Comparative analysis of Pneumonia Detection from Chest X-rays

APPLIED AI ASSESSMENT

ABSTRACT

Every 39 seconds, a child dies of Pneumonia, and in the world's most challenging countries, most vulnerable and poor children are more likely to suffer this disease. In 2018, Nigeria, my home country, had the most considerable child deaths caused by Pneumonia, and this disease is often easily preventable. In this study, I propose developing four convolutional layers' convolutional neural network (CNN) model architecture for detecting Pneumonia from chest X-rays. In addition, I compared the effect of convolutional layers and optimizers' hyperparameters, with four convolutional layers and Adam optimizer, the proposed model outperforming all other hyperparameters with 90% recall, 90% precision, 90% f1-score, and 91% accuracy. Six convolutional layers achieved 77% precision, 64% recall, 64% f1-score, and 72% accuracy, and Adagrad obtaining 82% precision, 54% recall, 47% f1-score, and 66% accuracy, were the least performing in terms of hyperparameters tuning. Finally, compared with four pretrained networks, the proposed model achieved second best, with the VGG-19 model edging out by 3.22% with 93% precision, 90% recall, 90% f1-score, and 91% accuracy.

1.0 INTRODUCTION

Pneumonia is a lung inflammatory usually induced by an infection. Older people, babies, and people with lung or heart issues risk becoming seriously ill and may require treatment (NHS, 2023). Annually, over 700,000 deaths occur due to pneumonia being the leading infectious disease killer in children, according to UNICEF (2023). Zhang et al. (2021) highlighted the increasing pneumonia mortality rates as a factor prompting the development of more effective and rapid detection methods, including radiology-based procedures.

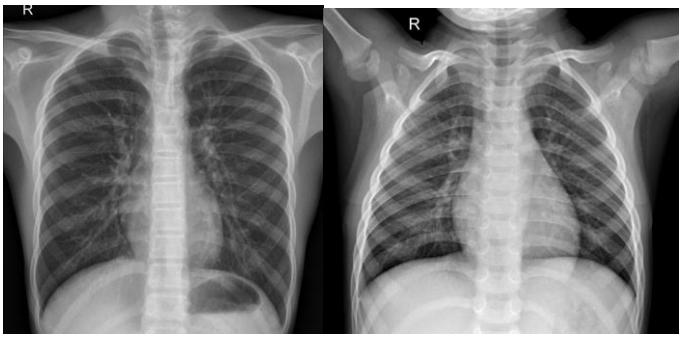

Fig. 1.1: X-ray images with normal chest samples

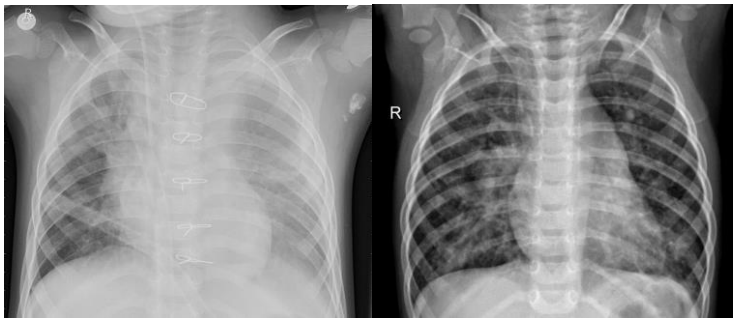

Fig. 1.2: X-ray images with pneumonia chest samples

Due to recent developments in the fields of Artificial Intelligence (AI), and Digital Image Processing (DIP), researchers can now employ machines to develop Computer-Aided Diagnosis (CAD) procedures to detect Pneumonia by analyzing chest X-ray images. Convolutional neural networks (CNN) are highly efficient for extracting features from medical images, but they require large amount of data (Krishnamurthy et al., 2021). Kundu et al. (2021) concluded that transfer learning addresses this issue by reusing models trained on large datasets, such as ImageNet, which has over 14 million images.

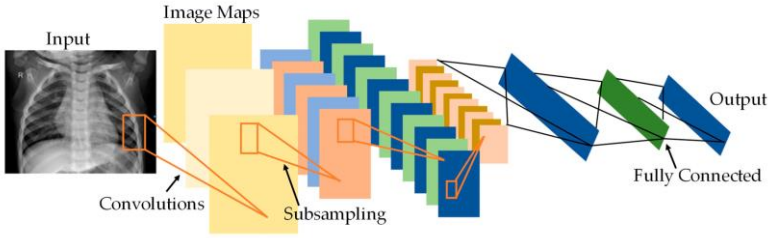

Fig. 1.3: A CNN architecture of chest X-ray image

In this study, I aim to develop a deep learning system to detect pneumonia and compare it against four pretrained networks. Is the proposed model better than the pretrained networks in

pneumonia detection? What are the effects of convolutional layers and optimizers? The outcome of this study is to showcase the importance of deep learning models in detecting pneumonia while comparing with existing pretrained networks.

2.0 BACKGROUND OF STUDY

The existing studies proposed deep learning methods such as ResNet50, MobileNetV2, and many more for pneumonia detection. However, only a few could explain the effect of convolutional layers and optimizers, which aids performance optimization.

2.1 RELATED WORKS

Krishnamurthy et al. (2021) applied fine-tuning and data augmentation on pre-trained model and observed that the DenseNet201 model outperformed other models.

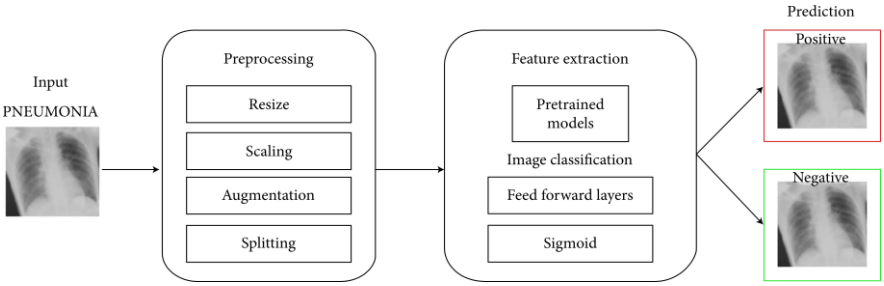

Fig. 2.1: A pipeline of the DNN models for diagnosing Pneumonitis.

Ikechukwu et al. (2021) revealed that sufficient fine-tuning on pretrained models were comparable to Iyke-Net, a CNN developed from scratch, with a recall of 92.03%. Figure 2.2 is a block diagram of the Iyke-Net model architecture.

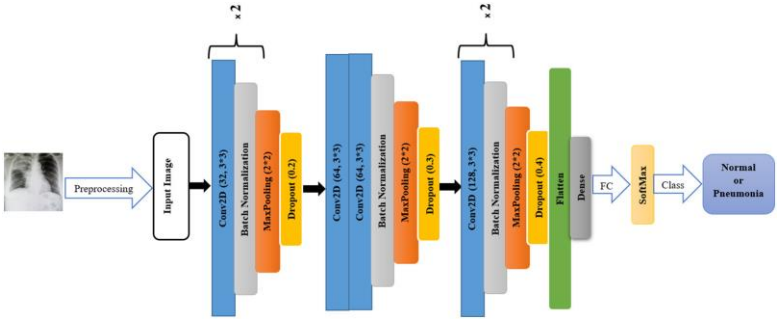

Fig. 2.2: Block diagram of Iyke-Net

Chouhan et al. (2020) suggested an ensemble of pretrained models that outperform individual models for detection and feature extraction and achieved a state-of-the-art performance. Kundu et al. (2021) designed an ensemble of pretrained networks that performed better than most ensemble techniques and employed deep learning to handle limited available data. ŞAHİN et al. (2021) evaluated four models by comparing pretrained networks and a recommended CNN model according to their performance, with the ResNet model being the best.

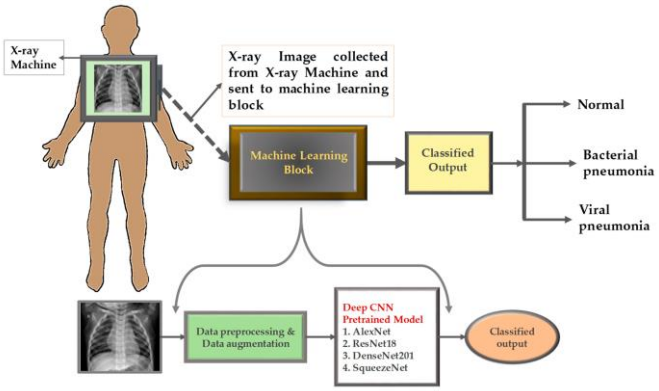

Fig. 2.3: Overview of the methodology

Mabrouk et al. (2022) proposed using an Ensemble Learning (EL) of pretrained models to diagnose pneumonia on chest X-ray images by using fine-tuning. Jain et al. (2020) compared two CNN models with pretrained models to classify X-ray images into binary classes by tuning hyperparameters.

3.0 OBJECTIVES

The objectives of this work are as follows:

- To detect chest X-ray images into normal and pneumonia using a CNN model as a reference.
- To investigate and compare the performance of pretrained networks such as MobileNetV2, DenseNet201, VGG-19, and ResNet50 in pneumonia detection.
- To analyze the effect of convolutional layers and optimizers for performance optimization.

4.0 METHODOLOGY

This study explored the following three (3) methods for detecting pneumonia and will be discussed extensively under this section.

1. Convolutional Neural Network model – Proposed Model
2. Hyperparameters tuning such as Convolutional layers and Optimizers.
3. Pretrained Networks

4.1 Convolutional Neural Network (CNN) – Proposed Model

Deep learning models in computer vision utilizes CNN and its application includes image processing, voice recognition, and pattern recognition. The proposed model is discussed in this section and Figure 4.1 shows the architecture of the proposed CNN model.

