Chest Disease Detection Applying Deep Neural Network

Submitted by

Junaed Mohammed Uddin	**160104109**
Anika Siddika	**160104110**
Syed Muztaba Ali	**160104115**

Supervised by

Mr. Mir Tafseer Nayeem
Assistant Professor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

September, 2020

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Classification task is a process of categorizing a given set of data into classes. It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points [1]. For example, we want to detect pneumonia disease from the chest x-ray images. So, we have two classes in the dataset. One is pneumonia, and the other is normal. Through the classification task, we can classify the images in the dataset and tell whether the image is normal or infected with pneumonia. Furthermore, pneumonia has the highest mortality rate among young children and old aged people around the world. It is the pre symptom of COVID-19, which spreads across the globe, with the number of victims, and the death rate is steadily rising. In the diagnosis of pneumonia, X-rays becomes very useful but time-consuming. However, indeed this method does not work well when the epidemic turns into a terrible situation. Immediately, there is a dire need for automated diagnosis systems to assist clinicians in making better decisions. It is a medical base experiment, and we are eager to do it. In this study, we have detected the disease for binary classification using Deep Convolutional Neural Network (DCNN) and Transfer learning (VGG16) architectures with improved accuracy that reduces the workload of human experts. This ablation experiment has revealed some crucial aspects of this work. Additionally, Image preprocessing is not a necessary step in deep learning. Without preprocessing, we have got satisfactory results in the (CXI) dataset.

Chapter 2

Related Works

To make a perfect diagnosis and detect the source of the disease's problems, and suitable time remains a significant challenge for the doctors to reduce the patients' suffering. Indeed, image processing and deep learning algorithms in the analysis and processing of biomedical images have given very satisfactory results. In this section, a brief review of some crucial contributions from the existing literature is presented.

In this paper [2], pneumonia was detected from the chest x-ray using a Deep Convolutional Neural Network (DCNN). This paper aims to build a system that can rapidly detect pneumonia, which can be occurred by SARS, Covid-19, etc. to reduce human error. The team also wanted to know if there was any Deep Learning technique that can outperform other DCNN methods. This paper claimed that they got an accuracy of 96%. To achieve the result, the researchers used a baseline CNN and transfer learning in this experiment. Different image sizes were used for a different architecture. Intensity Normalization, CLAHE, and data augmentation were applied in the pre-processing step. A baseline CNN with three convolutional layers and the fine-tuned version of VGG16, VGG19, DenseNet201, ResNet50, Inception V3, Inception ResNet V2, and MobileNet V2 were used for classification.

In this paper [3], pediatric pneumonia was detected from the chest x-ray as pediatric pneumonia was poorly studied. CNN could help this situation, but it could be considered black boxes and poorly understood. This lack of transparency is regarded as a severe condition of medical screening. Visualization tools were proposed in this paper to explain the model. This paper highlighted the advantages of visualizing CNN's behaviors. In this experiment, the researchers used a dataset which was consisted of anteroposterior chest radiograph of one to five years of age. The dataset has three classes, i.e., normal, bacterial pneumonia, and viral pneumonia. The researchers used a custom CNN and pre-trained VGG16 network for feature extraction and classification and used Bayesian optimization to search for optimal model parameters. Furthermore, gradient weighted class activation maps (grad-CAM) and Local Interpretable Model-Agnostic Explanation (LIME) visualization tools are used to explain predictions and evaluate the usefulness of the model in decision making.

In this paper [4], a scientific evaluation of various approaches for CNN-based X-ray classification on ChestX-ray14. In contrast, satisfactory results were obtained with networks optimized on the ImageNet dataset. It is supported transfer learning (ResNet-50, ResNet-101, ResNet-38) with and without fine-tuning as well as the training of dedicated X-ray network from scratch. Data augmentation is employed to increase the ChestX-ray14 dataset. The dataset has been divided into 70% training, 10% validation, and 20% testing. In training, different sized patches of the image are sampled, with sizes ranging between 8% and 100% of the image. For validation and testing, Images are resized to 256 * 256 and 480 * 480 pixels for tiny and substantial spatial sizes. To possess an excellent comparison to other groups, it reported results on this segmentation for best-performing architecture with different depths between ResNet38, ResNet-50, and ResNet-101.

In this paper [5], the researchers detected many types of lung diseases from the chest X-rays. The main goal of the researchers was to initiate future efforts by promoting public datasets. According to this Paper, many X-ray images were stored in many modern hospitals' Picture Archiving and Communication System (PACS). But they were loosely labeled. So, they presented a new chest X-ray database called ChesX-ray8. The researchers showed that the database's diseases could be successfully detected and even spatially located using weakly supervised multi-label classification and disease localization framework. In this experiment, at first, researchers constructed the "ChestX-ray8" database with their own institutes' PACS system. They used a variety of Natural Language Processing (NLP) to extract images from the system. The database consisted of eight diseases. Here, DNorm and MetaMap have been used to identify pathology. Images are resized to 1024x1024. Then they used a Deep Convolutional Neural Network (DCNN) for disease detection and object localization.

To detect chest x-ray diseases, most of the researchers have worked with a supervised learning algorithm. The above papers have highly satisfactory results for classifying chest diseases using supervised learning. But, there are very few research papers that have used unsupervised learning. In this context, these papers have missed this aspect. In this case, we would like to reimplement this classification task using image preprocessing based on supervised learning.

Chapter 3

Project Objective

Our main objective is to reimplement an efficient model that can correctly classify any diseases like Pneumonia, Brain Tumor, heart diseases, etc. To facilitate the work, we have divided the main task of the project into four subtasks. In this case, the subtasks are described step by step in the following sections. These are -

- Dataset Preprocess
- Trained with Convolutional Neural Network (CNN)
- Trained with Transfer Learning
- Comparison of the results between CNN and Transfer Learning

3.1 Dataset Preprocess

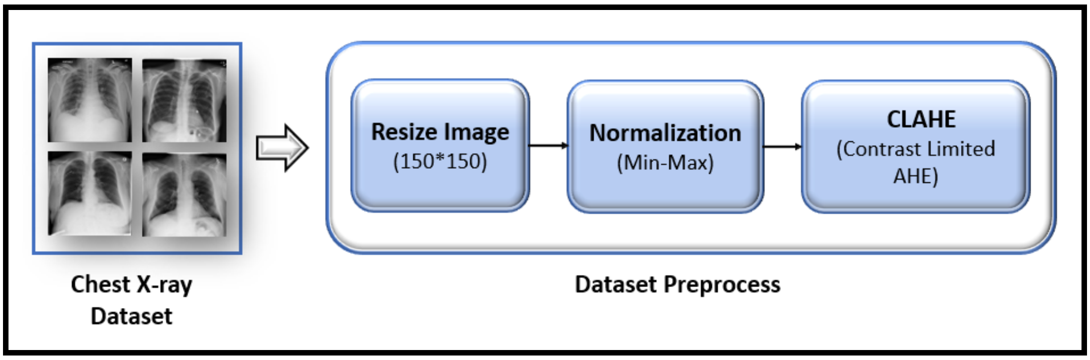


Figure 3.1: Dataset Preprocess

First of all, we have resized the chest x-ray images into 150*150 for reducing computational cost and normalize those images by using **min-max normalization** to make every image have the same scale. Then we have applied **CLAHE (Contrast Limited Adaptive Histogram Equalization)** to the X-ray images before training to redistribute the lightness values of the image.

