country of the Philippines, Pneumonia is the second leading killer. According to studies, many of these deaths caused by Pneumonia could have been prevented with timely intervention. In some remote parts of some countries, there won't be any healthcare facilities nearby, so they would not consider that option and instead rely on self-medication. This situation may increase the chances of Pneumonia and even death [8]. The threat of Pneumonia is very high now. According to UNICEF, every 46 seconds, a child dies of Pneumonia. A European study states several cases where doctors relied only on the patient's medical history to determine whether there was Pneumonia. Out of the 140 patients with Pneumonia, only 41 were diagnosed with lung disease [26].

Figure 1. shows the death rate caused because of Pneumonia per 100,000 people in the world for the year 2010 and Figure 2. shows the death rate caused because of Pneumonia per 100,000 people in the world for the year 2019.

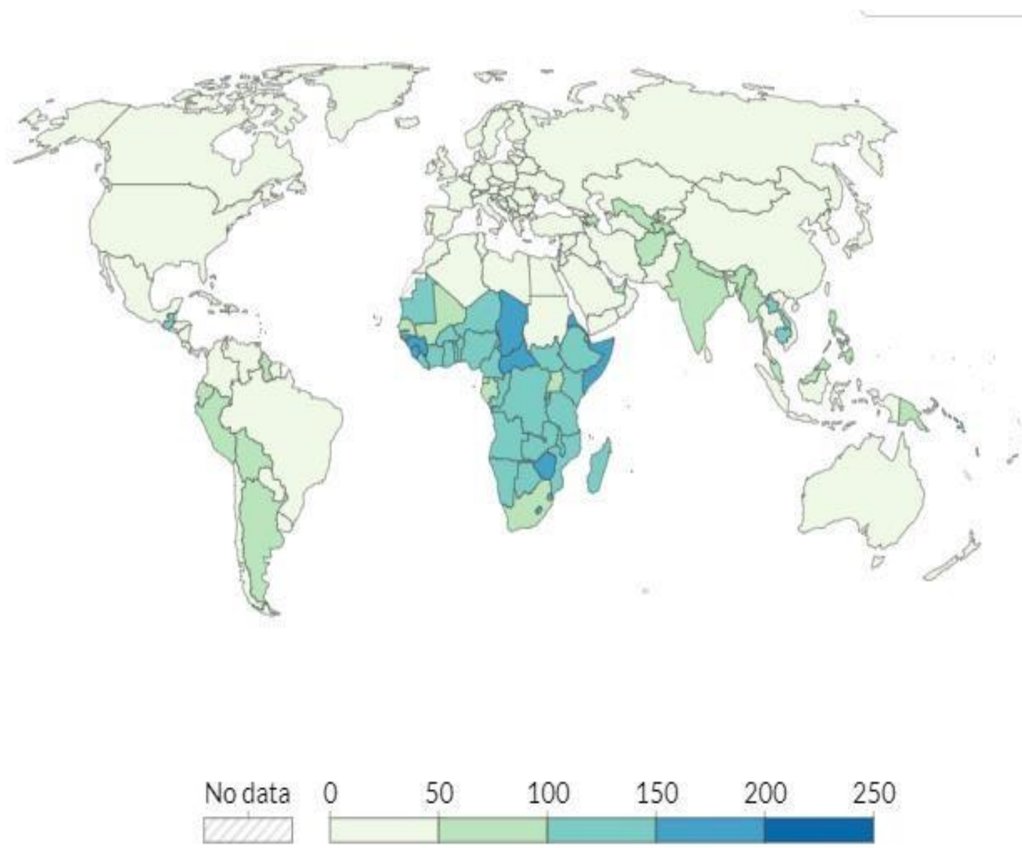


Figure 1.

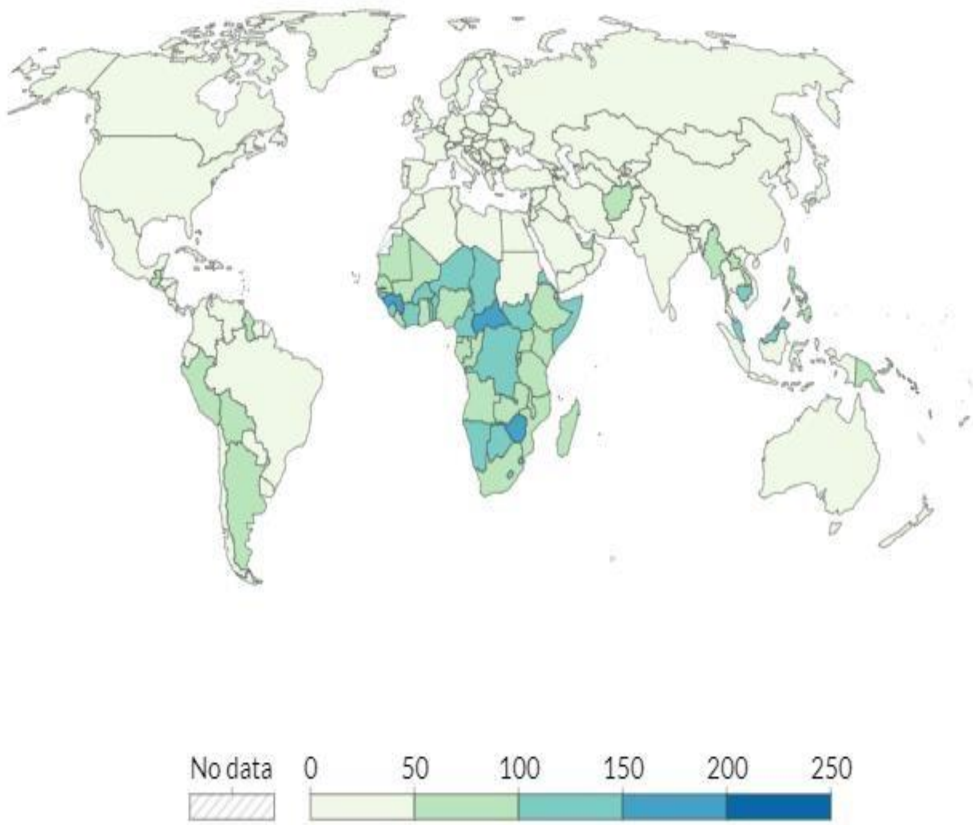


Figure 2.

As per the comparison, the death rate in 2019 is far less compared to 2010; this is primarily because healthcare facilities all across the world have significantly improved. However, not many changes are observed for the countries of Africa in that decade. We know for a fact that in Africa, very few have access to good healthcare, whereas the majority of the people are denied good healthcare access mainly because of monetary issues. So free automated systems for Pneumonia detection in these regions would prove to be beneficial for these underprivileged people.

In many remote areas of many countries, there is less access to healthcare facilities, and they have to travel far to reach the hospitals. According to studies, interpreting the chest X-ray image for Pneumonia is time-consuming compared to high-resolution CT scans [7]. There's a lot more need for health screening such as this, but not enough health facilities or doctors.

Due to the issues mentioned earlier and the ongoing rise of deep learning and machine learning in everyday life, more studies should be conducted to create an automatic system for the detection of Pneumonia. So having automated systems will enable the doctors and healthcare workers to serve a larger population and extend access to such underserved communities and will help in saving lives of many.

Several works have been done for detecting Pneumonia using chest X-rays varying from CNN models from scratch to Transfer learning models. Since data availability is less, transfer learning models are better since training from the begining requires a lot of data. Transfer learning models can be fine-tuned, which entails unfreezing the model's final few layers. Studies have shown that fine-tuning the transfer learning models will yield higher results than the transfer learning models without fine-tuning [9]. The results produced by previous works are very good.

Some works have also been done on Ensemble Learning. In ensemble models, multiple models are combined to produce the results.There are many advantages of doing ensemble models. Single models would have high variance, which means prediction would be bad for unseen data, The variance and generalisation error are reduced via an ensemble model. Average probability, weighted average probability, and majority voting are three of the most used ensemble approaches [10].

Even though many research works have been done in this field, there are still some new works. No previous works have compared Adam optimizer and RMSprop optimizer using the EfficientNetV2 models. Also, no comparison has been made for EfficientNetV2 models with the old models like VGG-16 and Xception. In this study, we will be making a comparison between RMSprop and Adam optimizer as well as a comparison between the EfficientNetV2 models and the old models. Also, to create an ensemble model and see which performs better.

1.2 CHALLENGES PRESENT

1. Limited amount of data: One of the primary challenges of using transfer learning for pneumonia diagnosis is the paucity of data.. There may not be enough data on pneumonia to properly fine-tune the pre-trained model, Even though transfer learning may utilise pre-trained models that have been developed on huge datasets.

2. Interpretability: Deep learning models used to identify pneumonia frequently involve millions of parameters, which makes them challenging to understand. As a result, selecting the features that the model uses to generate its predictions might be challenging.

3. Ethical considerations: When utilising transfer learning to identify pneumonia, there may be ethical issues to take into account, such as protecting patient data privacy and security and reducing the possibility of bias in the model's predictions.

1.3 OBJECTIVES OF THE PROJECT

1) To compare the performance of various deep-learning models, such as EfficientNetV2, VGG-16, and Xception, in identifying pneumonia.
2) To assess the effectiveness of two optimizers, Adam and RMSprop, while employing these models to identify pneumonia.
3) To assess the effectiveness of the top three models and create an ensemble model using them.
4) To contrast the effectiveness of the new (EfficientNetV2) and the old (VGG-16 and Xception) models.

1.4 SCOPE OF THE PROJECT

The study's aim is to develop a deep learning model that can precisely detect pneumonia from chest X-ray pictures. In clinical situations, where doctors and radiologists can utilise these models to help with patient diagnosis, this can be helpful. In order to maintain patient privacy and data security, A larger dataset must be used to verify and test the model initially while also taking into account the proper ethical issues.

The objective of the study is to create a transfer learning model to identify pneumonia in clinical situations using chest X-ray pictures. A larger dataset and ethical considerations are required for the model's validation. In this study, which adds to the body of literature and offers insights for clinical situations, deep learning models are compared, and the best optimisation technique is determined. Resources and regulatory approval are required for implementation.