The model contains four 3x3 kernel convolutional layers and one fully connected layer, and the activation function is Rectified Linear Unit (ReLu), with each convolutional layer having a default max-pooling size. The convolutional layer retrieves feature from images while maintaining the spatial relationship between image pixels intact using filters. The fully connected layer classifies the returned images' convolved features into respective classes (Ikechukwu et al., 2021). According to Chouhan et al. (2020), Pooling helps reduce the spatial size of representation generated by former kernels after convolution and extracting dominant features that are rotational and positional invariant — Max-pooling outputs the maximum value

of separated input squares of a given size. A sigmoid function closely follows the fully connected layer (or dense layer) and helps differentiate whether a patient has pneumonia. The sigmoid function's ability to convert input into probability during binary classification makes it ideal for this task. In addition, its non-linearity and differentiability make it work with many optimization algorithms (Saturn Cloud, 2023).

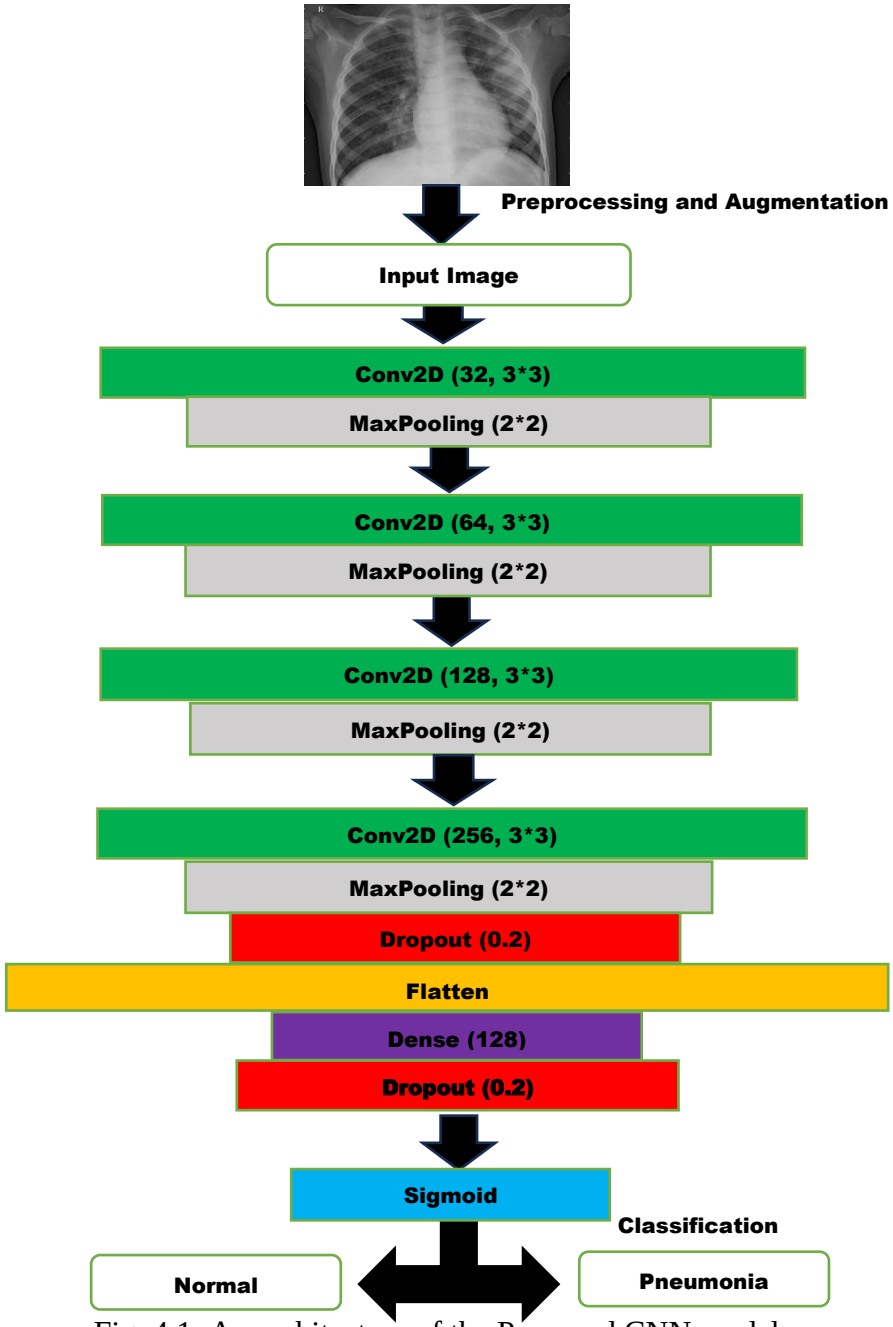

Fig. 4.1: An architecture of the Proposed CNN model

4.2 Hyperparameter tuning

The two hyperparameters focused on in this study are convolutional layers and optimizers. I plan to study the effects of tuning these hyperparameters for performance optimization in detecting pneumonia and compare them with the proposed model.

Convolutional layers: Zhou et al. (2022) defines this as a hidden layer that contains convolution units in a CNN for feature extraction from images. This study will employ four convolutional layers in the proposed model while comparing against three, five, and six convolutional layers.

Optimizers: Gupta (2021) defines optimizer algorithms as the method that improves the performance of deep learning models. They affect the speed and accuracy of training these models. In this study, Adam will serve as the proposed model optimizer while comparing against Stochastic Gradient Descent, RMSprop and Adagrad.

4.3 Pretrained Networks

This study will employ MobileNetV2, ResNet50, DenseNet201, and VGG-19 as pretrained networks, and they were all chosen based on existing works and yielded the best recall and accuracy values. In addition, I will compare my proposed model with these networks.

4.3.1 MobileNetV2

According to Tsang (2019) and Sandler (2019), there exist two block types for this model. One block has stride of 2 for downsizing and one is residual block with stride of 1, and both blocks have 3 layers. The 1st layer is 1x1 convolution with ReLU6, 2nd layer is depth wise convolution and the 3rd layer is a linear 1x1 convolution as shown in Table 4.1 and takes input size 224x224.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 4.1: Bottleneck residual block

Table 4.2 shows the architect of MobileNetV2.

Where, h: height, t:expansion factor, w: width, s: stride, k: kernel size, c: number of output channels, and n: repeating number.

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 4.2: Overall architecture of MobileNetV2

4.3.2 ResNet50

According to Murkherjee (2022), Residual Networks (ResNet) was first introduced in a paper by He et. al. in 2015 and was required due to pitfalls in modern networks at the time. The ResNet paper made popular the use of skip connection approaches in performing mapping that improves model accuracy. ResNet50 or Residual Network-50 is a CNN model that is 50 layers deep with 224 * 224 as its image input size. Figure 4.2 shows an architecture of a ResNet-50 model (Mukherjee, 2022).

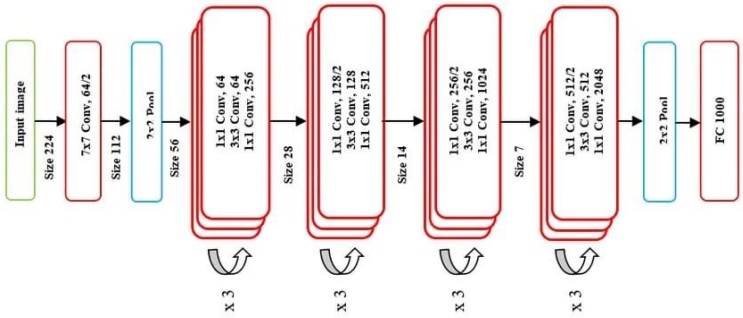

Fig. 4.2: An architecture of ResNet-50

4.3.3 DenseNet201

According to Aslan (2022), he highlights that DenseNet works in a feed-forward pattern by connecting each layer to every other layer. They encourage feature reuse, strengthen feature propagation, alleviate vanishing-gradient problems, and significantly reduce the number of parameters. DenseNet201, or Dense Convolutional Network, is a CNN model that is 201 layers deep with 224 * 224 as its image input size. Figure 4.3 is an architecture of a DenseNet201 model (Wang and Zhang, 2020).

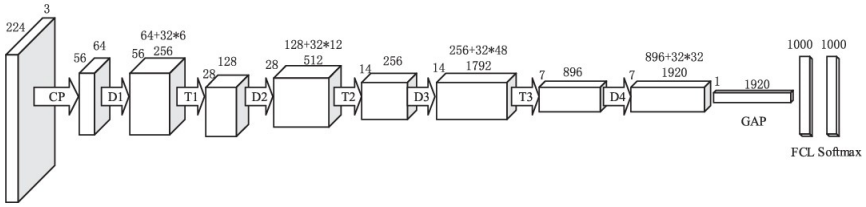

Fig. 4.3: An architecture of DenseNet201

4.3.4 VGG-19

Simonyan and Zisserman (2014) proposed the VGG network at the University of Oxford, UK. VGG-19 is a VGG model with 19 layers (16 convolution layers and 3 fully connected layers) and has an image input size of 224 * 224. The model promotes better feature extraction through its layers, which yielded a better performance as against other pretrained models and downsizing using Maxpooling. Figure 4.4 is an architecture of a VGG-19 model (Nguyen et al., 2022).

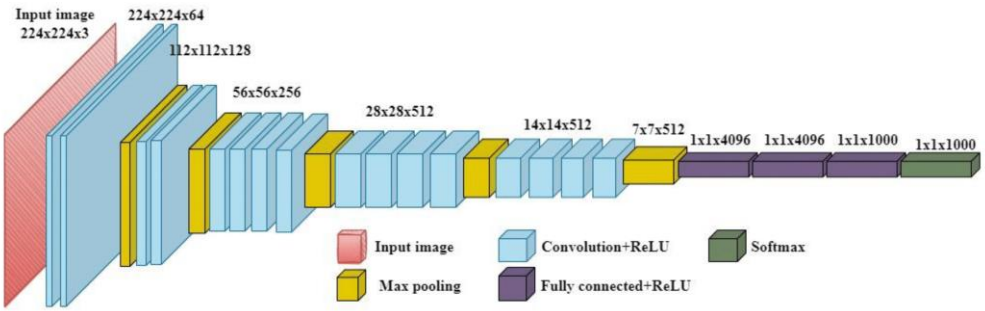

Fig. 4.4: An architecture of VGG-19

5.0 EXPERIMENTS

5.1 EXPERIMENTAL SETUP OF THE PROPOSED MODEL

Figure 5.1 shows the experimental setup of the proposed model I will employ in this study.