3.2 Trained with Convolutional Neural Network (CNN)

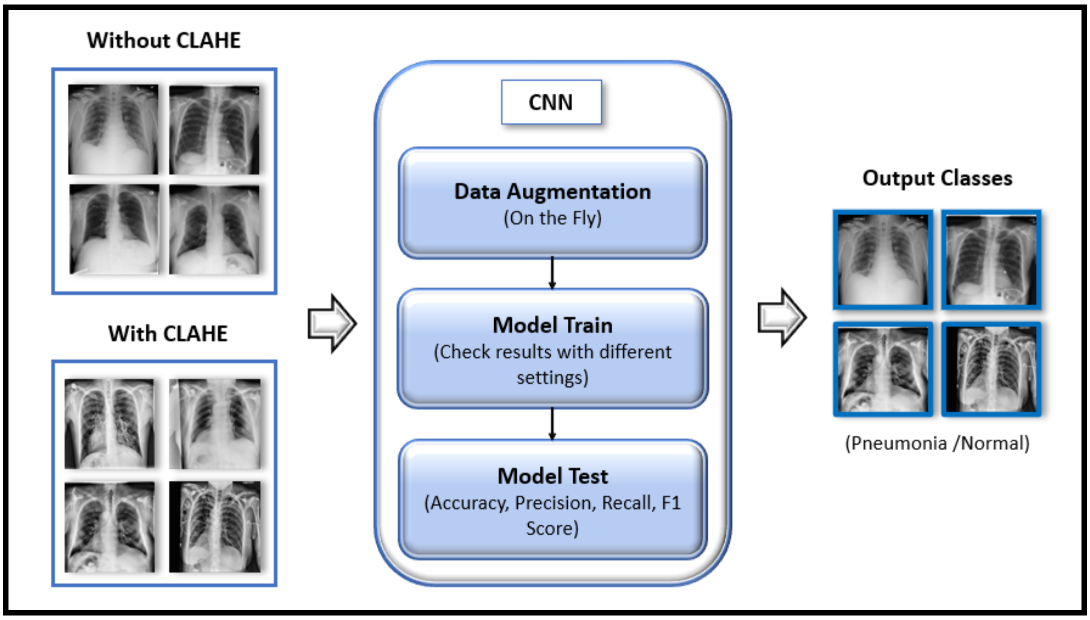


Figure 3.2: Trained with Convolutional Neural Network

Moreover, we have trained our dataset in two ways by using the Convolutional Neural Network. The first way is to train CNN without CLAHE on the dataset, and another is to train after applying CLAHE. We have trained the CNN model by augmenting the data using on the fly method in two conversions of the dataset and evaluate the results with different settings. Then, we have classified the output classes, either pneumonia or normal, by testing our model successfully.

3.3 Trained with Transfer Learning (VGG16)

Similarly, we have trained VGG16 after using CLAHE and without CLAHE, the renowned architecture of transfer learning. To avoid the risk of overfitting, we have used data augmentation using on the fly approach in two conversions of the dataset and evaluate the results by testing this architechture successfully.

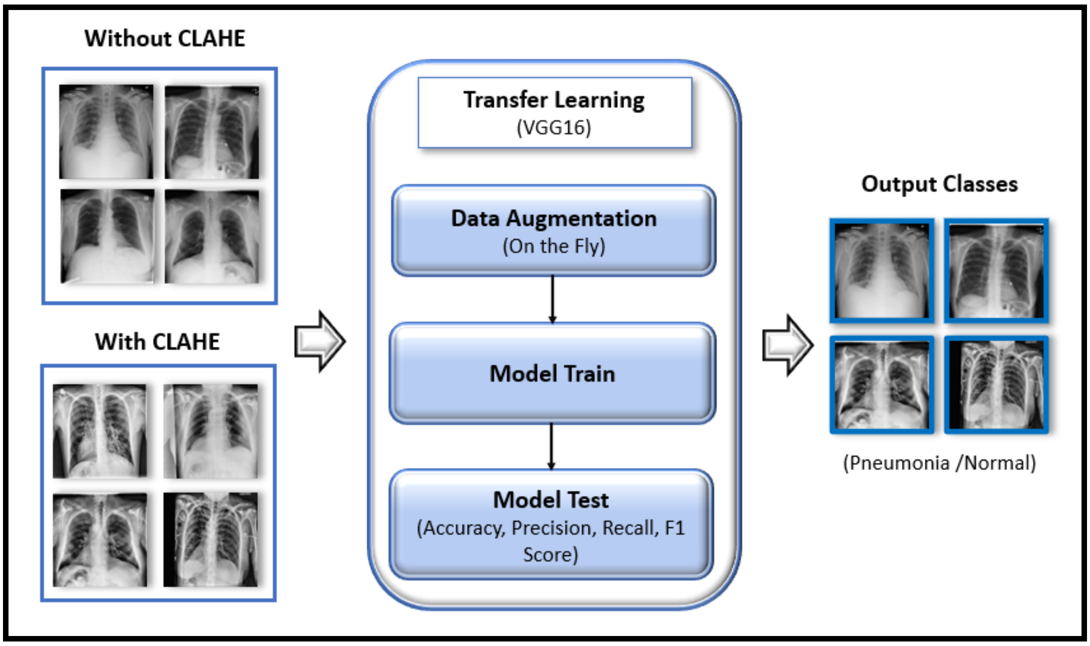


Figure 3.3: Trained with Transfer Learning (VGG16)

3.4 Comparison of the results between CNN and VGG16

Finally, we have compared the best results of VGG16 and CNN by using the evaluation metric. The full working procedure of our project is given below-