1. Literature Survey

2.1 PREVIOUS WORKS

Ayan, E et al.[11] They applied data augmentation technique on the image dataset and did training on the VGG-19 model, VGG-16 model, ResNet-50 model, Inception -V3 model, MobileNet model, SqueezeNet model, and Xception model. From the results, they took the top three models in terms of accuracy and created an ensemble model. The same procedure was done for 3 class classification parts (Pneumonia-bacteria or Viral). El Asnaoui, K et al.[12] They applied CLAHE Image processing to the dataset and also applied data augmentation technique and made three improved iterations of the transfer learning models InceptionResNet_V2, ResNet50, and MobileNet_V2.InceptionResNet_V2, ResNet50, and MobileNet_V2 were combined to produce all seven feasible ensemble models, and the effectiveness of each model was evaluated and contrasted. Sourab, S et al.[13] They applied the Data augmentation technique, implemented a CNN model from scratch consisting of 22 layers, and used machine learning methods for the classification. The experiments were able to produce very high accuracies above 96%. Almaslukh, B.Et al. [14] applied data augmentation techniques to the dataset using the pre-trained DenseNet121 transfer learning model. The dataset was subjected to three trials utilising three different splitting techniques. In the first trial, the dataset was divided using a random split. After combining the training and test sets, the second experiment was carried out by dividing the dataset. To categorise instances of viral and bacterial pneumonia in the third trial, the pictures were divided. The results of the execution time tests indicated that the detection time of the technique was short, indicating that it was a lightweight pneumonia detection strategy that was advantageous and acceptable for use in medical systems that used less energy. Ukwuoma, C et al. [15] they employed the DenseNet201, VGG-16, Xception, InceptionResNetV2, and EfficientNetB7 models, and they applied data augmentation techniques on the dataset. Two ensemble models were also put into practise. DenseNet201, VGG16, and GoogleNet transfer learning models make up Ensemble A, whereas DenseNet201, InceptionResNetV2, and Xception transfer learning models make up Ensemble B. All the models' performances were compared. Alhudhaif A et al. [16] applied the Data augmentation technique to the dataset. The models used were ResNet_18, DenseNet_201, and SqueezeNet transfer learning models. The performance of the DenseNet-201, ResNet-18, and Squeeze-Net designs was evaluated using 5-fold cross validation, and a comparison of accuracy, Precision, Recall, and F1-score was conducted for all folds of the K-fold. Zhang, D et al. [17] applied Dynamic Histogram Equalization for image improvement, and they proposed a model from scratch. The proposed model was used with and without a residual framework, VGG, ResNet50, MobileNet, Inception, and DenseNet121.The performance of the models mentioned above with input shape as "224X224" and loss function as "Binary Cross Entropy" were compared, Trained the model with and without an image enhancement, and a comparison was made. Ikechukwu, A et al. [18] applied The Data augmentation technique to the dataset. VGG-19 and ResNet-50 were compared to IKYENet, a CNN model that was created from scratch and tested against previously trained models. The results revealed that Iyke-Net and the pre-trained models had similar performance. It was observed that the model improved to 300 epochs, and after that, its accuracy decreased, so they retrained the model to 300 epochs . The model was trained for 13 hours. Avola, D. et al.[19] applied data augmentation technique to the dataset and a comparative study of the models was done, namely AlexNet, DenseNet, GoogleNet, MnasNet, ResNet50, ResNext, ShuffleNet, SqueezeNet, VGG16, and Wide ResNet50 are examples of mobile networks. On 50%, 20%, and 10% of the training data, respectively, models were examined. Each model's training and testing times were compared and examined. Rajasenbagam, T et al.[20] applied data augmentation to the dataset. They did DCGAN technique to create artificial images from real images. Comparison on accuracy was made on the datasets produced by DCGAN Augmented dataset, Basic image dataset and combining them both.They used the VGG19 model in addition, and further comparison was made. Mujahid, M et al. [21] applied a data augmentation technique to the dataset and made a CNN Model from scratch. The model consisted of 18 layers. The CNN model was built using the "binary_crossentropy" loss function and Adam optimizer. Performance was compared using an ensemble of the aforementioned CNNs made up of Resnet-50, VGG16, and Inception-V3. They did 10-Fold Cross Validation, and its accuracy was compared. Ravi, V. et al.[22] They suggested a multichannel deep-learning method for detecting chest diseases in this research. EfficientNetB0, EfficientNetB1, and EfficientNetB2 models were employed in the study. The features derived from these models were aggregated, processed through several non-linear fully connected layers, and then sent to a stacked ensemble classifier for the diagnosis of chest diseases. The stacked ensemble classifier for lung disease detection uses RF and SVM in the first stage and Logistic Regression in the second stage. Manickam, A. et al.[23] used image augmentation on the dataset to identify the presence of pneumonia using U-Net architecture-based segmentation, and categorised whether there was pneumonia (viral or bacterial) or normal using models that had been pre-trained on the ImageNet dataset, such as ResNet50, InceptionV3, and InceptionResNetV2.They employed two optimizers. In particular, Adam and Stochastic Gradient Descent (SGD) were utilised, and their performances were examined with batch sizes of 16 and 32. Ibrahim, D. et al.[24] performed data augmentation to the dataset and assessed how well the three lung illnesses (pneumonia, COVID-19, and lung cancer) were detected. Convolutional networks and Recurrent Neural Networks (RNN) were the models used. The VGG19 and ResNet152V2 pre-trained models were combined with the traditional CNN, gated recurrent unit (GRU), and bidirectional gated recurrent unit (BiGRU). These models were contrasted with one another. Huang, M. L. et al. [25] fine-tuned the models, For COVID-19 detection, they used InceptionV3, ResNet50V2, Xception, DenseNet121, MobileNetV2, EfficientNet-B0, and EfficientNetV2S. Additionally, they suggested the "LightEfficientNetV2" CNN model, which was constructed using CT and Chest X-ray data. They used 5-fold cross-validation to assess the models' performance. To examine the effectiveness of the suggested model, LightEfficientNetV2 was compared using it on three distinct datasets (SARS-CoV-2, NIH Chest X-rays, and COVID-CT).All models' performance was examined and contrasted.

Section 2.2 gives the comparative study of the results, dataset used and of papers discussed before.

2.2 COMPARISON OF RESULTS

S.No	Diseases	Dataset Used	Results Obtained
[11]	Pneumonia	MendeleyDataset (Kaggle)	Resnet-50-94.32 Xception-95.03 Mobilenet-94.87 Ensemble-95.83
[12]	Pneumonia and Covid19	Mendeley(Kaggle) and Covid19 Chest X-ray dataset	ResNet50 - 94.50% precision Accuracy of MobileNet_V2 is 93.73%. Accuracy of InceptionResNet_V2: 94.50% InceptionAccuracy of ResNet_V2 with ResNet50: 93.6 percent 93.82% accuracy for MobileNet_V2 and InceptionResNet_V2 95.17% accuracy for MobileNet_V2 and ResNet50 95.09% accuracy for MobileNet_V2,InceptionResNet_V2, ResNet50 The combination of MobileNet_V2 and ResNet50 provided the highest accuracy while being significantly faster.
[13]	Pneumonia	Mendeley(Kaggle) Dataset	CNN-RF->99.52 CNN-SVM-> 97.32 CNN-KNN->96.55
[14]	Pneumonia	Mendeley(Kaggle) Dataset	The Model achieved for the first experiment- 94.4% Accuracy Second experiment-98.9% Accuracy, Third experiment 96.3% Accuracy

[15]	Pnuemonia	Mendeley (Kaggle) Dataset Also, a 3-class classification problem (Pneumonia-Viral, Bacteria) with the dataset was done.	Out of all the models, DenseNet201 gave the highest accuracy of 96.95%, and all the other models were above 92%. Out of the Ensemble models, the Ensemble Model A model was found to be better. Ensemble A- 97.22% accuracy for 2- class problem and 97.20% for 3- class problem Ensemble B- 96.44% accuracy for 2- class problems and 96.43% for the 3-class problem
[16]	Covid-19 Pneumonia	Kaggle(Mendeley) Dataset, Cohen and Wang (Covid Datasets)	DenseNet_201 was found to have better accuracy on all folds of the 5- fold crossvalidation. DenseNet_201- 94.96% accuracy ResNet_19- 91.60% accuracy SqueezeNet- 89.92% accuracy
[17]	Pneumonia	Kaggle(Mendeley) Dataset	The proposed Model- Accuracy: 96.068% VGG-16- Accuracy,-94.359%, Xception Model- Accuracy, 96.068% DenseNet121- Accuracy-87.350% MobileNet - A ccuracy-95.473%.

			InceptionV3Net- Accuracy-97%
[21]	Pneumonia	Kaggle (Mendeley) Dataset	VGG-16 Model-97.93% Accuracy CNN- 98.25% Accuracy Inception-V3 96.58% Accuracy ResNet-50 97.87% Accuracy VGG-16+CNN-98.06% Accuracy Inception-V3+CNN -99.29% Accuracy ResNet-50+CNN -99.09% Accuracy
[22]	Covid-19, Pneumonia, Tuberculosis	Kaggle (Mendeley) TB Dataset Covid 19 Dataset	The proposed Ensemble model fared better in terms of accuracy for all three datasets than the other models. Accuracy-98% Dataset TB Accuracy-99% Covid19 dataset: 98% accuracy
[23]	Pneumonia	Kaggle(Mendeley) Dataset	Models were executed with batch size as 16 had better accuracies than Models with a batch size of 32. Models done with an SGD optimizer have better accuracy than the Adam optimizer. Models done with 0.0001 learning rate.
[24]	Covid-19 , Lung Cancer and Pneumonia	CT dataset	VGG19+CNN - Accuracy-98.05% ResNet152V2 with GRU- Accuracy-96.09% ResNet152V2- Accuracy-95.31% ResNet152V2 with Bidirectional-GRU- Accuracy-93.36% VGG19+CNN model having

			Precision 98.43%, F1 score 98.24%, Specificity 99.5%, Accuracy 98.05%, Recall 98.05%, Negative predictive value-99.3%, AUC-99.66%, MCC(Matthews correlation coefficient) 97.7%
[25]	Covid-19 and Pneumonia	SARS-CoV-2 dataset, NIH Dataset, and COVID-CT Dataset	Their LightEfficientNetV2 Model for Chest X-Ray images gave 98.33% accuracy and for CT scanned images- 97.48% accuracy Their LightEfficientNetV2 had far less running and training time as compared to the other models

Table 1. Comparison of previous works

2.3 WORK DONE OVER THE PREVIOUS YEARS

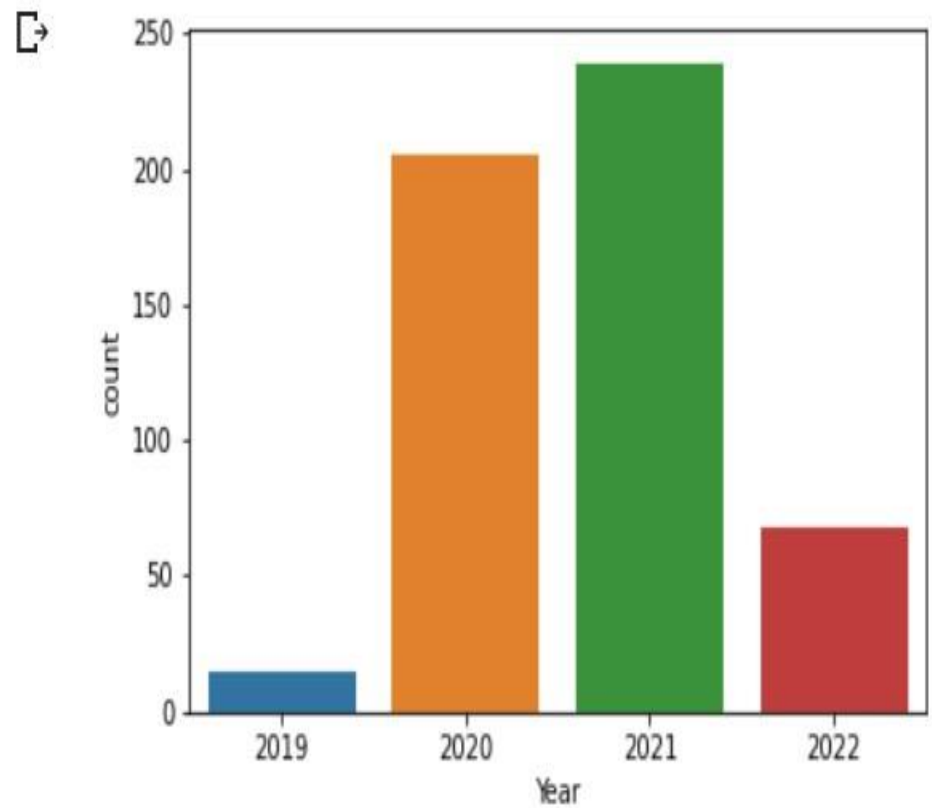


Figure 3.

Figure 3. shows all the works related to transfer learning and CNN from images, including Pneumonia detection, COVID-19, and others from the year 2019 to 2023 through web scraping in Google Scholar. After going through the papers, only the following publications were taken, namely Springer, ieeexplore.ieee.org, Elsevier, mdpi.com, hindawi.com, researchgate.net, and medrxiv.org. A total of 527 papers were scraped, and the highest number of papers were published in 2021 (239 papers), followed by 205 papers in 2020. This is because this was the peak time in the development of CAD (Computeraided Design) systems and when there were rapid works done on CNN and transfer learning. In 2022, 68 papers were done related to Transfer learning.

Figure 4. shows the distribution of the keywords while scraping the titles for works related to pneumonia detection using transfer learning. The keyword 'Convolution' takes up to be the majority describing how commonly the word convolution is used for most of the works, followed by the keyword ‘Pneumonia’.