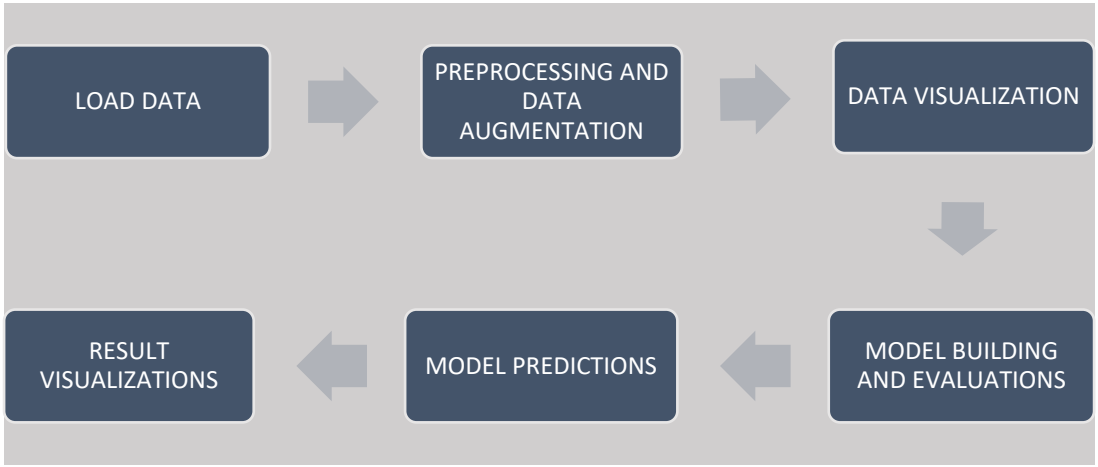

Fig. 5.1: Setup of the Proposed Models

5.2 EXPERIMENTAL SETUP OF THE PRETRAINED NETWORKS

Figure 5.2 shows the experimental setup of the pretrained networks I will employ in this study.

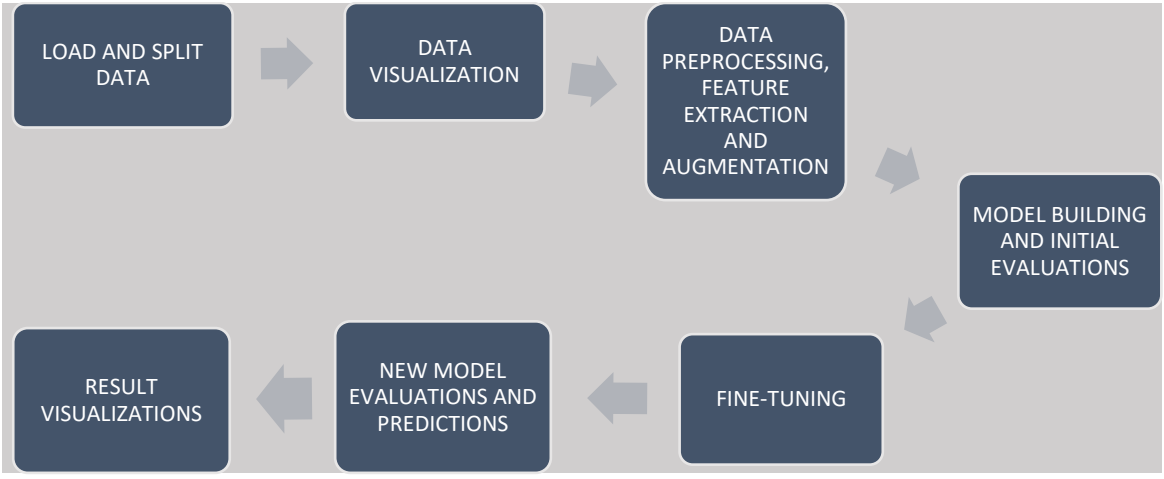

Fig. 5.2: Setup of the Pretrained Networks

Data Loading

This study uses Python's list directory (listdir) method to extract the list of files from a directory and the path library for handling the files in my operating system to load the dataset. In addition, I utilized Keras' Image dataset from directory in loading the datasets and splitting the training dataset into 80 percent as train and 20 percent as validation sets when training the pretrained networks' models.

Preprocessing and Data Augmentation

This section includes resizing, rescaling, splitting, and data augmenting. The input images were assigned a fixed size of image height and width as 224, respectively. The pixel images' scale was resized to 0 and 1 by dividing with the image pixel maximum value of 255. Figure 5.3 shows samples from the dataset after applying data augmentation.

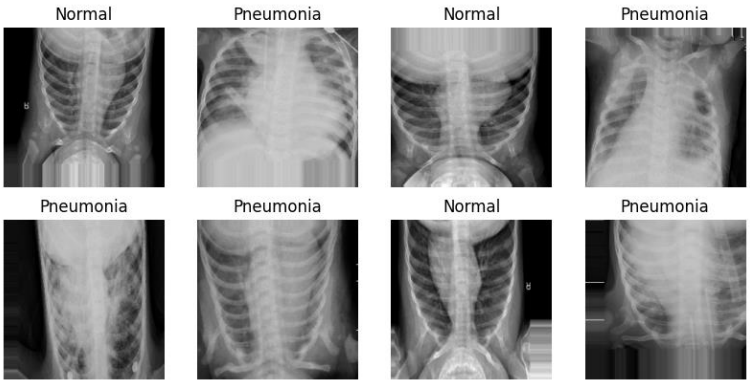

Fig. 5.3: Training Samples after Data Augmentation for the Proposed model

Data Augmentation helps when treating small image dataset, by artificially introducing diverse samples by transforming the training images. In this study, I implemented data augmentation and splitting using Keras library's Image Data Generator by tweaking the following: horizontal and vertical flip to flip in both directions randomly, 20 percent as validation sample, width and height shift ranges as 0.2 percent, shear and zoom ranges as 0.2, fill mode as nearest and from TensorFlow (2023), nearest is assigned to points outside the boundaries of the input.

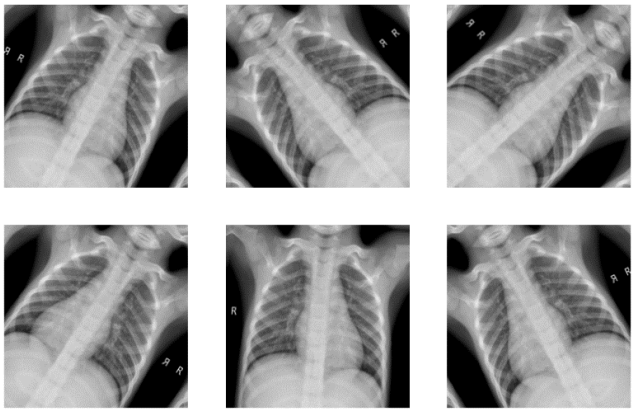

Fig. 5.4: Data Augmentation for the Pretrained Networks

For the pretrained networks, the preprocessing includes dataset loading, dataset configuration for performance using buffered prefetching, and rescaling pixel values by dividing the input values by 127.5 to the range of -1 to 1; I augmented the training set using Keras' Sequential library by randomly flipping horizontally and randomly rotating by 0.2 (TensorFlow, 2023).

Feature Extraction

In this study, I will employ feature extraction using the bottleneck layer of each pretrained network, as it retains more generality than the final/top layer according to TensorFlow (2023) by specifying include-top as False before freezing the created convolutional base by setting layer trainable as False.

Data Visualization

Visualization was carried out using matplotlib and Figure 3.5 shows the classes in the training and test datasets.

Model Building and Evaluations

Researchers best evaluate image classification models using the following metrics: Recall, Accuracy (Acc), Precision, and F1-Score. These metrics will form the basis for the performance evaluation of models in the above methodology.

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

where, TP = True Positive, TN = True Negative, FP = False Negative, and FN = False Negative

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

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

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Fine-tuning

Fine-tuning aims at ensuring specialized features adapt when working with new dataset instead of overwriting the generic learning (TensorFlow, 2023). This is achieved by unfreezing the top layers of the model initially frozen during feature extraction.

Result Visualizations

This study will employ matplotlib, seaborn, classification report and confusion matrix in computing and comparing results of each models.

Classification Report in machine learning is a performance evaluation metric showing F1-score, recall, precision and support (Kharwal, 2021).

Confusion Matrix is a matrix that visualizes classification performances.

5.3 HYPERPARAMETERS FOR THE MODELS

This study will utilize the following hyperparameters for the proposed model:

- No of Convolutional blocks = 4
- Optimizer = Adam
- Learning Rate = 0.001
- Batch size = 32
- No of epochs = 20
- Patience = 3
- 2 Dropouts = 0.20

Others include:

- Kernel size = (3,3)
- Filters = 32, 64, 128, 256
- Pool size = (2,2)

For the Pretrained Networks, I changed the following hyperparameters due to time constraints and optimizing performance.

- Learning Rate = 0.0001
- No of epochs = 10
- Fine-tuning no of epochs = 10

5.4 DATASET

Initially grouped into test, train and validation, merging the train and validation folders was necessary for better apportioning during training. The dataset compromises 5,856 Chest X-ray images grouped into Normal and Pneumonia.

The dataset was provided by Kermany et al. (2018) and based on chest X-ray scan images aged one to five years old pediatric patients at the Guangzhou Women and Children's Medical Center. The dataset is publicly available on Kaggle and was downloaded on 15 July 2023.

The merged dataset contains 5,232 training and 624 test samples. The test set comprises 234 normal and 390 pneumonia patients, while the training set comprises 3,883 pneumonia and 1349 normal patients. Figure 5.5 shows Normal and Pneumonia patients' chest X-ray scans from the training and test datasets.