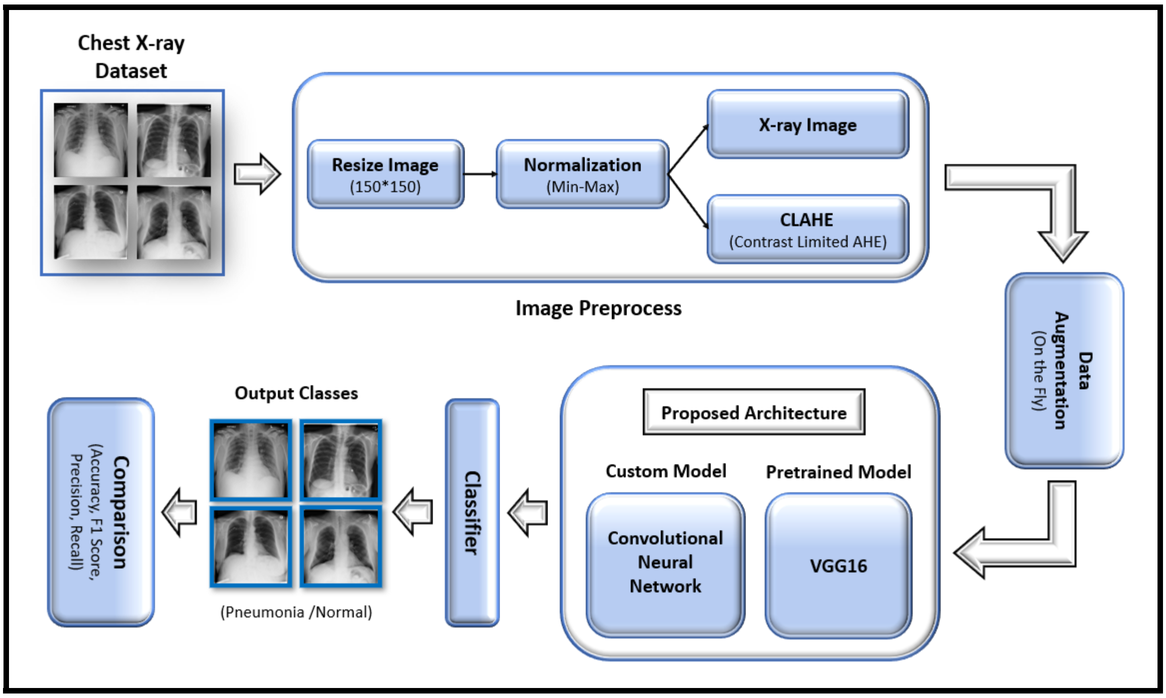


Figure 3.4: Full Flowchart of our Project

3.5 Dummy Input and Output

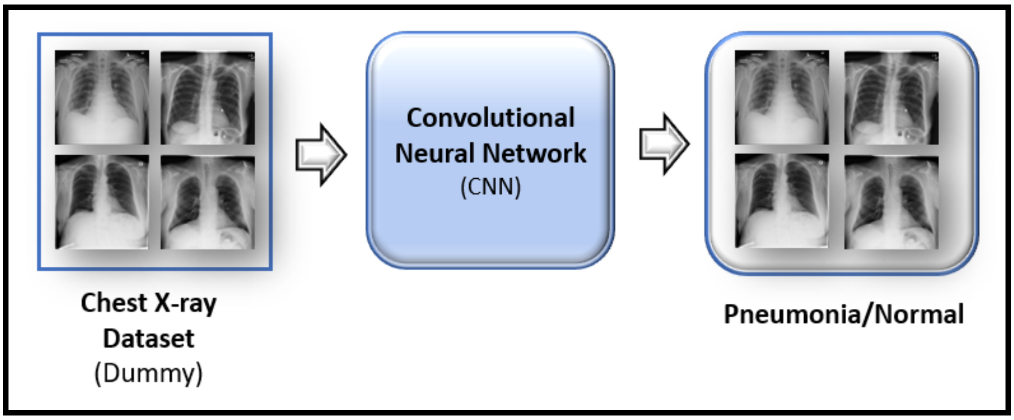


Figure 3.5: Dummy Input and Output of CNN

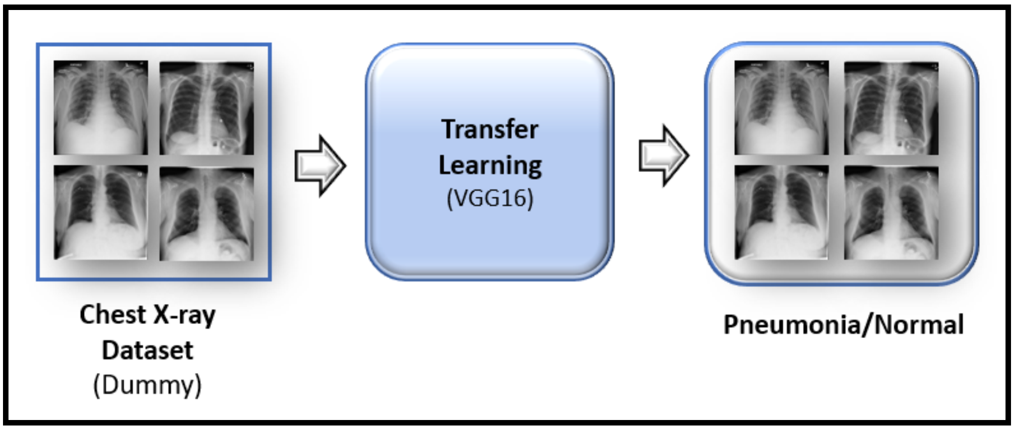


Figure 3.6: Dummy Input and Output of VGG16

Chapter 4

Methodologies

In this project, we have trained models using CNN and transfer learning architecture and determined which model is better based on the ablation experiment. These are described in the following two sections.

4.1 Convolutional Neural Network (CNN)

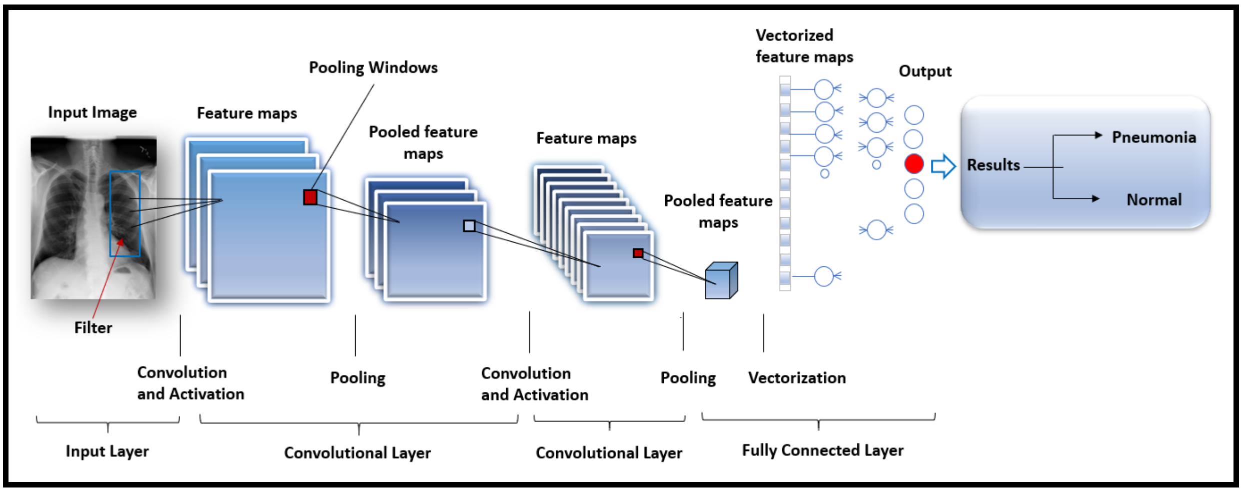


Figure 4.1: Baseline of CNN

Technically, Deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1 [6]. Without losing on the quality of models, CNN's are very effective in reducing the number of parameters. It is helpful to reduce the dimensionality of the Image and efficient to classify in the Image domain [7]. In that case, we have used CNN architecture for image classification. In this project, a total of seven CNN-centric ablation experiments have been performed, of which CNN's Setup-4 has given better results than other setups. We have used early stopping. If the validation accuracy of a setup didn't increase after consecutive five epochs then the training have stopped.