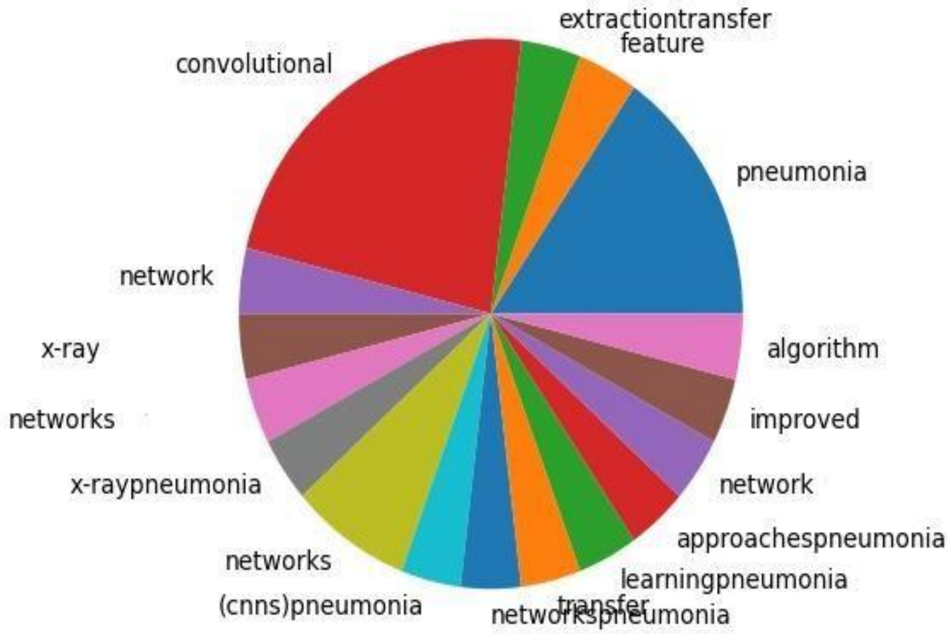

extractiontransfer
feature
convolutional
pneumonia
network
algorithm
x-ray
improved
networks
network
x-raypneumonia
approachespneumonia
networks
learningpneumonia
(cnns)pneumonia
transfer
networkspneumonia

Figure 4.

1. Methodology and Materials

3.1 INTRODUCTION

Numerous millions of individuals get pneumonia each year, a potentially fatal respiratory illness. Early identification of pneumonia is essential for successful treatment because postponed diagnosis might result in serious consequences. Convolutional neural networks (CNNs), a type of deep learning methodology, are one method for detecting pneumonia. CNNs are effective machine learning models that can be trained to find patterns in image data, making them suitable for the study of medical images.

The materials required for pneumonia detection using CNNs include a dataset of chest Xrays that have been classified as pneumonia or normal. The dataset should be large enough to provide sufficient variation in image characteristics and should be properly preprocessed to ensure that the images are of high quality. The methods for pneumonia detection using CNNs involve training the network on the labeled dataset, which involves feeding the network batches of images and To lessen the disparity between the output that was predicted and what was generated, the network's weights were adjusted. The trained CNN may then be used to accurately categorise fresh chest X-ray pictures as normal or pneumonia.

In our study, we aim to do a comparative study of different Deep-Learning models for the detection of Pneumonia to do a comparative study of two optimizers, namely Adam and RMSprop. Papers for Pneumonia detection using EfficientNetV2 models are a handful, and no comparative studies have been conducted on different optimizers for these models. We have also included some old models like VGG-16 and Xception to understand the difference in the performance of the new models compared to the old ones. We aim to build an Ensemble model from the top three performing models and to do a comparative study of all models. Our methodology is shown in Figure 5.

Figure 5.

3.2 DATASET DETAILS

The dataset used was from Guangzhou Women and Children's Medical Center, a publicly available dataset on Kaggle consisting of 5840 images in jpeg format and has sections for training and testing. There are 5216 photos in the training portion, and 624 images in the testing portion. There are two classifications at play: Normal and Pneumonia. The category for pneumonia includes pictures of both bacterial and viral pneumonia. However, we are not making the individual prediction for both (Bacterial Pneumonia and Viral Pneumonia). Table 2 provides information about the dataset.

Train		Testing	
Pneumonia	Normal	Pneumonia	Normal
3875	1341	390	234

Table 2.

The dataset was quite uneven, with more pneumonia cases than usual cases, hence data augmentation techniques were applied. Doing this will diminish any chances of the model being over-fit.

Figure 6 shows the X-ray images obtained from the dataset for Normal and Pneumonia cases.

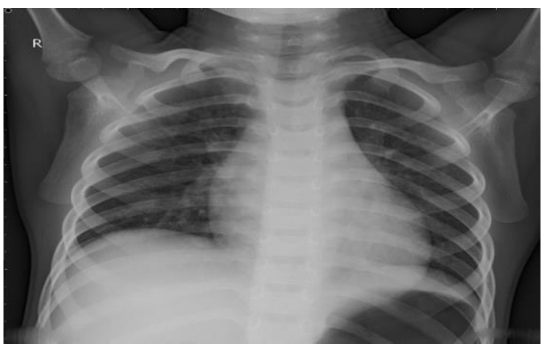


Figure 6 (a) Normal

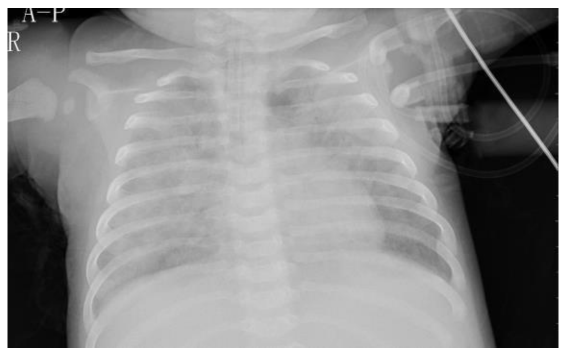


Figure 6(b) Pneumonia

3.3 IMAGE PRE-PROCESSING AND DATA AUGMENTATION

3.3.1 CLAHE

Image pre-processing is an interesting technique that enables us to get rid of undesirable distortions in the image and raise the picture's overall quality. In this study, we have done CLAHE image pre-processing technique. The CLAHE is an image pre-processing technique that aids in improving the contrast of the picture. CLAHE is an enhanced variation of AHE [28]. AHE is a pre-processing method for digital images that increases visual contrast. [29]. Studies conducted by Rubini et al. [30] showed that CLAHE produced better results than AHE. The CLAHE approach places a focus on boosting local contrast to get past the drawbacks of global techniques. Important hyper-parameters for this technique include the tile size and clip limit. The title size and Clip limit we have done for this study are shown below in Table 3.

Tile size	Clip limit
(8,8)	3

Table 3.

Figure 3(a) shows the image of a Normal Chest X-ray, Figure 3(b) shows the image of Pneumonia infected chest X-ray, Figure 4(a) shows the image of a Normal Chest X-ray after applying CLAHE, and Figure 4(b) shows the image of a Pneumonia infected Chest X-ray image.

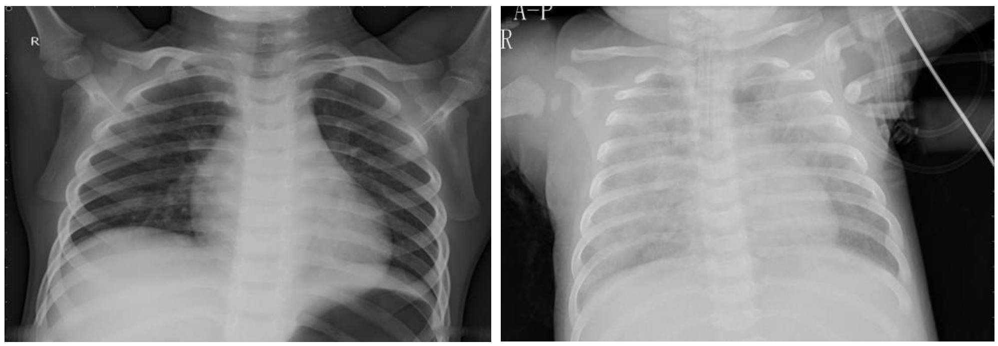
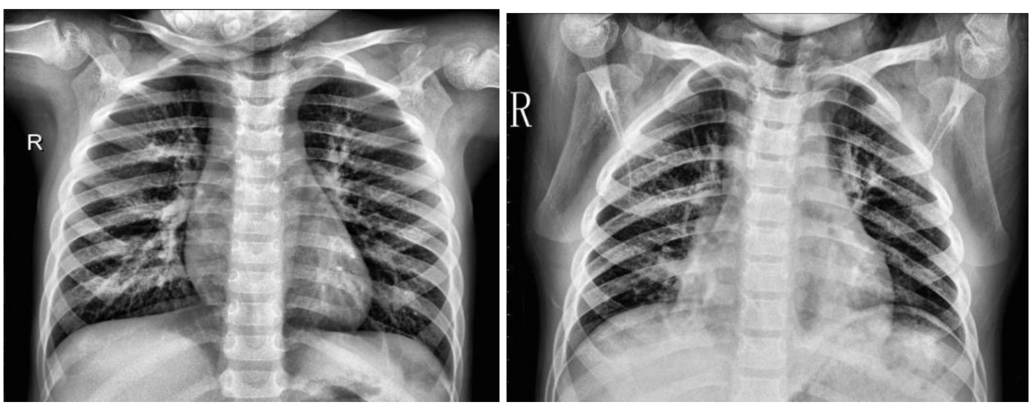


Figure 7 (a) Normal Figure 7(b) Pneumonia



Figure

8 (a) Normal Figure 8 (b) Pneumonia

Figure 7 and 8 clearly show the difference CLAHE image pre-processing has brought. It improved the quality of the picture, which will help with better feature extraction, thus improving the quality of the model.

3.3.2 DATA AUGMENTATION

Data augmentation is a very famous and commonly used technique in the field of Computer Vision. Using random (but realistic) transformations increases the variety and quantity of training data. For instance, image resizing, transformation, flipping, and many other options. This method enables us to obtain data that is already present in a more varied manner, improving the training set and, consequently, the trained model. Another advantage of undertaking this method is that it aids in lessening the overfitting issue. For this study, we have applied the following data augmentation transformations shown in Table 4 for the training images.

Transformations	**Value**
Rescale	1./255
Shear range	10
Zoom range	0.3
Horizontal flip	True
Vertical flip	True
Brightness range	0.5-2.0
Width shift range	0.2
Rotation range	20

Table 4.

3.4 TRANSFER LEARNING MODELS

3.4.1 INTRODUCTION AND DETAILS

The concept of transfer learning models is the application of similar knowledge acquired from one job to another. Many publicly available transfer learning models are available in the Keras library [31]. All of these models have been created for the ImageNet classification challenge. Studies have shown that fine-tuning a model performs comparatively better than the models without fine-tuning. So we will be fine-tuning the Deep-learning models, The Deep-learning models used are EfficientNetV2L, EfficientNetV2M, EfficientNetV2B1, EfficientNetV2B2, EfficientNetV2B3, VGG-16, and Xception. For all the models, 30 layers were frozen except the VGG-16 model. For the VGG-16 model, all of the layers were unfrozen. For all the models, global average pooling was used as the output of the convolution layer. We have added Batch Normalization, Dropout layers, and two fully connected layers. Our approach is shown in Figure 9.

We have applied ReduceLROnPlateau with a factor of 0.1, patience of 7, and monitor as "validation_loss.". Early Stopping was used with the monitor as "validation_accuracy," patience as 9, and mode as "max."

For all the models, we have used a learning rate of 0.0001. For all of the models, a batch size of 16 was used, and all of the images were resized to 224X224.