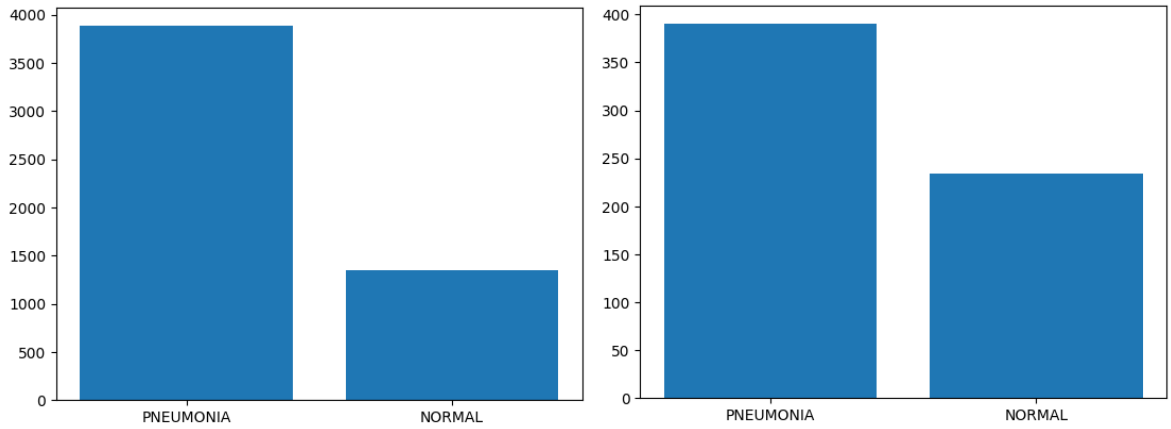

Fig. 5.5: Training and test datasets classification

System Specifications: 16.0GB RAM Intel(R) Core(TM) i7-9750H CPU@2.60GHz 2.59 GHz

6.0 RESULTS

6.1 RESULT OF THE PROPOSED MODEL

Model	Precision	Recall	F1 – Score	Accuracy
Proposed Model	0.90	0.90	0.90	0.91

Table 6.1: Result of the Proposed model

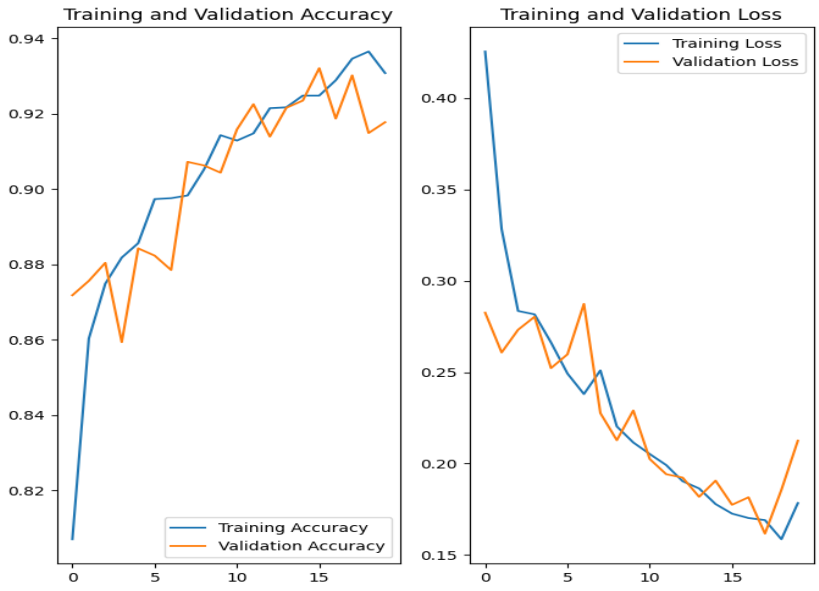

Fig. 6.1: Plots of Training and Validation Accuracy and Loss

	precision	recall	f1-score	support
0	0.89	0.86	0.87	234
1	0.92	0.93	0.93	390
accuracy			0.91	624
macro avg	0.90	0.90	0.90	624
weighted avg	0.91	0.91	0.91	624

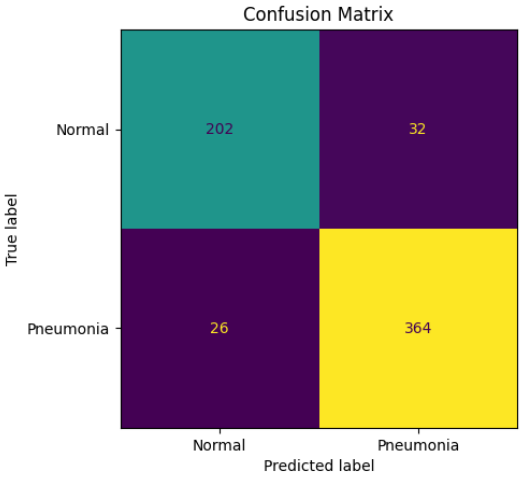

Fig. 6.2: Classification report and confusion matrix

Figures 6.1 and 6.2 in this section show the relationship between training and validation loss and accuracy, and the proposed model's classification report and confusion matrix.

6.2 EFFECTS OF HYPERPARAMETERS TUNING

6.2.1 CONVOLUTIONAL LAYERS

Model	Precision	Recall	F1 – Score	Accuracy
Proposed Model (4)	0.90	0.90	0.90	0.91
3	0.86	0.78	0.80	0.83
5	0.84	0.77	0.78	0.81
6	0.77	0.64	0.64	0.72

Table 6.2: Comparison of different models with different convolutional layers

In this section, Figures 6.3, 6.4, 6.5, 6.6, 6.7, and 6.8 show the relationship between training and validation loss and the accuracy, and the classification reports, confusion matrices of models with three, five, and six convolutional layers, respectively. Figure 6.9 shows the relationship plots of the models against the evaluation metrics in this study.

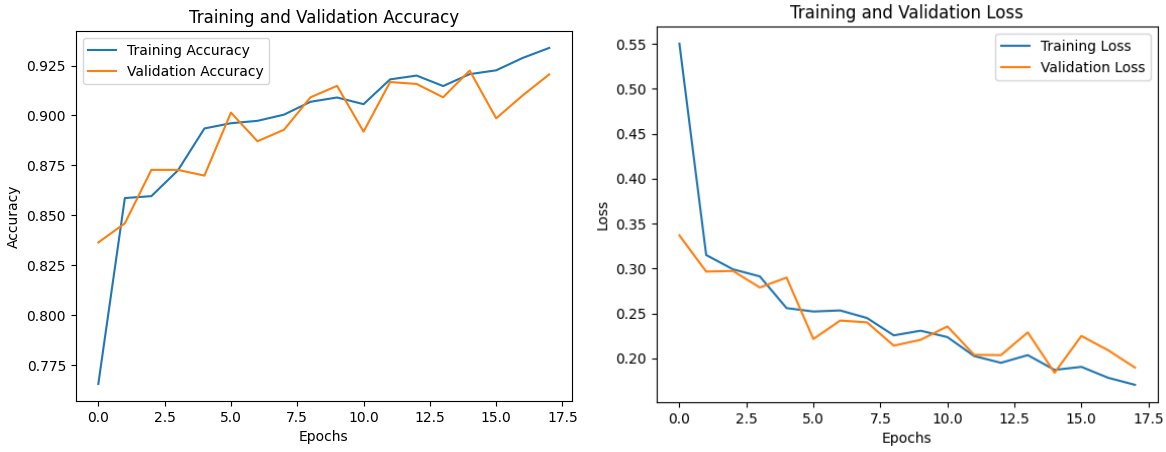

Fig. 6.3: Three convolutional layers accuracy and loss relationship plots

	precision	recall	f1-score	support
0	0.91	0.59	0.72	234
1	0.80	0.97	0.87	390
accuracy			0.83	624
macro avg	0.86	0.78	0.80	624
weighted avg	0.84	0.83	0.82	624

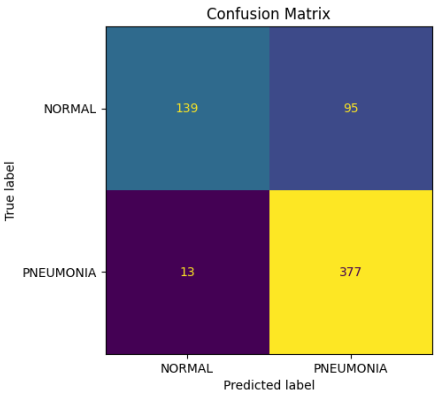

Fig. 6.4: Classification report and confusion matrix

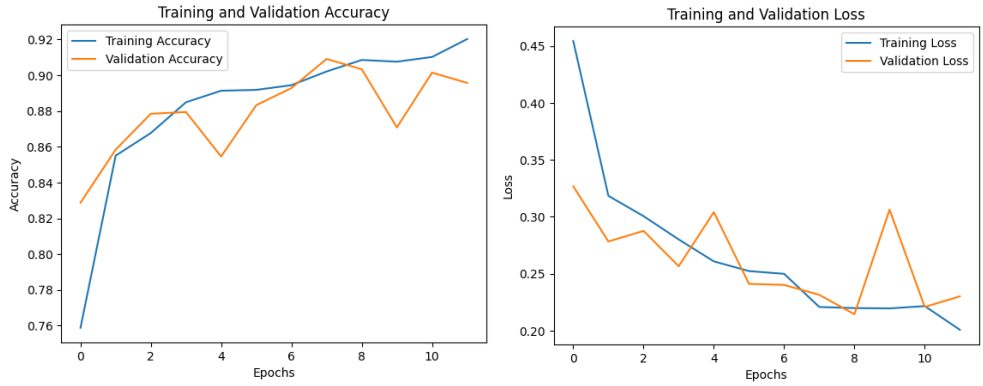

Fig. 6.5: Five convolutional layers accuracy and loss relationship plots

	precision	recall	f1-score	support
0	0.89	0.58	0.70	234
1	0.79	0.96	0.87	390
accuracy			0.81	624
macro avg	0.84	0.77	0.78	624
weighted avg	0.83	0.81	0.80	624