In each setup, we have changed the layers and hyperparameters of CNN model one by one for checking the result with CLAHE and without CLAHE. We have changed the layer from setup-1 to setup-4 and checked the result by keeping ReLU as activation function, and Adam as the optimizer. As seen, accuracy is precise for ten layers and optimum for the conversion of two datasets in the setup-4. But in the same configuration, the result is a downgrade for 12 layers in setup-5. So we left ten layers in setup-6 and just used tanh as activation function and sigmoid as optimizer in setup-6. It can be seen that the result of setup-6 and setup-7 as bad as the previous setup-5. So, through this ablation experiment, it can be said that setup-4 is better than other setups. Then we have calculated the testing result of setup-4 by evaluation metric and store the results in the corresponding table.

4.2 VGG16

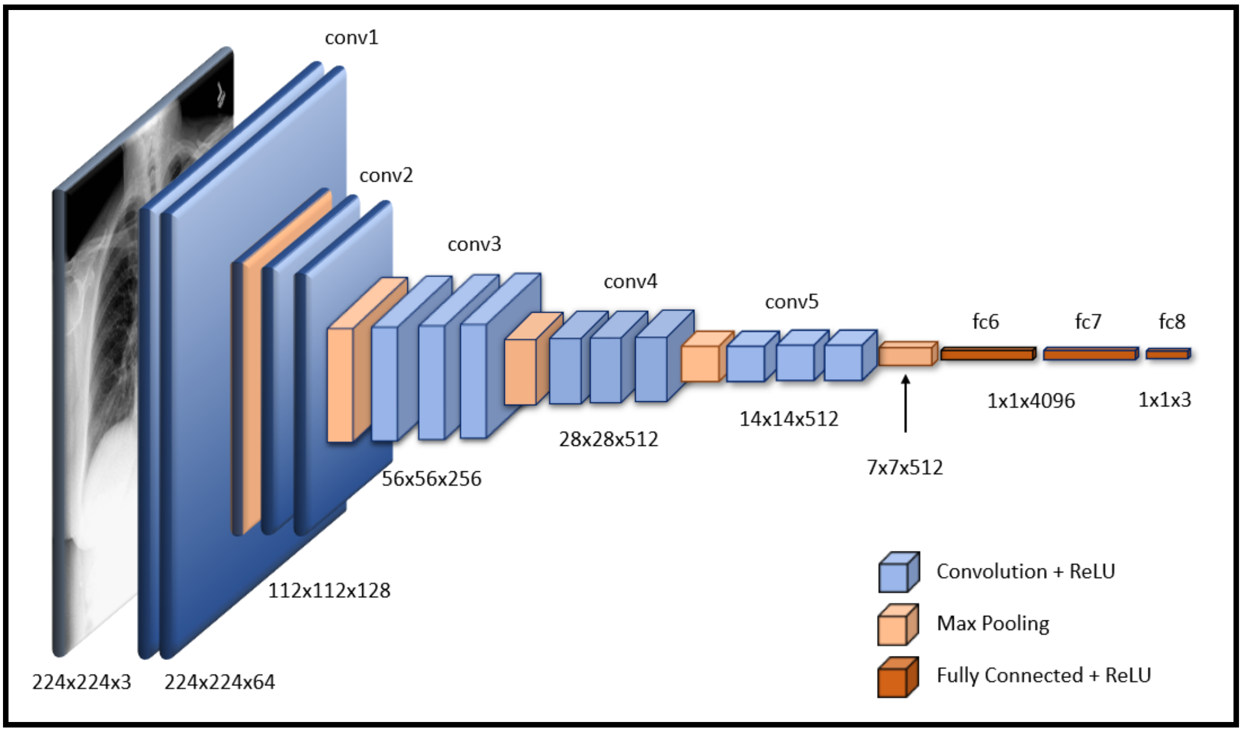


Figure 4.2: Baseline of VGG16

The VGG16 architecture is computationally less costly and less time consuming than other pre-trained architectures [8]. For this reason, we have used this architecture and trained the VGG16 model in two ways with CLAHE and without CLAHE. Based on the training and validation accuracy, we have checked the training evaluation by applying CLAHE and without CLAHE. Finally, we have found the testing accuracy by evaluation metric and stated that result in the table.

Chapter 5

Experiments

5.1 Chest X-Ray Image (CXI) Dataset

This present work introduces a publicly available image dataset which contains X-Ray and computed tomography (CT) images. This dataset, named chest X-Ray CT dataset, which is composed of 5856 images (jpeg format) and has two categories (4273 pneumonia and 1583 normal). This dataset is developed in the Guangzhou Women and Children's Medical Center, Guangzhou, China [9]. The dataset is further subdivided into three folders: training, validation, and test sets, and each folder contain images from both categories: pneumonia and normal. The dataset and their characteristics are given below.

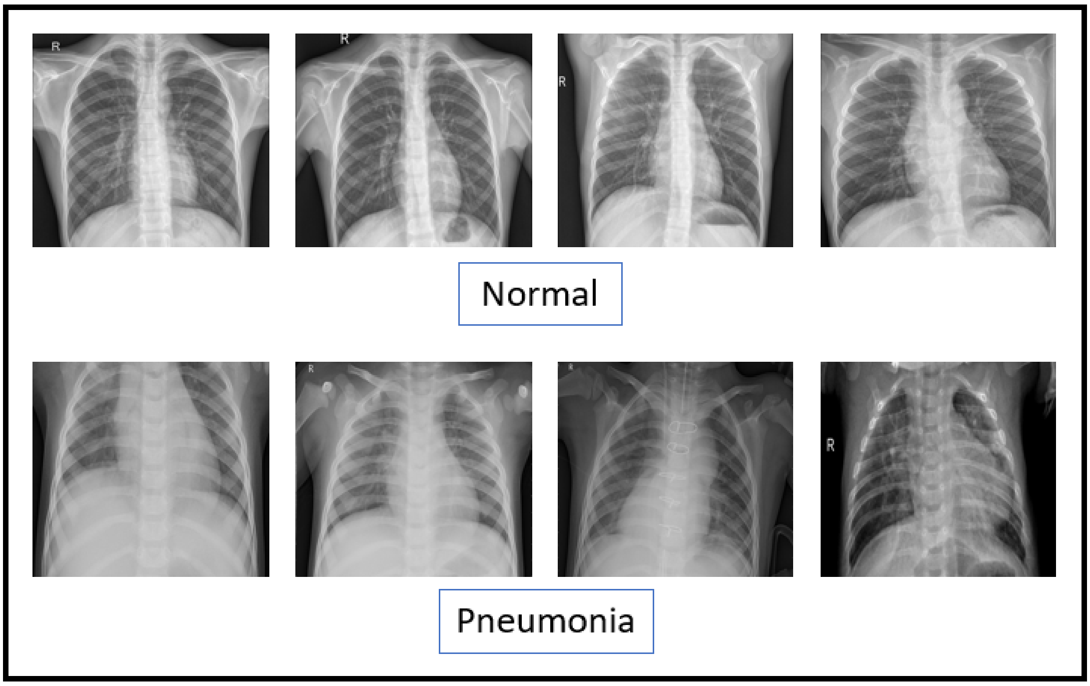


Figure 5.1: Dataset

Table 5.1: (CXI) Dataset and their characteristics

Category	Samples	Training Samples	Validation Samples	Testing Samples	Type	Depth	Total Samples
Pneumonia	4273	2565	854	854	JPEG	8-bit	5856
Normal	1583	951	316	316	JPEG	8-bit	