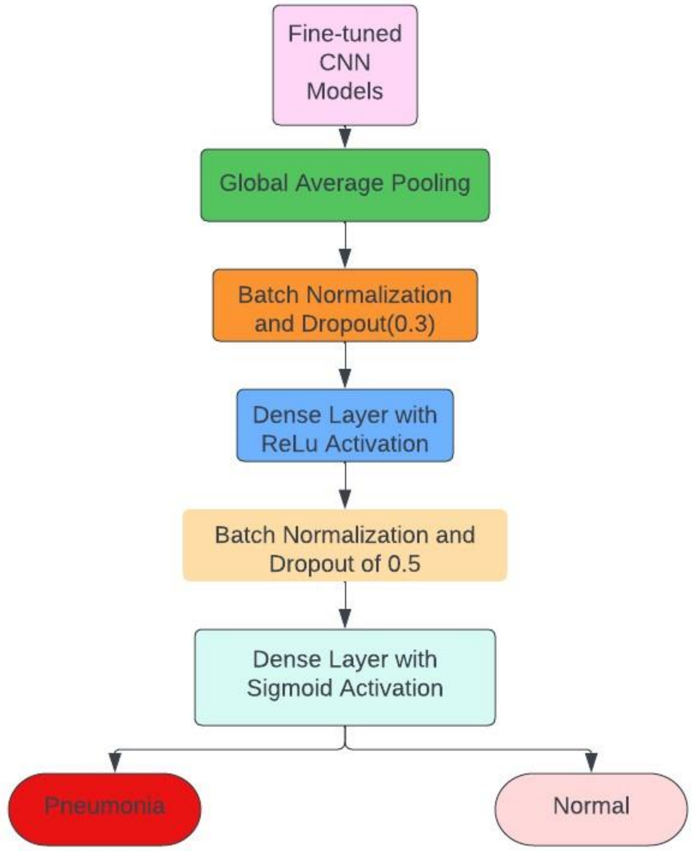


Figure 9.

3.4.2 VGG16

The 2014 ILSVR (ImageNet) competition was won by the VGG-16. It is often referred to as the most cutting-edge vision model design that is currently accessible. The VGG-16 network was trained with the aid of the ImageNet database. The existence of 16 weighted layers is indicated by the number 16 in VGG-16 [32]. Due to its extensive training, the Even with limited picture datasets, the VGG-16 network provides remarkable accuracy. VGG-16 is an image classification algorithm that can classify 1000 images into 1000 categories with an object detection accuracy of 92.7%. The figure of VGG-16 is shown in Figure 12.

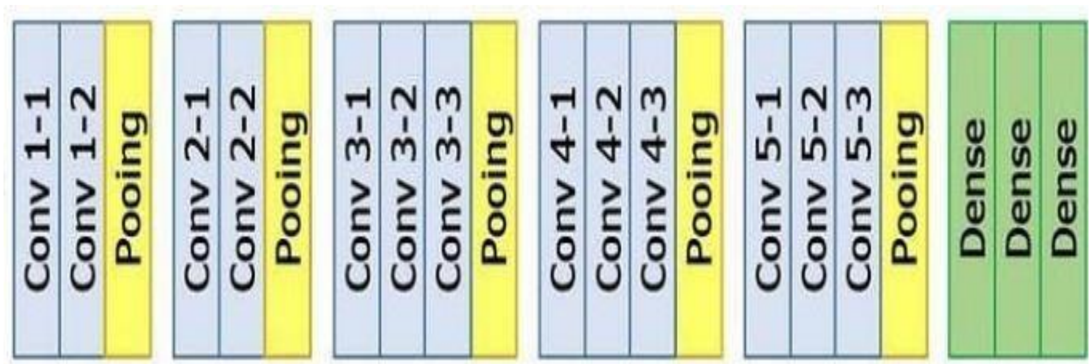


Figure 10.

3.4.3 XCEPTION MODEL

The Xception CNN architecture was introduced in the year 2017. The term stands for "extreme inception.". The Xception model is a CNN model incorporating Depthwise Separable Convolutions with 71 layers. This model was created by Francois Chollet, who works at Google [33]. Figure 7 illustrates the entry flow, middle flow, and exit flow components. and the middle flow is made to do eight times. On the ImageNet dataset, Xception barely outperforms Inception v3, and dramatically exceeds it on a larger image classification dataset with 17,000 classes [34]. The architecture of Xception model is shown in Figure 13.

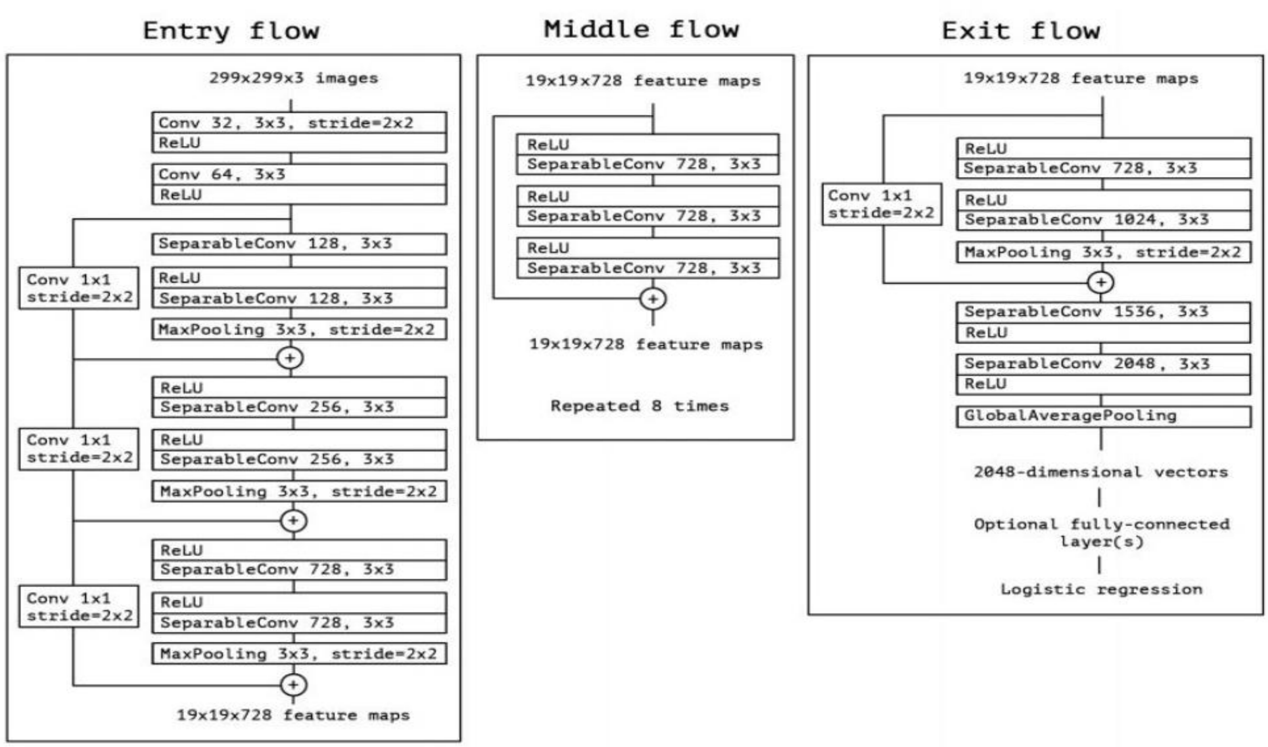


Figure 11.

3.4.4 EFFICIENTNETV2

Initially, a series of models were produced called EfficientNet, also known as EfficientNetV1, but it was having some issues, such as very sluggish training speeds for EfficientNet-B3 to EfficientNet-B7 when the picture size is big, Depthwise convolutions tend to hold down training, and equal scaling up of height, width and the resolution was not the optimal way. A constant increase in picture size causes a significant increase in RAM usage, which slows down training, because of the above-mentioned problems EfficientNetV2 convolutional neural network was created in 2021 to overcome them.
In addition, [35] demonstrated that increasing training speed may be accomplished by early replacement of the MBConv module with the Fused-MBConv module.However, using the Fused-MBConv module in place of each individual MBConv layer would result in a significant increase in the number of parameters and FLOPs, which would slow down training. In order to determine the ideal fusion of MBConv and Fused-MBConv, Neural Architecture Search (NAS) was applied. In this paper, they have implemented progressive learning, which overcomes the third issue addressed above. The architecture of EfficientNetV2S, created by NAS, is shown in Figure 14.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Figure 12.

A series of models were produced after scaling up and scaling down EfficientNetV2S namelyEfficientNetV2M, EfficientNetV2L, EfficientNetV2B0, EfficientNetV2B1, EfficientNetV2B2, EfficientNetV2B3, However in our study, we have excluded EfficientNetV2B0 and EfficientNetV2S since some works have already been done on Pneumonia Detection using those models.

4. Performance Metrics

4.1 DETAILS

The performances metrics namely recall, accuracy, precision, F1-score, and AUC score are employed in our investigation. The equations for the above metrics are shown below.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Weighted Precision} = \sum_{i=1}^{n} w_i \times Precision_i$$

$$\text{Weighted Recall} = \sum_{i=1}^{n} w_i \times Recall_i$$

$$\text{Weighted F1score} = \sum_{i=1}^{n} w_i \times F1score_i$$

Area Under Curve (AUC) Score and ROC Curve: The degree of separability is shown by the ROC (receiver operating characteristics), which is a probability curve. The sensitivity (% of true positives) against specificity (% of false positives) ratio is plotted on the ROC curve.

Where,

$$W_i = \frac{\text{No of samples in class i}}{\text{Total no of samples}}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1 Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN)

TP stands for the percentage of positively categorised categories that are actually positive.

FP stands for the percentage of negatively categorised categories that should really be classified as positive.

The number TN represents the percentage of negatively classed categories that are really classified as such.

The percentage of positive categories that are mistakenly labelled as negative is indicated by the abbreviation FN.

5. Results

5.1 RESULTS OBTAINED

These are set of results obtained after applying 5-fold cross validation on all the models for both RMSprop optimizer and Adam optimizer.

Tables 5 and 6 display the EfficientNetV2L model's Accuracy, Precision, Recall, F1-score, and AUC score using Adam and RMSprop optimizers.

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	96.65	96.67	96.65	96.61	0.97
2	96.26	96.24	96.26	96.24	0.96
3	95.78	95.88	95.78	95.7	0.97
4	96.06	96.17	96.07	96	0.97
5	95.87	96.02	95.88	95.79	98.16
Average	96.122	96.196	96.128	96.068	0.968
SD	0.3471743078	0.2992156413	0.3449202806	0.3676547293	43.4658076

Table 5. EfficientNetV2L-RMSprop

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	97.31	97.32	97.32	97.3	0.97
2	96.06	96.07	96.07	96.02	0.96
3	97.31	97.43	97.32	97.34	0.96
4	95.78	95.78	95.78	95.73	0.96
5	95.68	95.7	95.69	95.62	0.96
Average	96.428	96.46	96.436	96.402	0.962
SD	0.8171107636	0.8474373133	0.8191031681	0.8507761163	0.004472135955

Table 6. EFFICIENTNETV2L- Adam

The above two tables show that EfficientNetV2L with Adam optimizer has better accuracy and F1 score than EfficientNetV2L with RMSprop optimizer.

Tables 7 and 8 display the EfficientNetV2M model's Accuracy, Precision, Recall, F1score, and AUC score using Adam and RMSprop optimizers.

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	96.64	96.67	96.65	96.61	0.97
2	96.26	96.27	96.26	96.22	0.96
3	97.31	97.37	97.32	97.33	0.96
4	96.64	96.69	96.64	96.6	0.97
5	97.02	97.04	97.03	97	0.97
Average	96.77	96.808	96.78	96.752	0.966
SD	0.40246739	0.4159567285	0.4065095325	0.4248176079	0.005477225575

Table 7. EfficientNetV2M- RMSprop

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	96.07	96.14	96.07	96.01	0.97
2	97.31	97.33	97.32	97.29	0.98
3	97.02	97.02	97.03	97.01	0.97
4	95.87	95.87	95.88	95.83	0.96
5	95.39	95.39	95.4	95.34	0.95
Average	96.332	96.35	96.34	96.296	0.966
SD	0.8061141358	0.807062575	0.8069386594	0.8232132166	0.01140175425

Table 8. EfficientNetV2M- Adam

The above two tables show that EfficientNetV2M with RMSprop optimizer has better accuracy and F1 score than EfficientNetV2M with Adam optimizer.