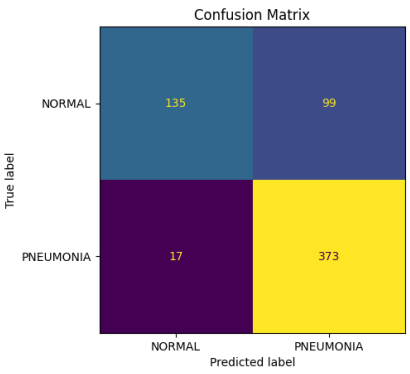

Fig. 6.6: Classification report and confusion matrix

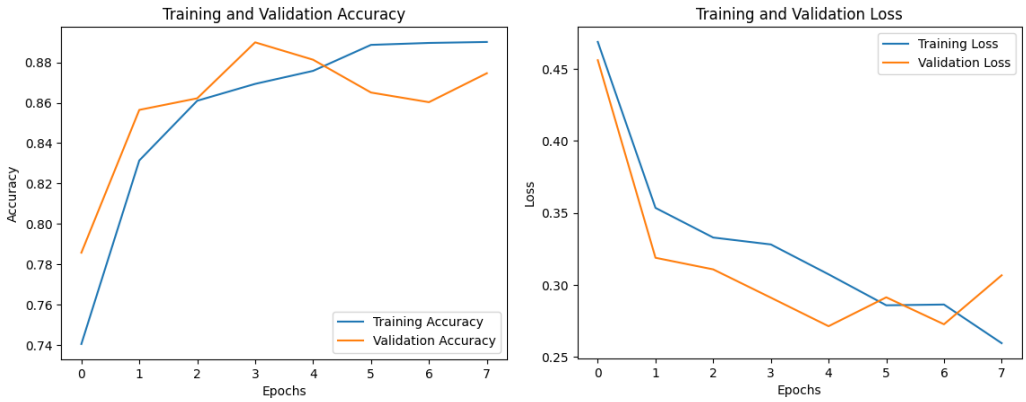

Fig. 6.7: Six convolutional layers accuracy and loss relationship plots

	precision	recall	f1-score	support
0	0.84	0.32	0.47	234
1	0.70	0.96	0.81	390
accuracy			0.72	624
macro avg	0.77	0.64	0.64	624
weighted avg	0.75	0.72	0.68	624

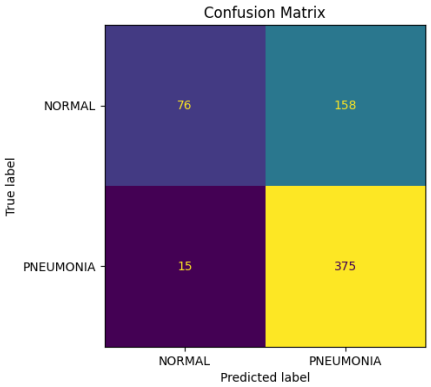

Fig. 6.8: Classification report and confusion matrix

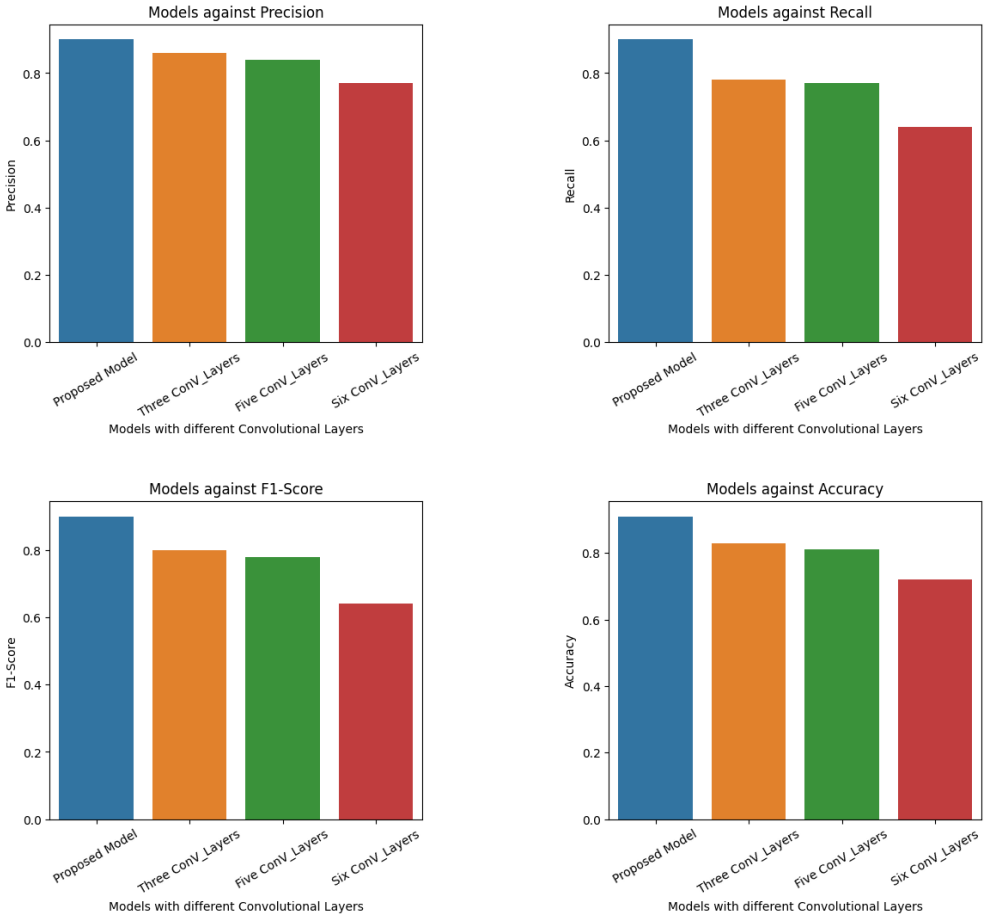

Fig. 6.9: Comparison of different models with different convolutional layers

6.2.2 OPTIMIZERS

Model	Precision	Recall	F1 – Score	Accuracy
Proposed Model (Adam)	0.90	0.90	0.90	0.91
SGD	0.85	0.78	0.79	0.82
Adagrad	0.82	0.54	0.47	0.66
RMSprop	0.85	0.75	0.77	0.81

Table 6.3: Comparison of different models with different optimizers

In this section, Figures 6.10, 6.11, 6.12, 6.13, 6.14 and 6.15 show the relationship between training and validation loss and the accuracy, and the classification reports, confusion matrices of models with Stochastic Gradient Descent (SGD), Adagrad, and RMSprop optimizers, respectively. Figure 6.16 shows comparison of different models with different optimizers.

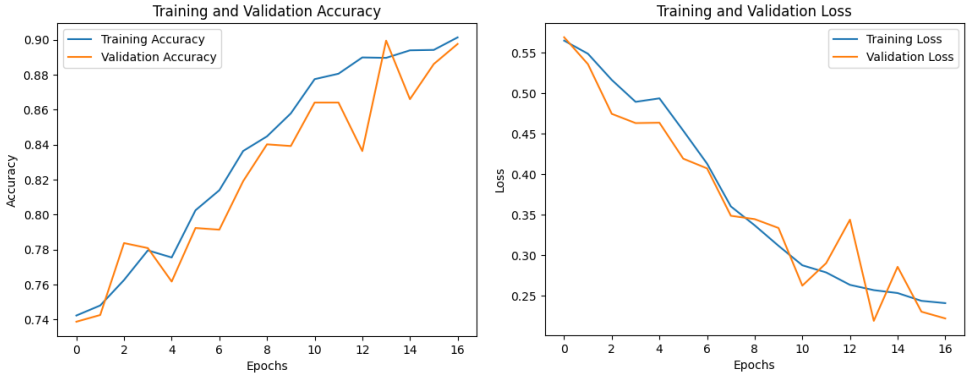

Fig. 6.10: SGD accuracy and loss relationship plots

	precision	recall	f1-score	support
0	0.91	0.59	0.72	234
1	0.80	0.96	0.87	390
accuracy			0.82	624
macro avg	0.85	0.78	0.79	624
weighted avg	0.84	0.82	0.81	624

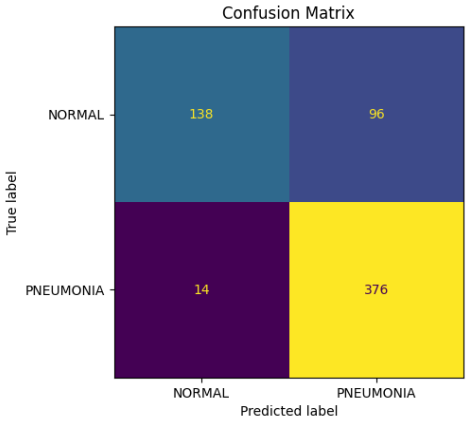

Fig. 6.11: Classification report and confusion matrix

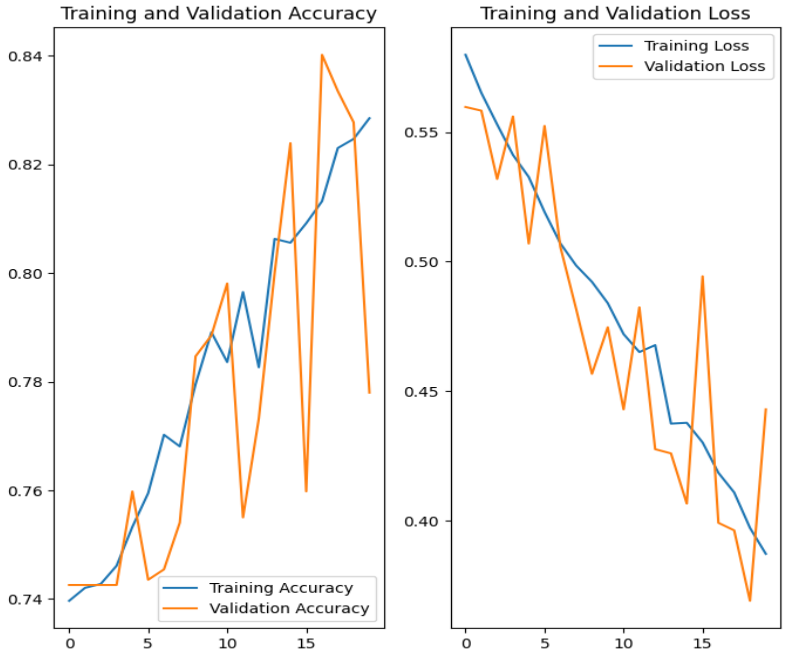

Fig. 6.12: Adagrad accuracy and loss relationship plots

	precision	recall	f1-score	support
0	1.00	0.08	0.15	234
1	0.64	1.00	0.78	390
accuracy			0.66	624
macro avg	0.82	0.54	0.47	624
weighted avg	0.78	0.66	0.55	624