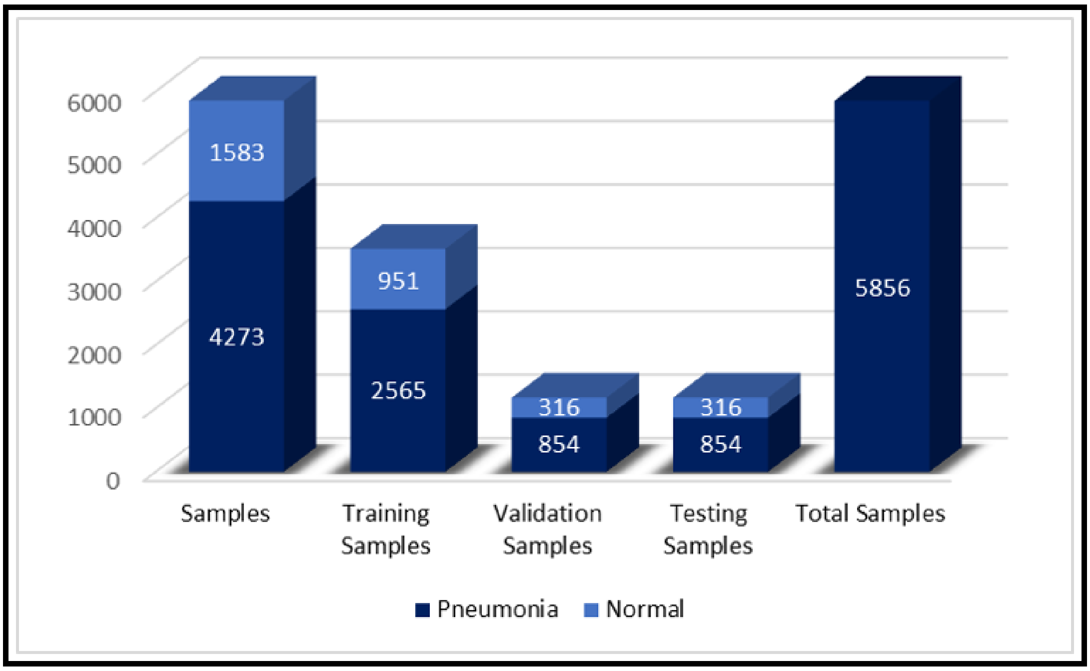


Figure 5.2: Characteristics of CXI Dataset

5.2 Evaluation Metric

Evaluation metrics are used to measure the quality of the statistical or Machine learning model. After extracting the appropriate feature, the last step is to classify the attained data and assign it to a specific class. There are many different types of evaluation metrics available to test a model [10]. These include classification accuracy, F1 Score, Recall, Precision, confusion matrix, etc., which defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

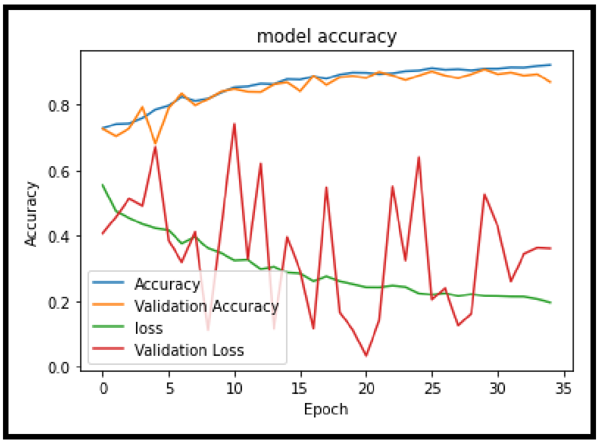
$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

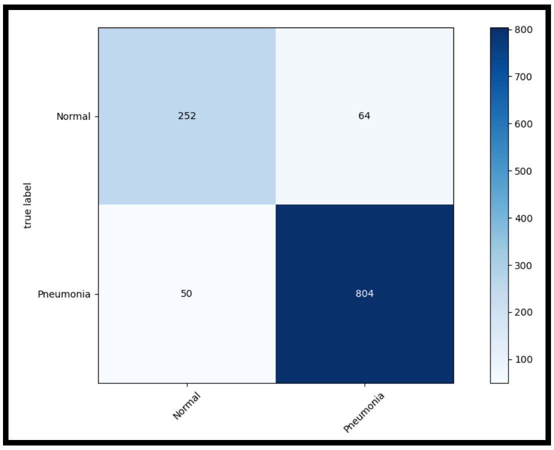
$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5.4}$$

Where, TP : True Positive, FP : False Positive, TN : True Negative, and FN : False Negative

5.2.1 CNN without CLAHE



(a) Training Accuracy and Loss



(b) Confusion Matrix

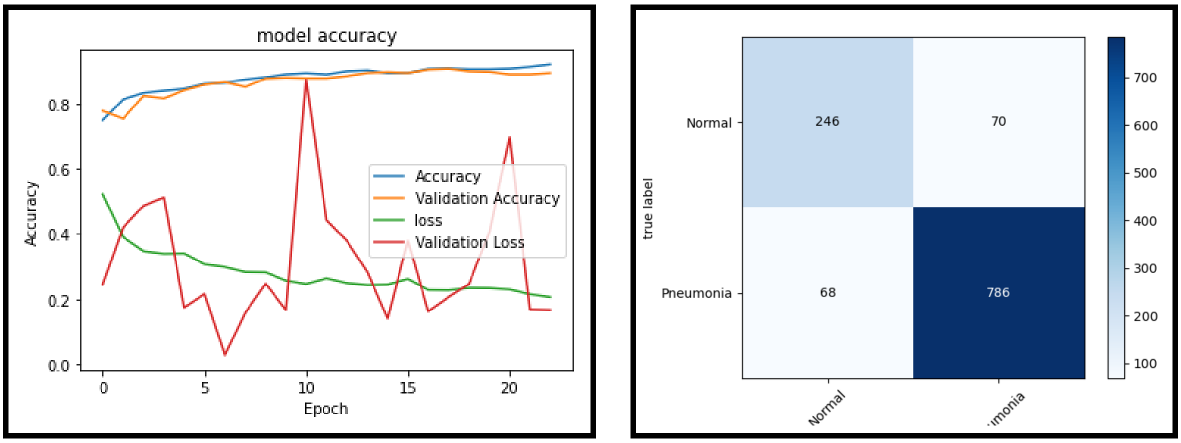
Figure 5.3: Training Accuracy, Loss and Confusion Matrix of Setup-4

From the above fig. 5.3, it is evident that this setup is really good. Here, the training and validation accuracy and loss is shown in fig. 5.3a and confusion matrix of setup-4 in fig. 5.3b. This setup-4 of CNN is trained without applying CLAHE. Moreover, the setup converges after 30 epochs and achieves a training accuracy of around 91% and validation accuracy 90%. After the model has been trained, it is tested using 1170 images. The result after testing the setup is presented above through a confusion matrix. From the fig, Confusion matrix, Precision, Recall, and the F1 score is calculated for each class and also the accuracy, macro average precision, recall and F1 score of the entire setup which is stated in the table 5.2 below.