Tables 9 and 10 display the EfficientNetV2B3 model's Accuracy, Precision, Recall, F1score, and AUC score using Adam and RMSprop optimizers.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	95.68	95.7	95.69	95.69	0.94
2	96.83	96.82	96.84	96.82	0.96
3	97.02	97.03	97.03	97	0.97
4	95.01	94.99	95.01	94.96	0.95
5	95.2	95.17	95.21	95.16	0.95
Average	95.948	95.942	95.956	95.956	0.954
SD	0.9271299801	0.9374806665	0.9296666069	0.9391911414	0.01140175425

Table 9. EfficientNetV2B3- RMSprop

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	95.78	95.79	95.79	95.79	0.94
2	96.83	96.83	96.84	96.81	0.97
3	95.2	95.17	95.21	95.16	0.95
4	96.26	96.25	96.26	96.23	0.96
5	97.02	97.05	97.03	97.04	0.96
Average	96.218	96.226	96.226	96.206	0.956
SD	0.7496132336	0.7663680578	0.7494864909	0.7629089068	0.01140175425

Table 10. EfficientNetV2B3- Adam

The above two tables show that EfficientNetV2M with RMSprop optimizer has better accuracy and F1 score than EfficientNetV2M with Adam optimizer.

Tables 11 and 12 display the EfficientNetV2B2 model's Accuracy, Precision, Recall, F1score, and AUC score using Adam and RMSprop optimizers.

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	95.78	95.76	95.79	95.79	0.95
2	96.45	96.49	96.45	96.45	0.97
3	95.78	97.37	97.37	97.37	0.95
4	93.95	93.93	93.96	93.96	0.94
5	95.3	95.36	95.3	95.3	0.96
Average	95.452	95.782	95.774	95.774	0.96
SD	0.934114554	1.287583007	1.276687119	1.276687119	0.01140175425

Table 11. EfficientNetV2B2- RMSprop

Fold	Accuracy	Precision	Recall	F1-Score	AUC
1	93.58	93.62	93.58	93.45	0.94
2	95.01	95.02	95.01	94.94	0.95
3	95.87	95.9	95.88	95.82	0.96
4	94.91	94.88	94.92	94.87	0.94
5	93.76	93.91	93.77	93.62	0.95
Average	94.626	94.666	94.632	94.54	0.948
SD	0.9512780876	0.9164496713	0.9530320037	0.9926983429	0.008366600265

Table 12. EfficientNetV2B2- Adam

The above two tables show that EfficientNetV2B2 with RMSprop optimizer has better accuracy and F1 score than EfficientNetV2B2 with Adam optimizer.

Tables 13 and 14 display the EfficientNetV2B1 model's Accuracy, Precision, Recall, F1score, and AUC score using Adam and RMSprop optimizers.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	94.54	94.57	94.54	94.45	0.95
2	94.91	94.93	94.92	94.84	0.95
3	95.78	95.76	95.78	95.75	0.95
4	95.87	95.86	95.88	95.87	0.95
5	94.63	94.67	94.63	94.54	0.95
Average	95.146	95.198	95.15	95.09	0.95
SD	0.6354761994	0.6105489333	0.6374166612	0.674277391	0

Table 13. EfficientNetV2B1- RMSprop

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	95.4	95.39	95.4	95.35	0.95
2	94.35	94.33	94.25	94.13	0.95
3	95.3	95.27	95.3	95.27	0.95
4	96.26	96.25	96.26	96.25	0.95
5	95.3	95.27	95.3	95.26	0.95
Average	95.332	95.302	95.302	95.252	0.95
SD	0.6766978646	0.6806761344	0.7131058827	0.7524759132	0

Table 14. EfficientNetV2B1- Adam

The above two tables show that EfficientNetV2B1 with Adam optimizer has better accuracy and F1 score than EfficientNetV2M with RMSprop optimizer.

Tables 15 and 16 display the VGG16 model's Accuracy, Precision, Recall, F1-score, and AUC score using Adam and RMSprop optimizers.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	95.68	95.7	95.69	95.69	0.94
2	98.17	98.18	98.18	98.18	0.98
3	95.3	95.3	95.3	95.3	0.94
4	91.85	91.75	91.85	91.74	0.91
5	96.64	96.76	96.64	96.68	0.95
Average	95.528	95.538	95.532	95.518	0.944
SD	2.335566312	2.392973046	2.338561524	2.386780258	0.0250998008

Table 15. VGG16- RMSprop.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	91.38	92.32	91.38	91.03	0.94
2	97.22	97.32	97.22	97.25	0.95
3	96.64	96.68	96.64	96.6	0.97
4	92.91	92.99	92.91	92.94	0.9
5	98.08	98.08	98.08	98.07	0.98
Average	95.24	95.478	95.246	95.178	0.948
SD	2.927213692	2.634923149	2.927213692	3.037000165	0.031144823

Table 16. VGG16- Adam.

The above two tables show that VGG16 with Adam optimizer has better accuracy and F1 score than VGG16 with RMSprop optimizer. However, the standard deviation is high, and values in different folds vary a lot making it clear that both are unstable models.

Table 17 and Table 18 display the Xception model's Accuracy, Precision, Recall, F1-score, and AUC score using Adam and RMSprop optimizers.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	94.44	94.86	94.44	94.28	0.96
2	93.48	93.87	93.28	93.28	0.95
3	96.93	97.02	96.93	96.88	0.98
4	94.24	94.5	94.25	94.1	0.96
5	96.16	96.28	96.16	96.09	0.97
Average	95.05	95.306	95.102	94.926	0.964
SD	1.436975991	1.303564344	1.492002011	1.498492576	0.01140175425

Table 17. Xception- Adam.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	94.06	94.3	94.06	93.91	0.95
2	95.11	95.16	95.02	93.91	0.96
3	94.34	94.77	94.34	94.17	0.96
4	94.05	94.49	94.06	93.88	0.96
5	94.34	94.69	94.34	94.18	0.96

Average	94.38	94.682	94.364	94.01	0.958
SD	0.4322614949	0.3236819427	0.3925302536	0.1511621646	0.004472135955

Table 18. Xception- RMSprop.

The above two tables show that Xception with Adam optimizer has better accuracy and F1 score than Xception with RMSprop optimizer. However, the standard deviation for the Xception model with the Adam optimizer is high, making it an unstable model. However Xception model with RMSprop optimizer tends to show stability even though the performance values are a bit lower than its counterpart. So Xception model with RMSprop is better.

Table 19. shows the comparison of all seven models with different optimizers in terms of the performance metrics mentioned before, and the top three models are highlighted.

Model	Accu racy	Preci sion	Reca lll	F1-Score	AUC Score
EfficientNetV2M-RMSprop	96.77	96.808	96.78	96.752	0.966
EfficientNetV2M-Adam	96.332	96.35	96.34	96.296	0.966
EfficientNetV2L-Adam	96.428	96.46	96.436	96.402	0.962
EfficientNetV2LRMSprop	96.122	96.196	96.128	96.068	0.968
EfficientNetV2B3-Adam	96.218	96.226	96.226	96.206	0.956
EfficientNetV2B3RMSprop	95.948	95.942	95.956	95.956	0.954
EfficientNetV2B2RMSprop	95.452	95.782	95.774	95.774	0.96
EfficientNetV2B2-Adam	94.626	94.666	94.632	94.54	0.948
EfficientNetV2B1RMSprop	95.146	95.198	95.15	95.09	0.95

EfficientNetV2B1-Adam	95.332	95.302	95.302	95.252	0.95
VGG16-Adam	95.528	95.538	95.532	95.518	0.944
VGG16-RMSprop	95.24	95.478	95.246	95.178	0.948
Xception-Adam	95.05	95.306	95.102	94.926	0.964
Xception-RMSprop	94.38	94.682	94.364	94.01	0.958

Table 19.

For our study, we give importance to the weighted average F1-score and AUC score because the dataset is unbalanced.

The weighted average F1-score for all models is above 94%, indicating that all models perform well on the dataset as it considers both precision and recall for both classes weighted by the number of samples in each class.

The AUC score for all the models is above 94.4%, implying that high true positive and low false positive rates are characteristics of the model. and indicating that all the models are performing well in distinguishing between positive and negative classes. Since AUC score is not affected by class imbalance it serves as a good metric.

The Training vs. Testing accuracy curve and ROC curve for the top three models for the first fold are shown in Tables 20, Table 21, and Table 22, respectively.

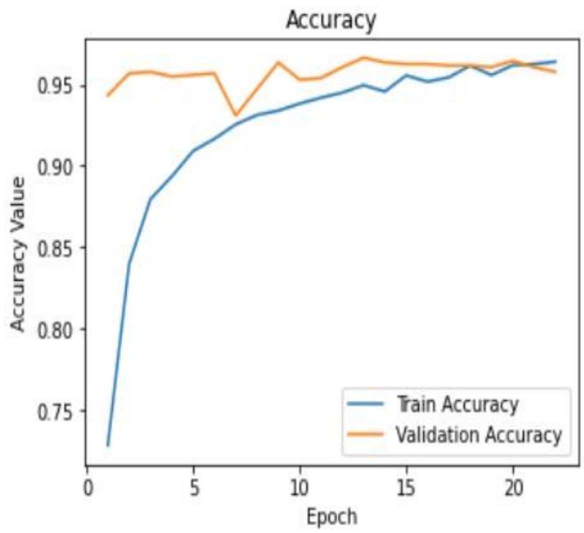
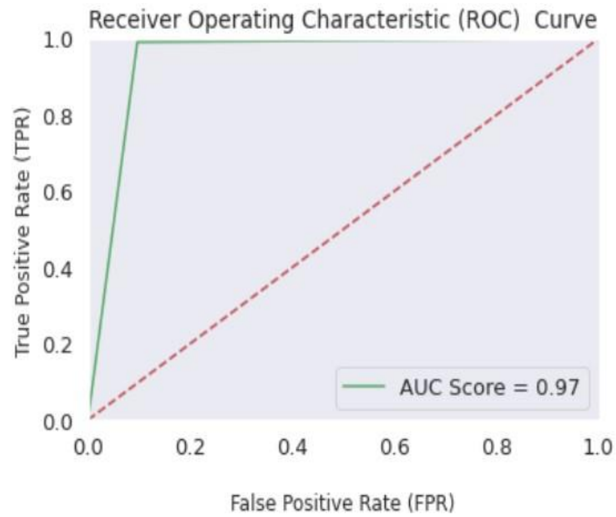
EfficientNetV2M-RMSprop (Fold 1)	
Training vs. testing curve	**ROC Curve**
	

Table 20.