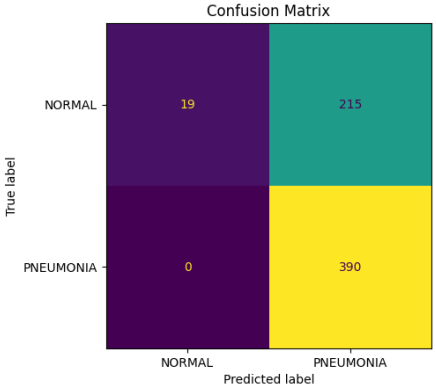

Fig. 6.13: Classification report and confusion matrix

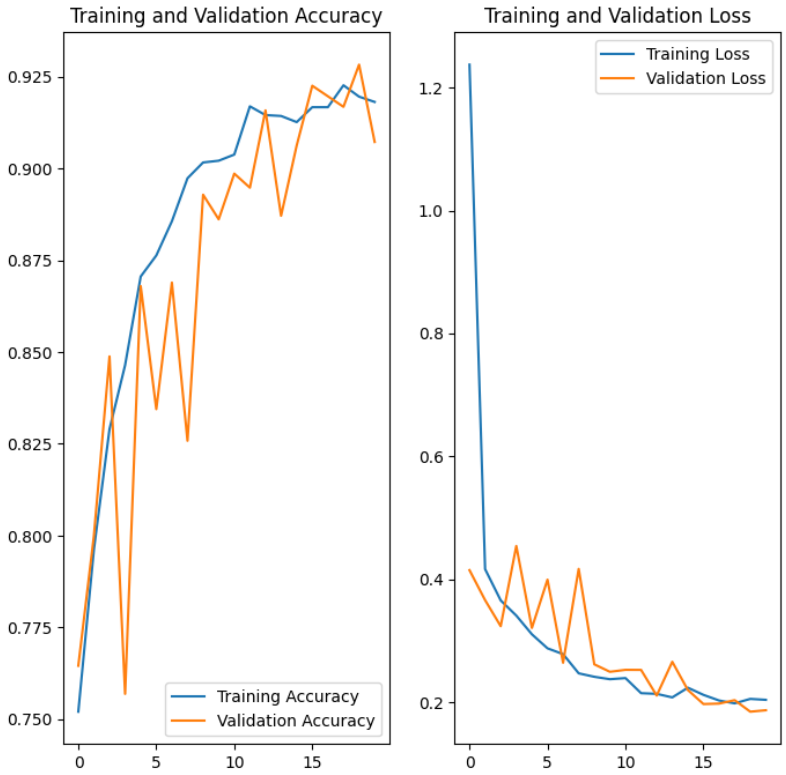

Fig. 6.14: RMSprop accuracy and loss relationship plots

	precision	recall	f1-score	support
0	0.93	0.53	0.68	234
1	0.78	0.97	0.86	390
accuracy			0.81	624
macro avg	0.85	0.75	0.77	624
weighted avg	0.83	0.81	0.79	624

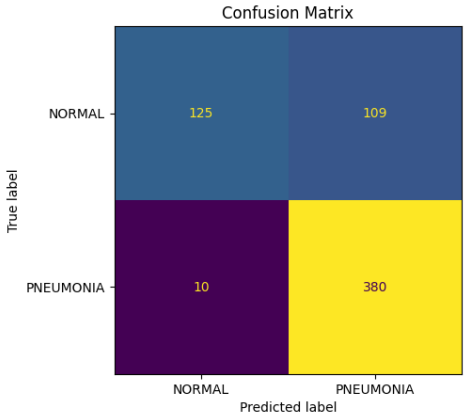

Fig. 6.15: Classification report and confusion matrix

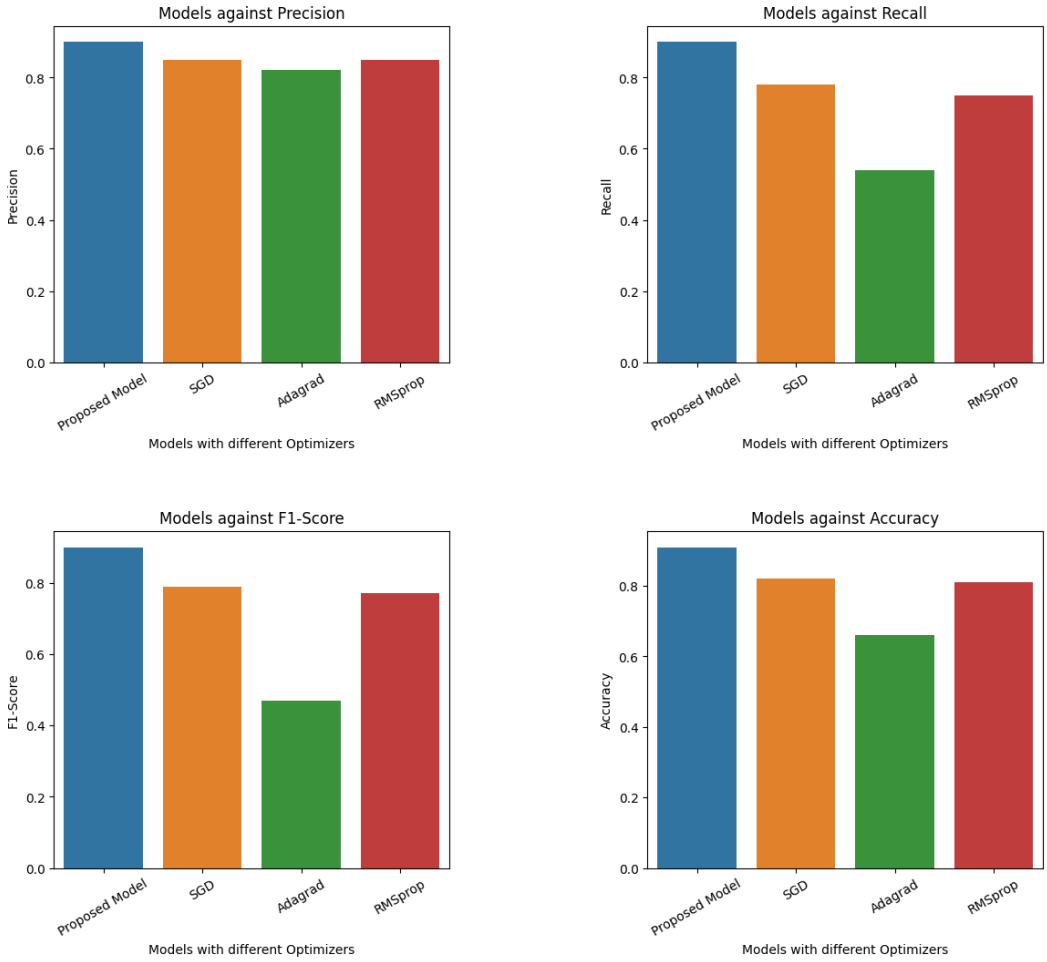

Fig. 6.16: Comparison of different models with different optimizers

6.3 PRETRAINED NETWORKS

6.3.1 TEST ACCURACY

Model	Test Accuracy
MobileNetV2	0.825
ResNet50	0.904
DenseNet201	0.896
VGG-19	0.872

Table 6.4: Test accuracy of pretrained networks

6.3.2 COMPARISON WITH THE PROPOSED MODEL

Model	Precision	Recall	F1 – Score	Accuracy
Proposed Model	0.90	0.90	0.90	0.91
MobileNetV2	0.93	0.78	0.82	0.88
ResNet50	0.93	0.89	0.90	0.91
DenseNet201	0.92	0.80	0.83	0.88
VGG-19	0.93	0.90	0.90	0.91

Table 6.5: Comparison of the pretrained networks with the proposed model

In this section, Figures 6.17, 6.18, 6.19, 6.20, 6.21, 6.22, 6.23 and 6.24 show the relationship between training and validation loss and the accuracy before and after fine-tuning, the classification reports, and confusion matrices of the pretrained networks, respectively. Figure 6.25 shows comparison with the proposed model.

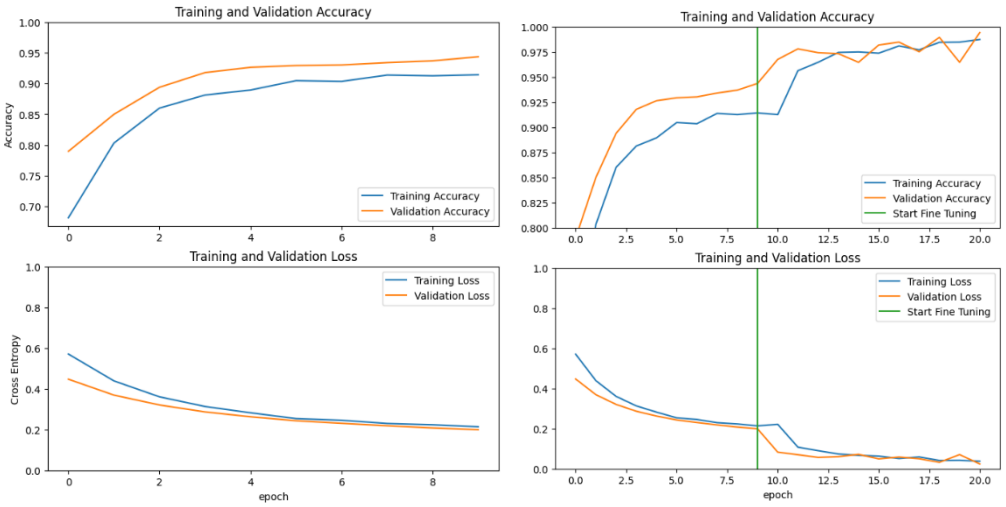

Fig. 6.17: MobileNetV2 accuracy and loss before and after fine-tuning relationship plots

	precision	recall	f1-score	support
0.0	1.00	0.56	0.71	9
1.0	0.85	1.00	0.92	23
accuracy			0.88	32
macro avg	0.93	0.78	0.82	32
weighted avg	0.89	0.88	0.86	32