Table 5.2: CNN Setup-4 (without CLAHE)

	Precision	**Recall**	**F1 Score**	**Support**
Normal	0.8344	0.7975	0.8155	316
Pneumonia	0.9263	0.9414	0.9338	854
Macro avg.	0.8803	0.8695	0.8747	
Accuracy	90.25			

5.2.2 CNN with CLAHE



(a) Training Accuracy and Loss (b) Confusion Matrix

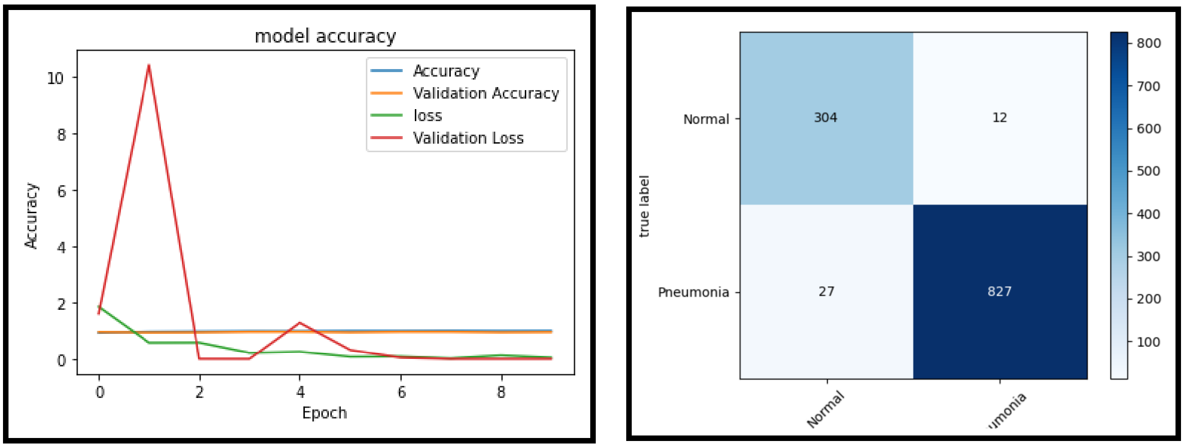
Figure 5.4: Training Accuracy, Loss and Confusion Matrix of Setup-4

From the above fig. 5.4, it is evident that this setup is pretty good. Here, the training and validation accuracy and loss is shown in fig. 5.4a and confusion matrix of setup-4 in fig. 5.4b. This setup-4 of CNN is trained with applying CLAHE. Moreover, the setup converges after 18 epochs and achieves a training accuracy of around 90.7% and validation accuracy 90.6%. After the model has been trained, it is tested using 1170 images. The result after testing the setup is presented above through a confusion matrix. From the fig, Confusion matrix, Precision, Recall, and the F1 score is calculated for each class and also the accuracy, macro average precision, recall and F1 score of the entire setup which is stated in the table 5.3 below.

Table 5.3: CNN Setup-4 (CLAHE)

	Precision	Recall	F1 Score	Support
Normal	0.7834	0.7785	0.7810	316
Pneumonia	0.9182	0.9204	0.9193	854
Macro avg.	0.8508	0.8495	0.8502	
Accuracy	88.21			

5.2.3 Transfer Learning without CLAHE



(a) Training Accuracy and Loss

(b) Confusion Matrix

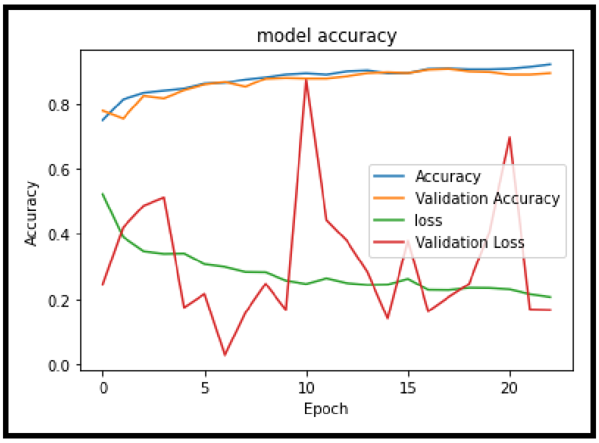
Figure 5.5: Training Accuracy, Loss and Confusion Matrix of VGG16

From the above fig. 5.5, It is evident that this VGG16 pre-trained model is performing well. Here, the training and validation accuracy and loss is shown in fig. 5.5a and confusion matrix of VGG16 in fig. 5.5b. This VGG16 is trained without applying CLAHE. Moreover, VGG16 converges after 5 epochs and achieves a training accuracy of around 98% and validation accuracy 97%. After the VGG16 has been trained, it is tested using 1170 images. The result after testing the pretrained model is presented above through a confusion matrix. From the fig, Confusion matrix, Precision, Recall, and the F1 score is calculated for each class and also the accuracy, macro average precision, recall and F1 score of the entire setup which is stated in the table 5.4 below.

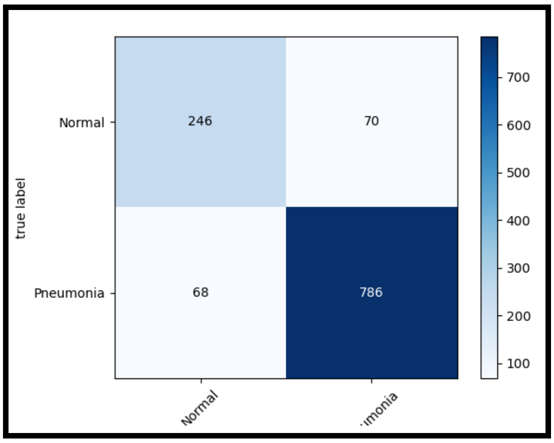
Table 5.4: VGG16 (without CLAHE)

	Precision	Recall	F1 Score	Support
Normal	0.9184	0.9620	0.9397	316
Pneumonia	0.9857	0.9684	0.9770	854
Macro avg.	0.9521	0.9652	0.9584	
Accuracy	96.67			

5.2.4 Transfer Learning with CLAHE



(a) Training Accuracy and Loss



(b) Confusion Matrix

Figure 5.6: Training Accuracy, Loss and Confusion Matrix of VGG16

From the above fig. 5.6, It is evident that this VGG16 pre-trained model is performing well. Here, the training and validation accuracy and loss is shown in fig. 5.6a and confusion matrix of VGG16 in fig. 5.6b. This VGG16 is trained with applying CLAHE. Moreover, VGG16 converges after 6 epochs and achieves a training accuracy of around 99% and validation accuracy 98%. After the VGG16 has been trained, it is tested using 1170 images. The result after testing the pretrained model is presented above through a confusion matrix. From the fig, Confusion matrix, Precision, Recall, and the F1 score is calculated for each class and also the accuracy, macro average precision, recall and F1 score of the entire setup which is stated in the table 5.5 below.