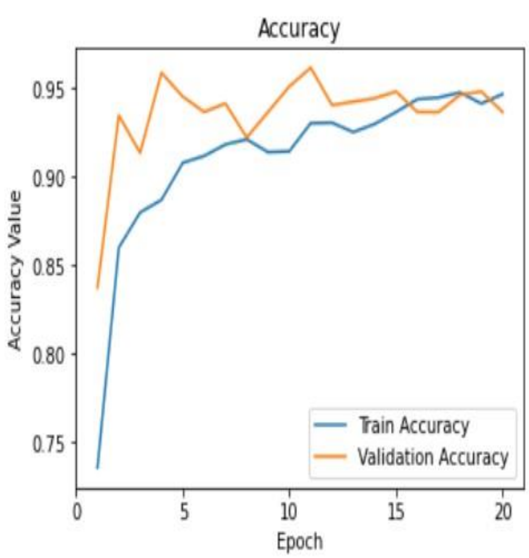
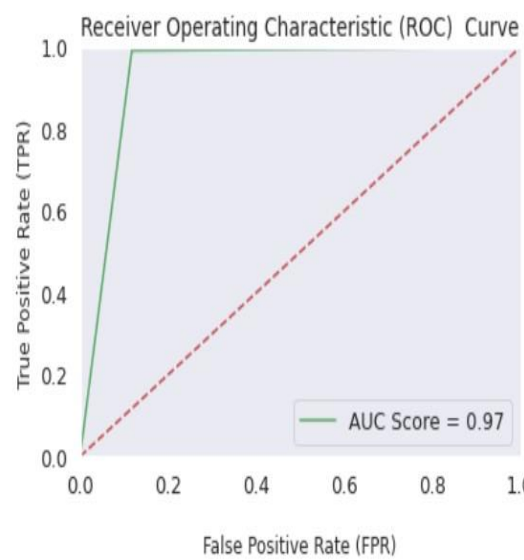
EfficientNetV2M-Adam (Fold 1)	
Training vs. testing curve	**ROC Curve**
	

Table 21.

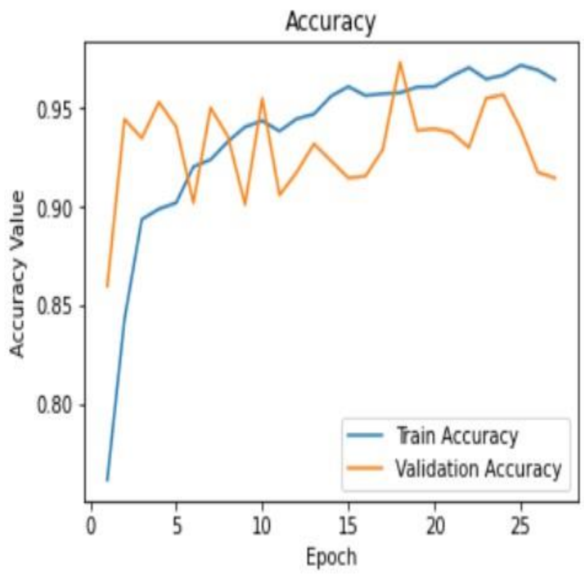
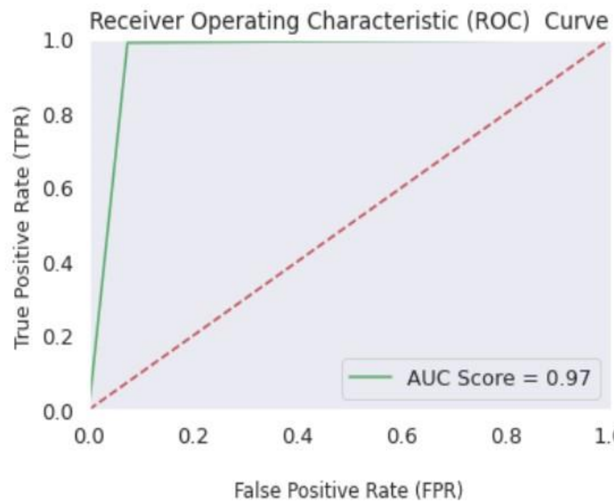
<table>
<tr><td>EfficientNetV2L-Adam(Fold 1)</td><td></td></tr>
<tr><td>Training vs. testing curve</td><td>ROC Curve</td></tr>
<tr><td>
</td><td>
</td></tr>
</table>

Table 22.

5.2 ENSEMBLE MODEL

The previous results show that the top-performing models are EfficientNetV2M-RMSprop, EfficientNetV2M-Adam, and EfficientNetV2L-Adam. We have combined these three models to create an ensemble model using the average ensembling method, and its performances were compared with the previous models. Table 23. Shows the results obtained for the ensemble model.

Fold	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
1	96.36	98.8	96.25	97.5	0.96
2	96.64	98.94	96.51	97.7	0.97
3	98.08	98.83	98.58	98.7	0.98
4	97.6	98.82	97.93	98.17	0.97
5	97.41	99.34	97.16	98.23	0.98

Average	97.218	98.966	97.286	98.06	0.972
SD	0.7064134767	0.2268920448	0.9721779672	0.472704982	0.008366600265

Table 23.

The standard deviation for F1-score is 0.47, and the accuracy is 0.70, which makes it a good model. The average AUC score for the Ensemble model is 0.972, indicating that the model's performance is good in distinguishing between positive and negative classes. Figure 15 and 16 Shows the ROC curve and confusion matrix respectively for the first fold.

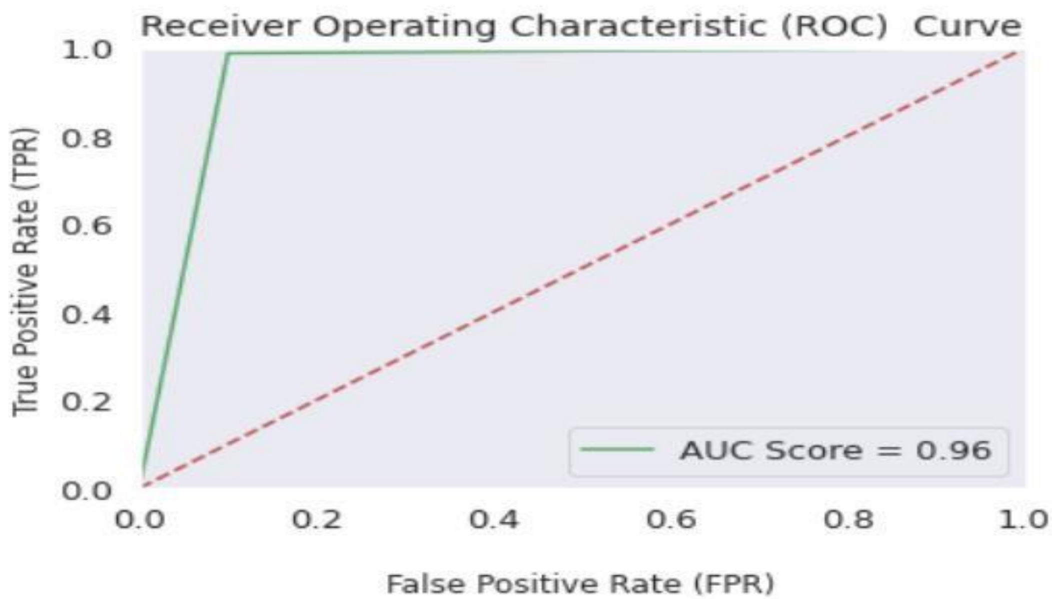


Figure 13.

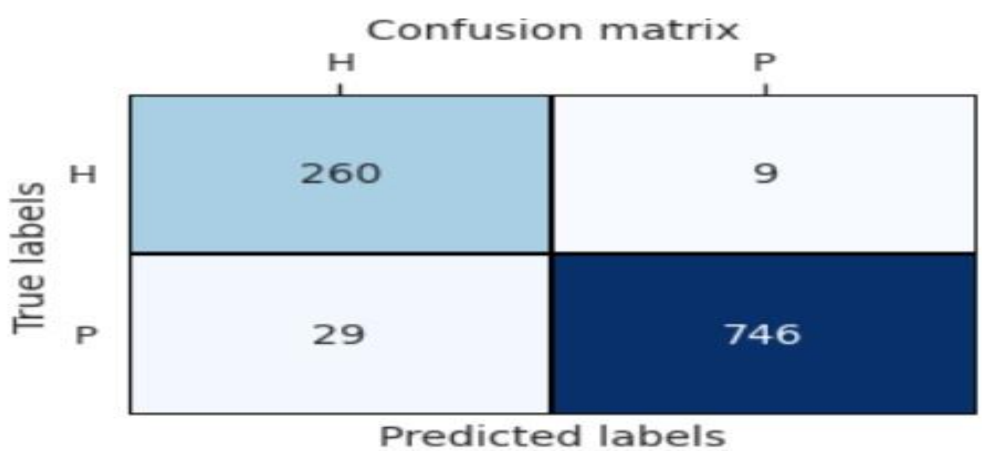


Figure 14.

Figure 17. shows the comparison between the Ensemble model and the top three models

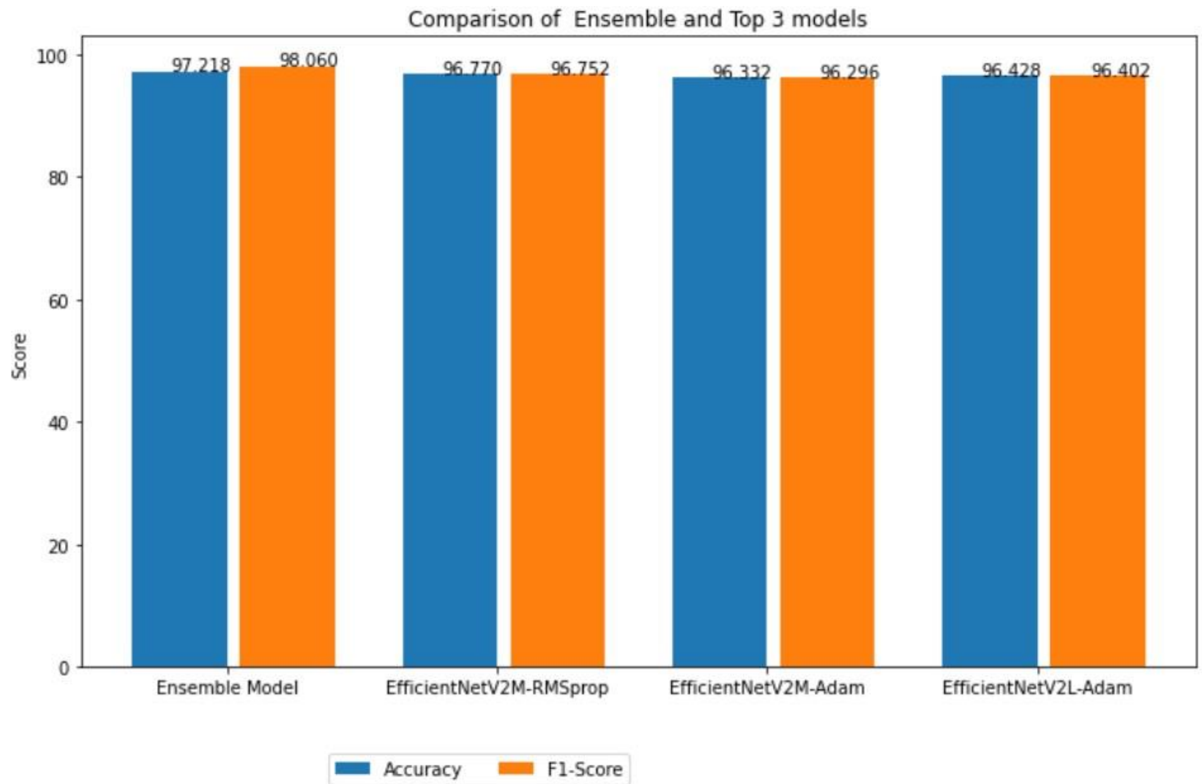


Figure 15.

We prioritise the weighted average F1-score due to the dataset's imbalance. As you can see, Ensemble models weighted average F1-score outperforms all other models. Also, all of the other metrics are found to be better.