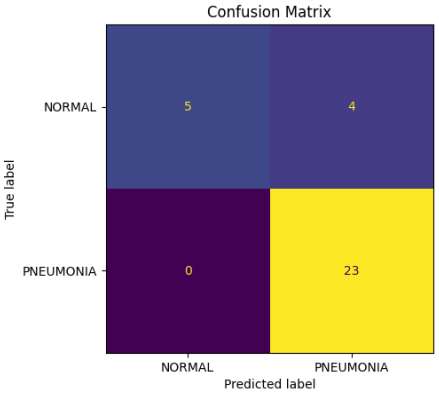

Fig. 6.18: Classification report and confusion matrix

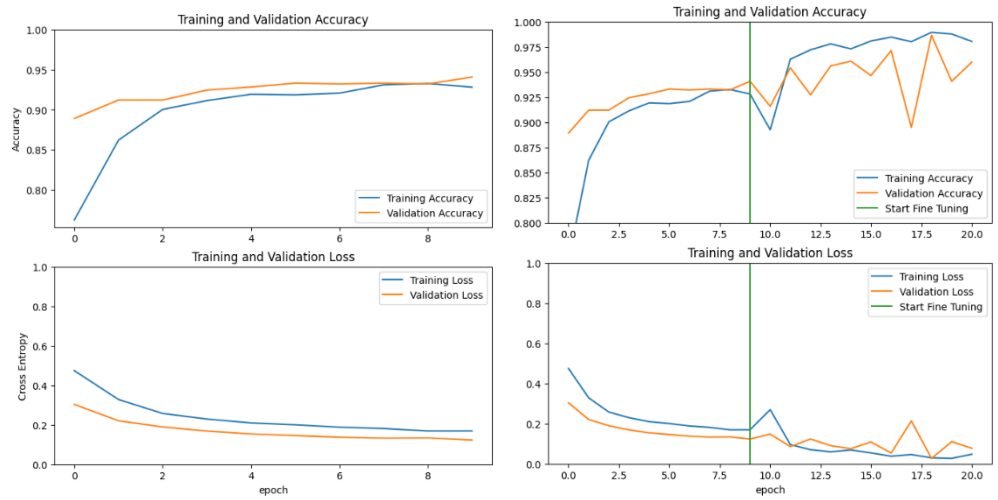

Fig. 6.19: ResNet50 accuracy and loss before and after fine-tuning relationship plots

	precision	recall	f1-score	support
0.0	1.00	0.79	0.88	14
1.0	0.86	1.00	0.92	18
accuracy			0.91	32
macro avg	0.93	0.89	0.90	32
weighted avg	0.92	0.91	0.90	32

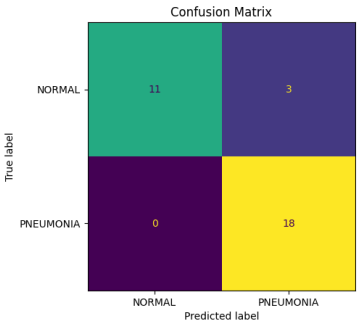

Fig. 6.20: Classification report and confusion matrix

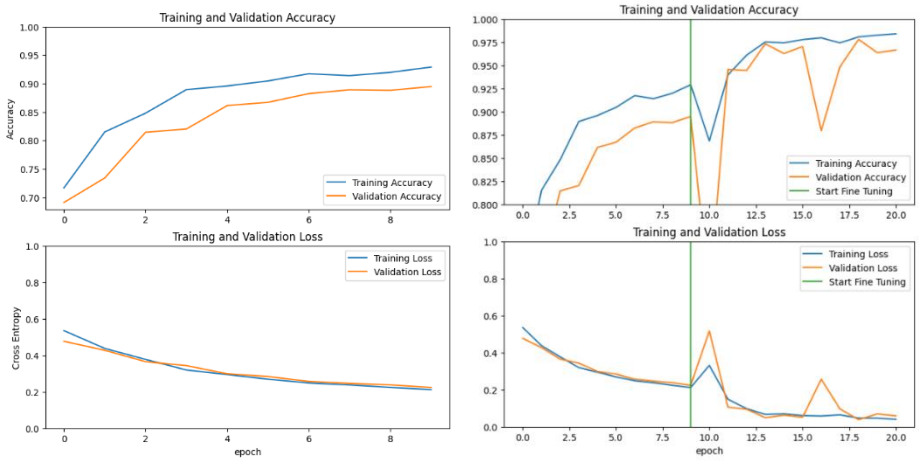

Fig. 6.21: DenseNet201 accuracy and loss before and after fine-tuning relationship plots

	precision	recall	f1-score	support
0.0	1.00	0.60	0.75	10
1.0	0.85	1.00	0.92	22
accuracy			0.88	32
macro avg	0.92	0.80	0.83	32
weighted avg	0.89	0.88	0.86	32

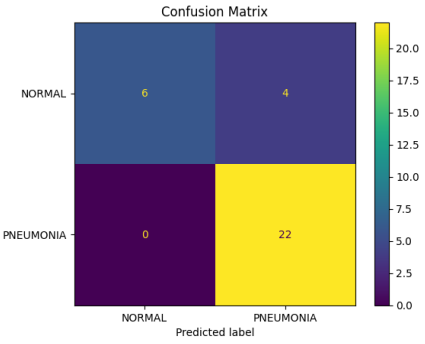

Fig. 6.22: Classification report and confusion matrix

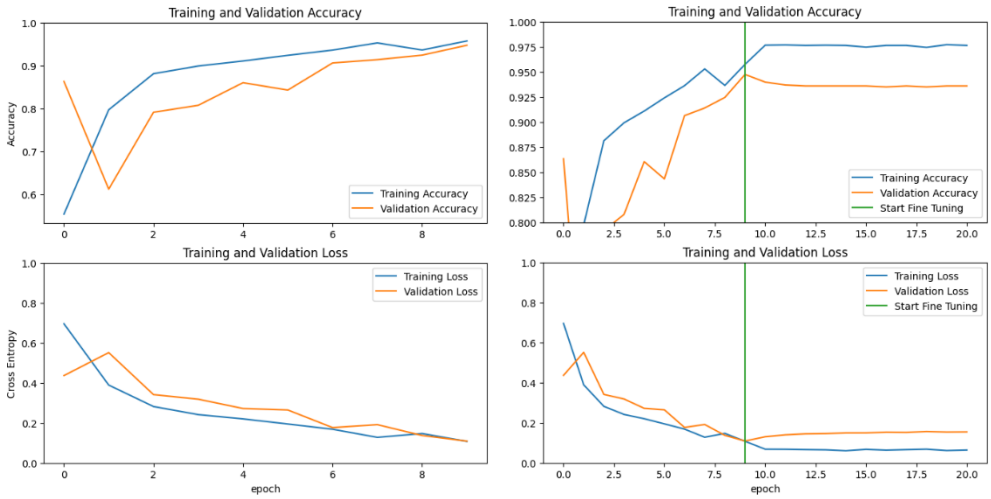

Fig. 6.23: VGG-19 accuracy and loss before and after fine-tuning relationship plots

	precision	recall	f1-score	support
0.0	1.00	0.80	0.89	15
1.0	0.85	1.00	0.92	17
accuracy			0.91	32
macro avg	0.93	0.90	0.90	32
weighted avg	0.92	0.91	0.90	32