Table 5.5: VGG16 (CLAHE)

	Precision	**Recall**	**F1 Score**	**Support**
Normal	0.8817	0.9430	0.9113	316
Pneumonia	0.9784	0.9532	0.9656	854
Macro avg.	0.9300	0.9481	0.9385	
Accuracy	95.04			

5.3 Results

5.3.1 Ablation Experiments of CNN

Table 5.6: Training Evaluation-1

Setup-1		**Without CLAHE**	**With CLAHE**
Layers	5	Loss: 0.3772	Loss: 0.2925
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.8369	Accuracy: 0.8711
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.3851	Validation Loss: 0.1954
Loss Function	Categorical_Crossentropy		
Epoch	17 (Normal) & 23 (CLAHE)	Validation Accuracy: 0.8672	Validation Accuracy: 0.8914
MaxPooling	2x2		

In table 5.6, Setup-1 is five layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. Apart from these, there are some more hyperparameters in the table above which we have used in this setup. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.8369 and validation accuracy 0.8672. On the contrary, we get training accuracy 0.8711 and validation accuracy 0.8914 by trained with CLAHE on the dataset. So, the training accuracy has improved by applying CLAHE in setup-1.

Table 5.7: Training Evaluation-2

Setup-2		**Without CLAHE**	**With CLAHE**
Layers	7	Loss: 0.2852	Loss: 0.2677
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.8749	Accuracy: 0.8882
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.0704	Validation Loss: 0.3015
Loss Function	Categorical_Crossentropy		
Epoch	16 (Normal) & 24 (CLAHE)	Validation Accuracy: 0.8991	Validation Accuracy: 0.9078
MaxPooling	2x2		

In table 5.7, Setup-2 is seven layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.8749 and validation accuracy 0.8991. On the contrary, we get training accuracy 0.8882 and validation accuracy 0.9078 by trained with CLAHE on the dataset. So, the training accuracy has improved by applying CLAHE in setup-2.

Table 5.8: Training Evaluation-3

Setup-3		**Without CLAHE**	**With CLAHE**
Layers	9	Loss: 0.2297	Loss: 0.2970
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.9067	Accuracy: 0.8679
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.2548	Validation Loss: 0.2468
Loss Function	Categorical_Crossentropy		
Epoch	20 (Normal) & 9 (CLAHE)	Validation Accuracy: 0.9043	Validation Accuracy: 0.8871
MaxPooling	2x2		

In table 5.8, Setup-3 is nine layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.9067 and validation accuracy 0.9043. On the contrary, we get training accuracy 0.8679 and validation accuracy 0.8871 by trained with CLAHE on the dataset. So, the training accuracy has improved without applying CLAHE in setup-3.

Table 5.9: Training Evaluation-4

Setup-4		**Without CLAHE**	**With CLAHE**
Layers	10	Loss: 0.2151	Loss: 0.2278
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.9098	Accuracy: 0.9070
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.5260	Validation Loss: 0.2063
Loss Function	Categorical_Crossentropy		
Epoch	30 (Normal) & 18 (CLAHE)	Validation Accuracy: 0.9078	Validation Accuracy: 0.9060
MaxPooling	2x2		

In table 5.9, Setup-4 is ten layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.9098 and validation accuracy 0.9078. On the contrary, we get training accuracy 0.9070 and validation accuracy 0.9060 by trained with

CLAHE on the dataset. So, the training accuracy has improved without applying CLAHE in setup-4.

Table 5.10: Training Evaluation-5

Setup-5		Without CLAHE	With CLAHE
Layers	12	Loss: 0.3294	Loss: 0.2029
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.8475	Accuracy: 0.9171
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.5093	Validation Loss: 0.0553
Loss Function	Categorical_Crossentropy		
Epoch	16 (Normal) & 30 (CLAHE)	Validation Accuracy: 0.8534	Validation Accuracy: 0.9198
MaxPooling	2x2		

In table 5.10, Setup-5 is twelve layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.8475 and validation accuracy 0.8534. On the contrary, we get training accuracy 0.9171 and validation accuracy 0.9198 by trained with CLAHE on the dataset. Furthermore, the training accuracy has decreased from the previous setup-4 in case of without applying CLAHE. So, this setup is not good.

Table 5.11: Training Evaluation-6

Setup-6		Without CLAHE	With CLAHE
Layers	10	Loss: 0.5818	Loss: 0.5863
Optimizer	Adam		
Learning Rate	0.001	Accuracy: 0.7315	Accuracy: 0.7309
Activation Function	tanh		
Batch Size	32	Validation Loss: 0.6435	Validation Loss: 0.5630
Loss Function	Categorical_Crossentropy		
Epoch	4 (Normal) & 11 (CLAHE)	Validation Accuracy: 0.7319	Validation Accuracy: 0.7353
MaxPooling	2x2		

In table 5.11, Setup-6 is ten layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Adam as an Optimizer with a learning rate of 0.001 and tanh as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.7315 and validation accuracy 0.7319. On the contrary, we get training accuracy 0.7309 and validation accuracy 0.7353 by trained with CLAHE on the dataset. In both cases, there is a significant reduction in training accuracy from previous setup-4. So, this setup is not good.

Table 5.12: Training Evaluation-7

Setup-7		Without CLAHE	With CLAHE
Layers	10	Loss: 0.3398	Loss: 0.3560
Optimizer	SGD		
Learning Rate	0.001	Accuracy: 0.8406	Accuracy: 0.8300
Activation Function	ReLU		
Batch Size	32	Validation Loss: 0.3890	Validation Loss: 0.2632
Loss Function	Categorical_Crossentropy		
Epoch	7 (Normal) & 6 (CLAHE)	Validation Accuracy: 0.8431	Validation Accuracy: 0.8526
MaxPooling	2x2		

In table 5.12, Setup-7 is ten layers of the custom architecture of Neural Network with 2-way softmax activation. In this setup, we have used Sigmoid as an Optimizer with a learning rate of 0.001 and ReLU as an activation function for Convolutional Layer. From the table above, we can see that if we trained the model in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.8406 and validation accuracy 0.8431. On the contrary, we get training accuracy 0.8300 and validation accuracy 0.8526 by trained with CLAHE on the dataset. In both cases the training accuracy has increased from Setup-5, but not yet better than Setup-4.