6. Observations and Findings

- EffcientNetV2M obtained the highest accuracy with the RMS optimizer, with a weighted average F1-score of 96.752% and accuracy of 96.77%.
- The top three accuracies and weighted average F1 score were observed by EfficientNetV2L with Adam optimizer, EfficientNetV2M with RMSprop optimizer, and EfficientNetV2M with Adam optimizer.
- Weighted average F1-score and AUC score of all the models were good indicating that all the models performed well on the dataset.
- Accuracies obtained for the models with the Adam optimizer were comparatively higher.
- (For EffcientNetV2M and EffcientNetV2B2, RMSProp performed better). However, the differences were very low.
- The standard deviation observed for VGG 16 models was 2.3 with the RMSprop optimizer and 2.9 with the Adam optimizer making it an unstable model, whereas for the EfficientNetV2 models, SD was below 1. Exception models with the Adam optimizer had an SD of 1.4, whereas RMSprop produced an SD of 0.4.
- From the standard deviation, there is a trend that models with RMSprop are more stable as compared to models with Adam optimizer.
- EfficientNetV2 Models were found to perform better than the old models like VGG-16 and Xception.
- The Ensemble model generated an F1 score of 98.06% and the an accuracy of 97.218%, making it the best model. For the ensemble model, the first two folds had comparatively lower accuracy than one of its constituent models but performed better than all in the rest of the folds. However, for the F1-score, The Ensemble model performed better than all the folds.

7. Conclusion and Future Works

7.1 CONCLUSION AND FUTURE WORKS

The comparison between RMSprop and Adam optimizers was successfully made. We got an insight into how well the new EfficientNetV2 models performed and how they outperformed the old models. This work has several future scopes, and we have taken the ensemble from the top 3 performing models using the average method. Several other combinations of ensembling could be done, and performance analysis could be done on another dataset. Also, more learning could be done if the dataset were divided based on gender, as studies have shown that gender-related factors cannot be counted out [36].

Moving forward, there are several potential future works that could build upon the existing study. One possibility is to test the models on a larger dataset to improve their accuracy and robustness. Another potential avenue for exploration is to compare other optimization algorithms such as Adagrad, Adadelta, and Nadam with Adam and RMSprop to determine which algorithm performs best for pneumonia detection. Additionally, To enhance the performance and accuracy of the deep learning models, other imaging methods like CT scans and MRI scans can be included. Finally, to determine their effectiveness in realworld clinical settings, the models could be tested in clinical decision support systems and integrated into radiologists' and doctors' workflows.

APPENDICES

APPENDIX 1: SOURCE CODE

A1.1 CLAHE CODE

```
i=1     for     img     in
train_pneumonia:    img  =
cv2.imread(img, 0)
 clahe = cv2.createCLAHE (clipLimit=3.0, tileGridSize=(8,8))   cl_img=clahe.apply(img)
 Ak=cv2.imwrite("/content/drive/MyDrive/Capstone_data/Clahe_train/pneumonia/"+str(
i)+".jpeg",cl_img)   i+=1

i=1 for img in test_normal:
img = cv2.imread(img, 0)
 clahe = cv2.createCLAHE (clipLimit=3.0, tileGridSize=(8,8))   cl_img=clahe.apply(img)
 Ak=cv2.imwrite("/content/drive/MyDrive/Capstone_data/Clahe_test/normal/"+str(i)+".j
peg",cl_img)   i+=1
```

A1.2 DATA AUGMENTATION

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator ,load_img,img_t
o_array
train_datagen = ImageDataGenerator(
rescale = 1./255,
shear_range=10,
zoom_range=0.3,
horizontal_flip=True,
vertical_flip=True,
brightness_range=[0.5,2.0],
width_shift_range = 0.2,
      rotation_range=20, fill_mode = 'nearest',)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

A1.3 RMSPROP OPTIMIZER CODE FOR ALL MODELS

```
VALIDATION_ACCURACY = []
```

```
VALIDATION_LOSS = []
TRAINING_ACCURACY=[]
TRAINING_LOSS=[]

save_dir = '/content/drive/MyDrive/Capstone_data/Clahe_train'
fold_var=1 fine_tune_at=30 t=df_train[['class']] for train_index,
val_index in skf.split(np.zeros(len(Y)),Y):
  print("K-Fold"+str(fold_var))
training_data = df_train.iloc[train_index]
  validation_data = df_train.iloc[val_index]

  train_data_generator = train_datagen.flow_from_dataframe(dataframe=training_data,x
_col="image",y_col="class",target_size=(224,224),
                              batch_size=16,class_mode='categorical',shuffle=True)
valid_data_generator  = test_datagen.flow_from_dataframe(dataframe=validation_data
,x_col="image",y_col="class",target_size=(224,224),
                              batch_size=16,class_mode='categorical',shuffle=False)

  # CREATE NEW MODEL
  model = Sequential()
  conv_base = EfficientNetV2L(input_shape=[224,224]+[3], include_top=False, weights
='imagenet')                 for     layer     in
conv_base.layers[:fine_tune_at]:
      layer.trainable  =  False          x=
conv_base.output                    x    =
GlobalAveragePooling2D()(x)       x =
BatchNormalization()(x)            x   =
Dropout(0.3)(x)        x  =  Dense(512,
activation   ='relu')(x)             x   =
BatchNormalization()(x)            x   =
Dropout(0.5)(x)
  x    =    Dense(2,    activation    ='sigmoid')(x)
model = Model(conv_base.input, x)
  reduce_lr = ReduceLROnPlateau(monitor='val_loss',patience=7)
  early_stopping = EarlyStopping(monitor = "val_accuracy",patience = 9,verbose = 1,m
ode = "max",)      checkpoint = ModelCheckpoint(monitor = "val_accuracy",filepath =
"/content/drive/M yDrive/Capstone_data/crefv2lmnt_rms"+str(fold_var)+".h5",verbose =
1,save_best_only
= True, )
  opt = keras.optimizers.RMSprop(lr=0.0001)
```

```
model.compile(loss = "binary_crossentropy", optimizer =opt, metrics = "accuracy")
s1=train_data_generator.n//train_data_generator.batch_size
s2=valid_data_generator.n//valid_data_generator.batch_size
lr_sched = LearningRateScheduler(lambda epoch: 1e-4 * (0.75 ** np.floor(epoch / 2)))
history = model.fit(train_data_generator,    epochs = 30,     batch_size = 16,
validation_data = valid_data_generator,
validation_steps = s2,    steps_per_epoch
= s1,
callbacks    =    [reduce_lr,    early_stopping,    checkpoint,lr_sched])
printHistory(history,"EfficientNetV2L_RMSprop",early_stopping.stopped_epoch+1)
modela  =  load_model("/content/drive/MyDrive/Capstone_data/crefv2lmnt_rms"+str(fo
ld_var)+".h5")

results = modela.evaluate(valid_data_generator)    results
=  dict(zip(model.metrics_names,results))        Y_pred  =
modela.predict_generator(valid_data_generator)    y_pred =
np.argmax(Y_pred,   axis=1)              target_names   =
['Normal','Pneumonia']
cm = confusion_matrix(valid_data_generator.classes, y_pred)
conf(cm,target_names)    print('Classification Report')
print(classification_report(valid_data_generator.classes, y_pred, target_names=target_
names))              auc    =    roc_auc_score(valid_data_generator.classes,    y_pred)
plot_roc_curve(valid_data_generator.classes, y_pred,auc)
TRAINING_ACCURACY.append(history.history['accuracy'])
TRAINING_LOSS.append(history.history['loss'])
VALIDATION_ACCURACY.append(history.history['val_accuracy'])
VALIDATION_LOSS.append(history.history['val_loss'])
fold_var += 1
```

For other models change the model name in the model name part.

A1.4 ADAM OPTIMIZER CODE FOR ALL MODELS

```
VALIDATION_ACCURACY = []
VALIDATION_LOSS = []
TRAINING_ACCURACY=[]
TRAINING_LOSS=[]

save_dir = '/content/drive/MyDrive/Capstone_data/Clahe_train'
fold_var=1 fine_tune_at=30 t=df_train[['class']] for train_index,
val_index  in  skf.split(np.zeros(len(Y)),Y):      print("K-
```

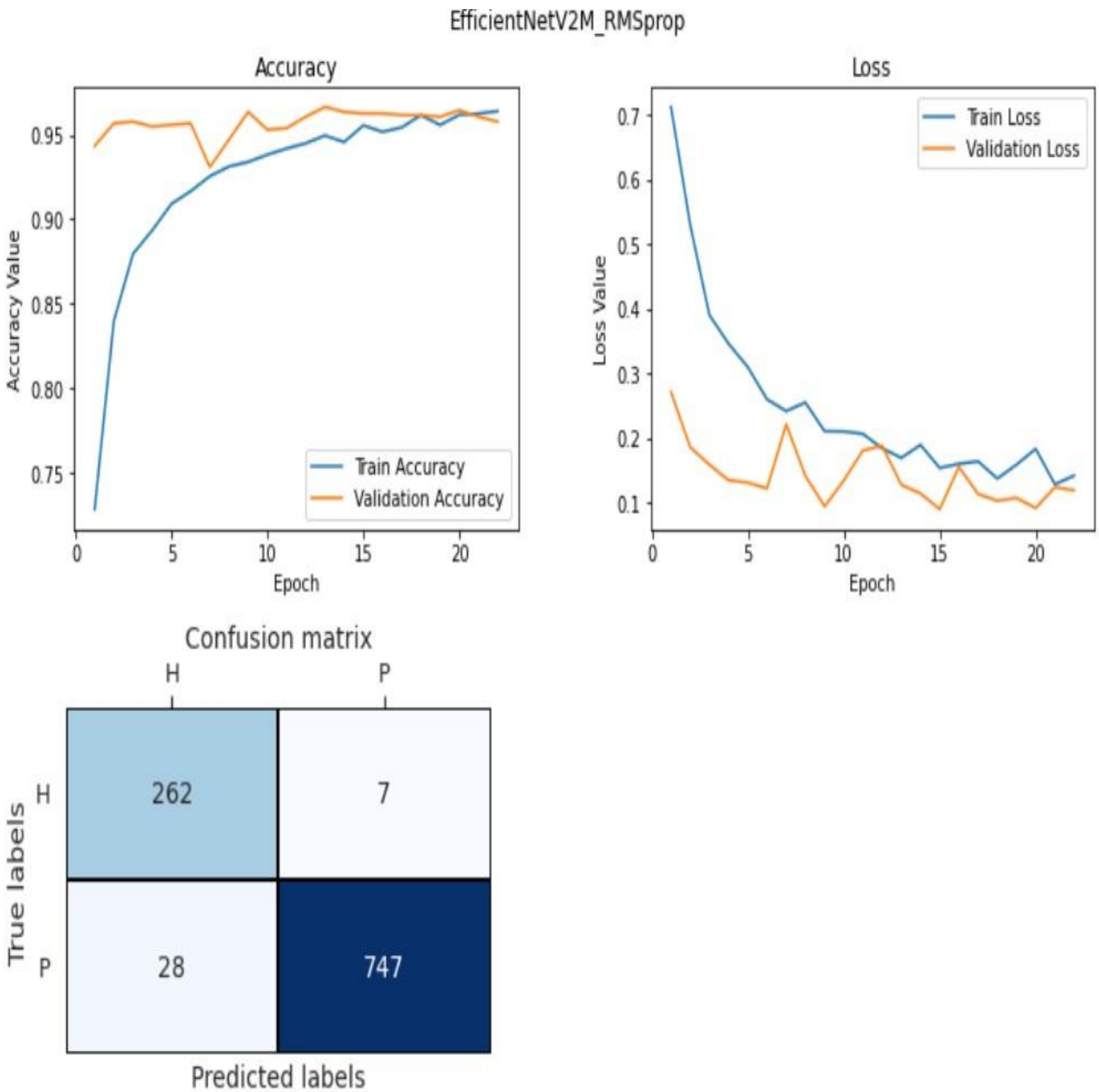
```
Fold"+str(fold_var))                              training_data       =
df_train.iloc[train_index]
   validation_data = df_train.iloc[val_index]

   train_data_generator = train_datagen.flow_from_dataframe(dataframe=training_data,x
_col="image",y_col="class",target_size=(224,224),
                                    batch_size=16,class_mode='categorical',shuffle=True)
valid_data_generator  = test_datagen.flow_from_dataframe(dataframe=validation_data
,x_col="image",y_col="class",target_size=(224,224),
                                    batch_size=16,class_mode='categorical',shuffle=False)

   # CREATE NEW MODEL
   model = Sequential()
   conv_base = EfficientNetV2L(input_shape=[224,224]+[3], include_top=False, weights
='imagenet')                        for     layer     in
conv_base.layers[:fine_tune_at]:
      layer.trainable  =  False         x=
conv_base.output                  x    =
GlobalAveragePooling2D()(x)     x =
BatchNormalization()(x)           x  =
Dropout(0.3)(x)
   x = Dense(512, activation ='relu')(x)
x  =  BatchNormalization()(x)        x  =
Dropout(0.5)(x)
   x    =    Dense(2,    activation    ='sigmoid')(x)
model = Model(conv_base.input, x)
   reduce_lr = ReduceLROnPlateau(monitor='val_loss',patience=8)
   early_stopping = EarlyStopping(monitor = "val_accuracy",patience = 9,verbose = 1,m
ode  =  "max",)       checkpoint  =  ModelCheckpoint(monitor  =  "val_accuracy",filepath  =
"/content/drive/M   yDrive/Capstone_data/crefv2lmnt_adam"+str(fold_var)+".h5",verbose
= 1,save_best_onl y = True, )
   opt = keras.optimizers.Adam(lr=0.0001)
   model.compile(loss  =  "binary_crossentropy",  optimizer  =opt,  metrics  =  "accuracy")
s1=train_data_generator.n//train_data_generator.batch_size
s2=valid_data_generator.n//valid_data_generator.batch_size
   lr_sched = LearningRateScheduler(lambda epoch: 1e-4 * (0.75 ** np.floor(epoch / 2)))
history = model.fit(train_data_generator,     epochs = 30,      batch_size = 16,
   validation_data = valid_data_generator,
validation_steps = s2,      steps_per_epoch
= s1,
   callbacks          =          [reduce_lr,          early_stopping,          checkpoint,lr_sched])
printHistory(history,"EfficientNetV2L_Adam",early_stopping.stopped_epoch+1)
modela    =    load_model("/content/drive/MyDrive/Capstone_data/crefv2lmnt_adam"+str(
fold_var)+".h5")
```

```
results = modela.evaluate(valid_data_generator)     results
=  dict(zip(model.metrics_names,results))          Y_pred  =
modela.predict_generator(valid_data_generator)     y_pred =
np.argmax(Y_pred,    axis=1)                  target_names    =
['Normal','Pneumonia']
cm = confusion_matrix(valid_data_generator.classes, y_pred)
conf(cm,target_names)     print('Classification Report')
print(classification_report(valid_data_generator.classes, y_pred, target_names=target_
names))                  auc    =    roc_auc_score(valid_data_generator.classes,    y_pred)
plot_roc_curve(valid_data_generator.classes, y_pred,auc)
TRAINING_ACCURACY.append(history.history['accuracy'])
TRAINING_LOSS.append(history.history['loss'])
VALIDATION_ACCURACY.append(history.history['val_accuracy'])
VALIDATION_LOSS.append(history.history['val_loss'])
fold_var += 1
```