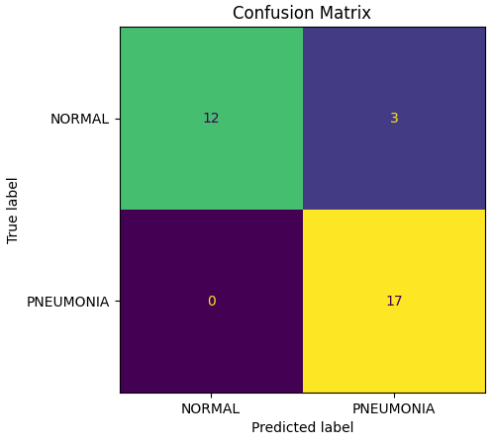

Fig. 6.24: Classification report and confusion matrix

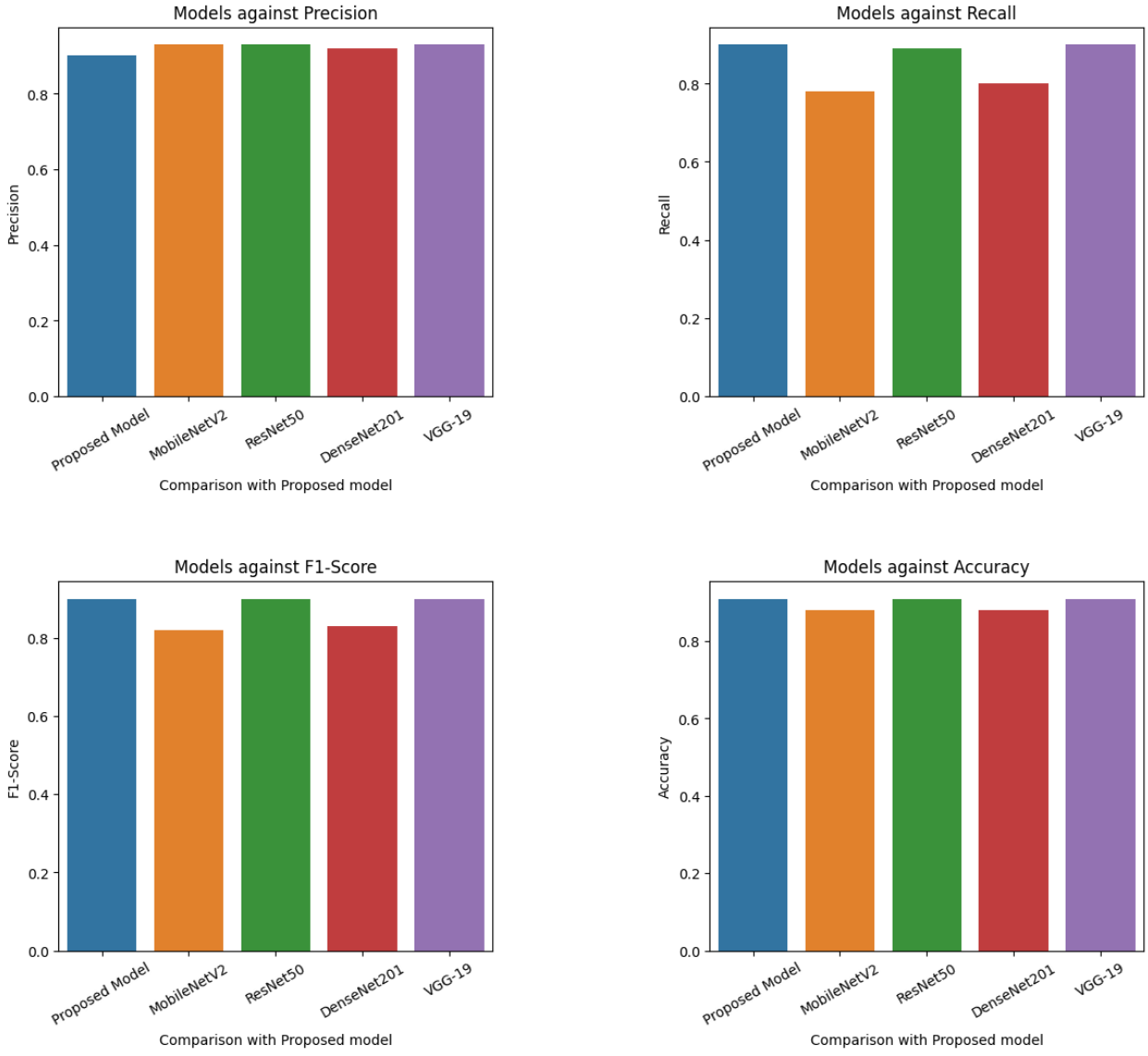

Fig. 6.25: Comparison with the proposed model

6.4 DISCUSSION OF RESULTS

The result shows the performance of the proposed model built from scratch in accurately detecting pneumonia, with the confusion matrix indicating that it can accurately classify 202 Normal and 364 Pneumonia samples and misclassified 32 Normal and 26 Pneumonia samples, and comparison with hyperparameters tuning and pretrained networks.

The proposed model with four convolutional layers and Adam Optimizer achieved the best results compared to other hyperparameters tuning and, therefore, is highly recommended for future works. According to Huilgol (2020), recall plays a vital role in accurately identifying the number of patients with pneumonia, and the proposed model yielded 90% recall, which is 11.1% higher than the next model.

Finally, compared to other pretrained networks, the proposed model achieved 91% accuracy alongside ResNet50 and VGG-19. Regarding overall performance, the VGG-19 was the best with 90% recall, 90% f1-score, and 93% precision, as seen in the confusion matrix, where it accurately classified all pneumonia samples and misclassified 3 Normal samples.

7.0 CONCLUSION

This study described a proposed CNN model for detecting Pneumonia on chest X-ray images, achieving a 91% accuracy, 90% f1-score, 90% precision, and 90% recall. In addition, this model is then compared with models of varying hyperparameters, such as convolutional layers and optimizers. Adagrad and six convolutional layers were the least-performing models while tuning the hyperparameters, with the proposed model obtaining the best result. Additionally, VGG-19 and the proposed model were the best in detecting Pneumonia, with VGG-19 edging out by 3.22% precision, with other evaluation metrics the same for both models. Therefore, I recommend using Adam Optimizers and four convolutional layers to future researchers for performance optimization when diagnosing Pneumonia.

Limitations like time constraints in building the model, dataset size, and availability of recent X-ray images are challenging. In the future, I will research to explore ways to access new chest X-ray images and increase the number of epochs for better performance and diagnosing Pneumonia into fungus, bacteria, and viruses.

REFERENCES

Aslan, Z. (2022) *Transfer Learning Using DenseNet201 - MLearning.ai - Medium*. Medium. Available online: https://medium.com/mlearning-ai/transfer-learning-using-densenet201-525749762ca9 [Accessed 8 Sep. 2023].

Chouhan, V.S., Sanjay Kumar Singh, Aditya Khamparia, Gupta, D., Tiwari, P., Moreira, C., Robertas Damaševičius and Hugo, V. (2020) A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images. *Applied sciences*, 10(2), pp.559–559. Available online: https://doi.org/10.3390/app10020559

Gupta, A. (2021) *A Comprehensive Guide on Optimizers in Deep Learning*. Analytics Vidhya. Available online: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/ [Accessed 9 Sep. 2023].

Huilgol, P. (2020) *Precision and Recall | Essential Metrics for Machine Learning (2023 Update)* Analytics Vidhya. Available online: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/ [Accessed 10 Sep. 2023].

Ikechukwu, V. A., Murali, S., R Deepu and 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), pp.375–381. Available online: https://doi.org/10.1016/j.gltp.2021.08.027

Jain, R., Preeti Nagrath, Kataria, G., Kaushik, V. and D. Jude Hemanth (2020) Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurement*, 165, pp.108046–108046. Available online: https://doi.org/10.1016/j.measurement.2020.108046

Kermany, D., Goldbaum, M.H., Cai, W., Valentim, C.C.S., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M.K., Pei, J., Yin, M., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I. and Shi, A. (2018) Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5), pp.1122-1131.e9 Available online: https://doi.org/10.1016/j.cell.2018.02.010

Kharwal, A. (2021) *Classification Report in Machine Learning | Aman Kharwal*. thecleverprogrammer. Available online: https://thecleverprogrammer.com/2021/07/07/classification-report-in-machine-learning/ [Accessed 10 Sep. 2023].

Krishnamurthy, S., Srinivasan, K., Saeed Mian Qaisar, Vincent, P. and Chang, C. Y. (2021) Evaluating Deep Neural Network Architectures with Transfer Learning for Pneumonitis Diagnosis. *Computational and Mathematical Methods in Medicine*, 2021, pp.1–12. Available online: https://doi.org/10.1155/2021/8036304

Kundu, R., Das, R., Zong Woo Geem, Han, G. T. and Sarkar, R. (2021) Pneumonia detection in chest X-ray images using an ensemble of deep learning models. *PLOS ONE*, 16(9), pp. e0256630 – e0256630. Available online: https://doi.org/10.1371/journal.pone.0256630

Mabrouk, A., Díaz, R.P., Abdelghani Dahou, Mohamed Abd Elaziz and Kayed, M. (2022) Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks. *Applied sciences*, 12(13), pp.6448–6448. Available online: https://doi.org/10.3390/app12136448

Mooney, P. (2018) *Chest X-Ray Images (Pneumonia)*. Kaggle.com. Available online: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia [Downloaded 15 July 2023].

Mukherjee, S. (2022) *The Annotated ResNet-50 - Towards Data Science*. Medium. Available online: https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758 [Accessed 7 Sep. 2023].

Nguyen, T.H., Nguyen, T.N. and Ngo, B.V. (2022) A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. *AgriEngineering*, 4(4), pp.871–887. Available online: https://doi.org/10.3390/agriengineering4040056

NHS (2023) *Pneumonia* Available online: https://www.nhs.uk/conditions/pneumonia/ [Accessed 5 Sep. 2023].

ŞAHİN, M. E., Ulutaş, H. and Yüce, E. (2021) *A deep learning approach for detecting pneumonia in chest X-rays*. ResearchGate. Available online: https://www.researchgate.net/publication/355255310_A_deep_learning_approach_for_detecting_pneumonia_in_chest_X-rays [Accessed 10 Sep. 2023].

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L. C. (2018) *MobileNetV2: Inverted Residuals and Linear Bottlenecks*. arXiv.org. Available online: https://arxiv.org/abs/1801.04381 [Accessed 7 Sep. 2023].

Saturn Cloud (2023) *Keras Binary Classification: A Deep Dive into the Sigmoid Activation Function | Saturn Cloud Blog*. Saturncloud.io. Available online: https://saturncloud.io/blog/keras-binary-classification-a-deep-dive-into-the-sigmoid-

activation-function/#:~:text=The%20sigmoid%20activation%20function%20is,well%20with%20many%20optimization%20algorithms. [Accessed 6 Sep. 2023].

Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv.org. Available online: https://arxiv.org/abs/1409.1556 [Accessed 8 Sep. 2023].

TensorFlow. (2023) *tf.keras.preprocessing.image.ImageDataGenerator | TensorFlow v2.13.0*. Available online: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator [Accessed 7 Sep. 2023].

Tsang, S. H. (2019) *Review: MobileNetV2 — Light Weight Model (Image Classification)*. Medium. Available online: https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c [Accessed 7 Sep. 2023].

UNICEF (2023) *Pneumonia in Children Statistics - UNICEF DATA* Available online: https://data.unicef.org/topic/child-health/pneumonia/ [Accessed 5 Sep. 2023].

Wang, S. and Zhang, Y. (2020) *DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple...* ResearchGate. Available online: https://www.researchgate.net/publication/342356888_DenseNet-201-Based_Deep_Neural_Network_with_Composite_Learning_Factor_and_Precomputation_for_Multiple_Sclerosis_Classification [Accessed 10 Sep. 2023].

Zhang, D., Ren, F., Li, Y., Lei, N. and Ma, Y. (2021) Pneumonia Detection from Chest X-ray Images Based on Convolutional Neural Network. *Electronics*, 10(13), pp.1512–1512. Available online: https://doi.org/10.3390/electronics10131512

Zhou, X., Liu, H., Shi, C. and Liu, J. (2022) Model design and compression. *Elsevier eBooks*, pp.39–58. Available online: https://doi.org/10.1016/b978-0-32-385783-3.00011-9