So from the above ablation experiment, we can say that setup-4 is optimum for two conversions of the dataset in training evaluation.

5.3.2 Training Experiment of VGG16

Table 5.13: Training Evaluation with VGG16

Hyperparameters of VGG16		Without CLAHE	With CLAHE
Optimizer	Adam	Loss: 0.2570	Loss: 0.0742
Batch Size	32	Accuracy: 0.9839	Accuracy: 0.9943
Loss Function	Categorical_Crossentropy	Validation Loss: 1.2858	Validation Loss: 1.5628
Epoch	5 (Normal) & 6 (CLAHE)	Validation Accuracy: 0.9747	Validation Accuracy: 0.9872

For this model, VGG16 on ImageNet dataset have used as the classifier. Most of the layers have untouched. A new prediction layer replaces only the final layer with two-way softmax activation. In this pretrained model (VGG16), we have used Adam as an Optimizer with a learning rate of 0.001 and 'Categorical Crossentropy' as loss function. From the table above, we can see that if we trained VGG16 in the above configuration without applying CLAHE on the dataset, then we get training accuracy 0.9839 and validation accuracy 0.9747. On the contrary, we get training accuracy 0.9943 and validation accuracy 0.9872 by trained with

CLAHE on the dataset. In both cases the accuracy has increased from CNN (setup-4), and accuracy has improved in training evaluation by applying CLAHE in VGG16.

5.3.3 Comparison of results between CNN and VGG16

Table 5.14: Comparison of results between CNN and VGG16

Model	Dataset	Accuracy	Precision	F1 Score	Recall
CNN (Setup-4)	Without Preprocess (CLAHE)	90.25	0.8803	0.8747	0.8695
VGG16	Without Preprocess (CLAHE)	96.67	0.9521	0.9584	0.9652

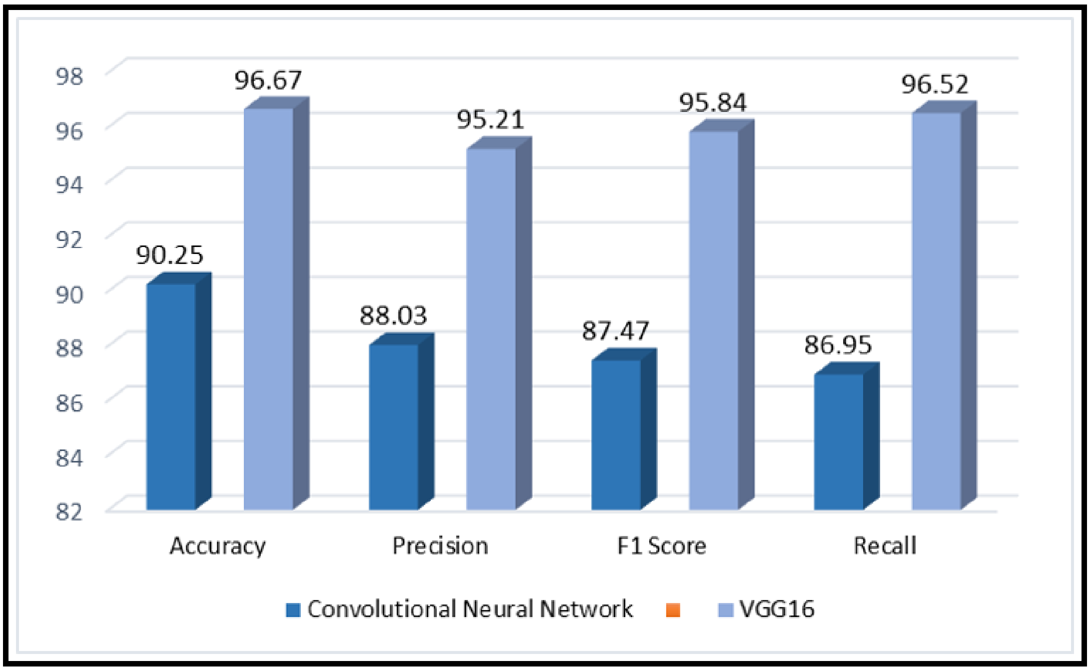


Figure 5.7: Comparison of results between CNN and VGG16

An analysis of the table and graph above reveals several essential aspects which is given below-

- In the case of deep learning, it is not necessary to preprocess the dataset. We have good accuracy in both CNN and VGG16 without preprocessing (CLAHE).
- It can be seen from the table above that not only the result of testing accuracy is good with VGG16 pre-trained model but also the result of F1 score, precision, Recall is better than CNN.

We can explore the above table and graph through the ablation experiment and say that we have got the satisfactory testing accuracy using VGG16 pre-trained model.

Chapter 6

Conclusion

Our goal is to propose a deep learning-based approach to classify pneumonia from chest X-ray images using Deep Convolutional Neural Networks (DCNN) and transfer learning (VGG16) using with preprocessing (CLAHE) and without preprocessing, and compared that outcomes between those models by evaluation metric to see which model gives the optimal results. But there are some differences in the results by training the model with CLAHE and without CLAHE. In the case of applying without CLAHE, VGG16 gave good results than CNN, which is 96.67%. On the contrary, using with CLAHE, none of the proposed models (CNN & VGG16) has been susceptible to give satisfactory results. In this context, we can say that image processing is not necessary for deep learning, and CNN doesn't work well for any setup. It works excellently for specific setup (Setup-4) on CXI datasets.

References

[1] "What is classification in deep learning?." https://www.edureka.co/blog/classification-in-machine-learning/. Accessed: 2020-9-14.

[2] K. E. Asnaoui, Y. Chawki, and A. Idri, "Automated methods for detection and classification pneumonia based on x-ray images using deep learning," *arXiv preprint arXiv:2003.14363*, 2020.

[3] S. Rajaraman, S. Candemir, G. Thoma, and S. Antani, "Visualizing and explaining deep learning predictions for pneumonia detection in pediatric chest radiographs," in *Medical Imaging 2019: Computer-Aided Diagnosis*, vol. 10950, p. 109500S, International Society for Optics and Photonics, 2019.

[4] I. M. Baltruschat, H. Nickisch, M. Grass, T. Knopp, and A. Saalbach, "Comparison of deep learning approaches for multi-label chest x-ray classification," *Scientific reports*, vol. 9, no. 1, pp. 1–10, 2019.

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

[6] "Understanding of convolutional neural network (cnn) — deep learning." t.ly/7xLN. Accessed: 2020-9-14.

[7] "Why are convolutional neural networks good for image classification?." https://medium.com/datadriveninvestor/why-are-convolutional-neural-networks-good-for-image-classificat
Accessed: 2020-9-14.

[8] "Vgg neural networks: The next step after alexnet." https://towardsdatascience.com/vgg-neural-networks-the-next-step-after-a
Accessed: 2020-9-14.

[9] N. Ansari, A. R. Faizabadi, S. Motakabber, and M. I. Ibrahimy, “Effective pneumonia detection using res net based transfer learning,” 2020.

[10] “What are evaluation metrics?.” t.ly/ytiz. Accessed: 2020-9-14.