For other models change the model name in the model name part.

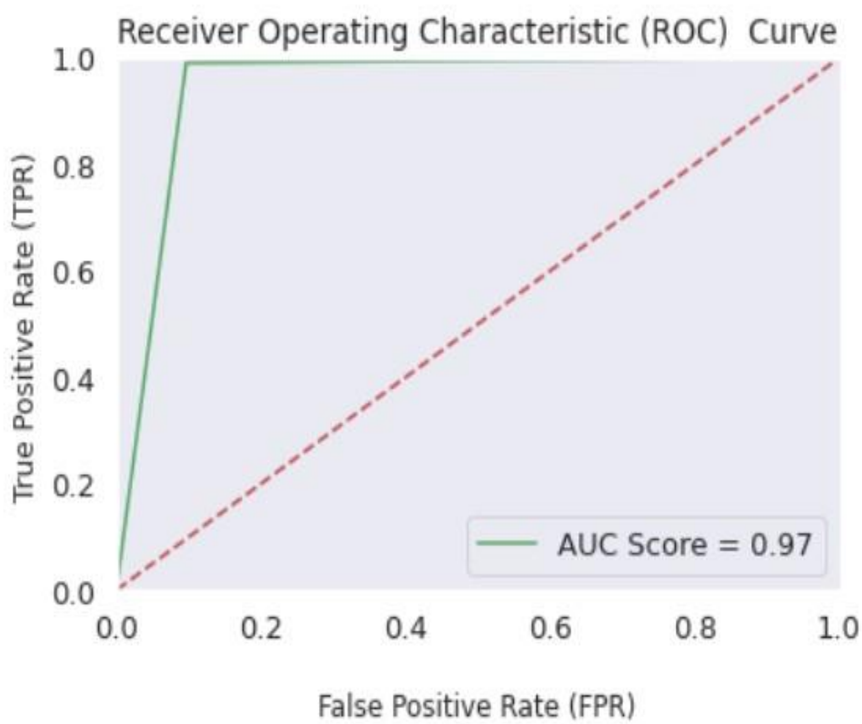
Appendix 2: Sample Output

A1.1 EFFICIETNETV2M-RMSPROP RESULTS



Classification Report

	precision	recall	f1-score	support
Normal	0.90	0.97	0.94	269
Pneumonia	0.99	0.96	0.98	775
accuracy			0.97	1044
macro avg	0.95	0.97	0.96	1044
weighted avg	0.97	0.97	0.97	1044



REFERENCES

[1] Pneumonia.Available online: https://www.who.int/news-room/factsheets/detail/pneumonia

[2] Pneumonia.Available online: https://www.radiologyinfo.org/en/info/pneumonia

[3] Pneumonia.Available online: https://www.uptodate.com/contents/clinical-evaluationand-diagnostic-testing-for-community-acquired-pneumonia-in-adults

[4] Hashmi, M. F., Katiyar, S., Keskar, A. G., Bokde, N. D., & Geem, Z. W. (2020). Efficient pneumonia detection in chest xray images using deep transfer learning. *Diagnostics*, *10*(6), 417

[5] Pneumonia.Available online: https://www.nhs.uk/conditions/pneumonia/treatment/#:~:text=Mild%20pneumonia%20ca n%20usually%20be,cases%20may%20need%20hospital%20treatment.

[6] Mabrouk, A., Díaz Redondo, R. P., Dahou, A., Abd Elaziz, M., & Kayed, M. (2022). Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. *Applied Sciences*, *12*(13), 6448.

[7] Kallander, K., Burgess, D. H., & Qazi, S. A. (2016). Early identification and treatment of pneumonia: a call to action. *The Lancet. Global Health*, *4*(1), e12.

[8] Pajuelo, M. J., Anticona Huaynate, C., Correa, M., Mayta Malpartida, H., Ramal Asayag, C., Seminario, J. R., ... & Paz-Soldan, V. A. (2018). Delays in seeking and receiving health care services for pneumonia in children under five in the Peruvian Amazon: a mixed-methods study on caregivers' perceptions. *BMC health services research*, *18*(1), 1-11.

[9] Peters, M. E., Ruder, S., & Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

[10] Kundu, R., Das, R., Geem, Z. W., Han, G. T., & Sarkar, R. (2021). Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *Plos one*, *16*(9), e0256630.

[11] Ayan, E., Karabulut, B., & Ünver, H. M. (2022). Diagnosis of pediatric pneumonia with ensemble of deep convolutional neural networks in chest x-ray images. Arabian Journal for Science and Engineering, 47(2), 2123-2139.

[12] El Asnaoui, K. (2021). Design ensemble deep learning model for pneumonia disease classification. International Journal of Multimedia Information Retrieval, 10(1), 55-68. [13] Sourab, S. Y., & Kabir, M. A. (2022). A comparison of hybrid deep learning models for pneumonia diagnosis from chest radiograms. Sensors International, 3, 100167.

[14] Almaslukh, B. (2021). A lightweight deep learning-based pneumonia detection approach for energy-efficient medical systems. Wireless Communications and Mobile Computing, 2021.

[15] Ukwuoma, C. C., Qin, Z., Heyat, M. B. B., Akhtar, F., Bamisile, O., Muaad, A. Y., ... & Al-Antari, M. A. (2022). A hybrid explainable ensemble transformer encoder for pneumonia identification from chest X-ray images. Journal of Advanced Research.

[16] Alhudhaif, A., Polat, K., & Karaman, O. (2021). Determination of COVID-19 pneumonia based on generalized convolutional neural network model from chest X-ray images. Expert Systems with Applications, 180, 115141.

[17] Zhang, D., Ren, F., Li, Y., Na, L., & Ma, Y. (2021). Pneumonia detection from chest X-ray images based on convolutional neural network. Electronics, 10(13), 1512.

[18] Ikechukwu, A. V., Murali, S., Deepu, R., & Shivamurthy, R. C. (2021). ResNet-50 vs VGG-19 vs training from scratch: a comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images. Global Transitions Proceedings, 2(2), 375-38.

[19] Avola, D., Bacciu, A., Cinque, L., Fagioli, A., Marini, M. R., & Taiello, R. (2022). Study on transfer learning capabilities for pneumonia classification in chest-x-rays images. Computer Methods and Programs in Biomedicine, 221, 106833.

[20] Rajasenbagam, T., Jeyanthi, S., & Pandian, J. A. (2021). Detection of pneumonia infection in lungs from chest X-ray images using deep convolutional neural network and content-based image retrieval techniques. Journal of Ambient Intelligence and Humanized Computing, 1-8.

[21] Mujahid, M., Rustam, F., Álvarez, R., Luis Vidal Mazón, J., Díez, I. D. L. T., & Ashraf, I. (2022). Pneumonia Classification from X-ray Images with Inception-V3 and Convolutional Neural Network. Diagnostics, 12(5), 1280.

[22] Ravi, V., Acharya, V., & Alazab, M. (2022). A multichannel EfficientNet deep learning-based stacking ensemble approach for lung disease detection using chest X-ray images. Cluster Computing, 1-23.

[23] Manickam, A., Jiang, J., Zhou, Y., Sagar, A., Soundrapandiyan, R., & Samuel, R. D. J. (2021). Automated pneumonia detection on chest X-ray images: A deep learning

approach with different optimizers and transfer learning architectures. Measurement, 184, 109953.

[24] Ibrahim, D. M., Elshennawy, N. M., & Sarhan, A. M. (2021). Deep-chest: Multiclassification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases. Computers in biology and medicine, 132, 104348.

[25] Huang, M. L., & Liao, Y. C. (2022). A lightweight CNN-based network on COVID19 detection using X-ray and CT images. Computers in Biology and Medicine, 105604.

[26] Pneumonia.Available online: https://www.reuters.com/article/health-pneumoniaidINL4N0AY04820130129

[27] Salehi, M., Mohammadi, R., Ghaffari, H., Sadighi, N., & Reiazi, R. (2021). Automated detection of pneumonia cases using deep transfer learning with paediatric chest X-ray images. *The British journal of radiology*, *94*(1121), 20201263.

[28] Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Kashem, S. B. A., ... & Chowdhury, M. E. (2021). Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images. *Computers in biology and medicine*, *132*, 104319.

[29] Habib, N., Hasan, M. M., Reza, M. M., & Rahman, M. M. (2020). Ensemble of CheXNet and VGG-19 feature extractor with random forest classifier for pediatric pneumonia detection. *SN Computer Science*, *1*, 1-9.

[30] Rubini C, Pavithra N (2019) Contrast enhancement of MRI images using AHE and CLAHE techniques. Int J Innov Technol Explor Eng 9(2):2442–2445.

[31] https://keras.io/api/applications/

[32] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[33] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251-1258).

[34] https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41

[35] Tan, M., & Le, Q. (2021, July). Efficientnetv2: Smaller models and faster training. In *International conference on machine learning* (pp. 10096-10106). PMLR.

[36] Corica, B., Tartaglia, F., D’Amico, T., Romiti, G. F., & Cangemi, R. (2022). Sex and gender differences in community-acquired pneumonia. *Internal and Emergency Medicine*, *17*(6), 1575-1588.