CAPSTONE PROJECT

Project Title: X-Ray Pneumonia Prediction

PGA-02

Abstract

The goal of the project is to predict whether a patient is diagnosed with Pneumonia disease or not. The data that is used consists of X-Ray images that are divided into 2 folders train and test. Several deep learning models are trained using train dataset & applied to test dataset in order to evaluate the model performance. These performance measures are then compared to determine which model is best in prediction of Pneumonia disease in patients.

Submitted by - Begum Zubeda Abbasuddin

Table of Content

Capstone Project – PGA-02

X-Ray Pneumonia Prediction

1. Introduction

Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patient's routine clinical care.

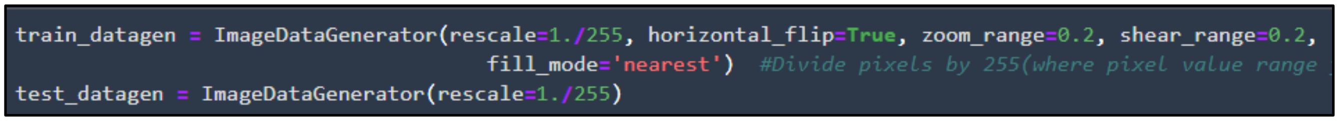
For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

The Dataset

The dataset is organized into 2 folders (train, test) and contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal).

2. Normalizing Data & Image Augmentation

The X-Ray Images are converted into pixels ranging between 0-255 of shape (height, width, color channel (RGB)) so that it can be processed by the model.



```
train_datagen = ImageDataGenerator(rescale=1./255, horizontal_flip=True, zoom_range=0.2, shear_range=0.2,
                                    fill_mode='nearest')  #Divide pixels by 255(where pixel value range
test_datagen = ImageDataGenerator(rescale=1./255)
```

The process of standardizing features present in the data in a fixed range or same scale is referred to as normalization. It is performed during the data pre-processing to handle highly varying magnitudes or values or units. In this case, Normalization of data is done by scaling image pixels. Here, each pixel is divided by 255 pixel value such that pixel values ranges between 0 and 1, because 0-255 RGB range pixel values would be too high for our models to process (given a typical learning rate).

Image Augmentation is referred as a process of applying various random transformation on images, which can be applied for following reasons:

- Make the most of our few training examples.
- Helps prevent overfitting and helps the model generalize better. Overfitting happens when a model exposed to too few examples learns patterns that do not generalize to new data.

Several Image Augmentation that can be applied are:

- 'horizontal_flip' is for randomly flipping half of the images horizontally.

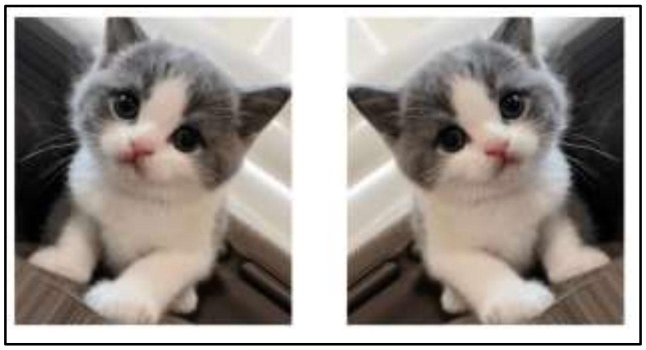


Figure 2.1

- ‘shear_range’ is for randomly applying shearing transformations.

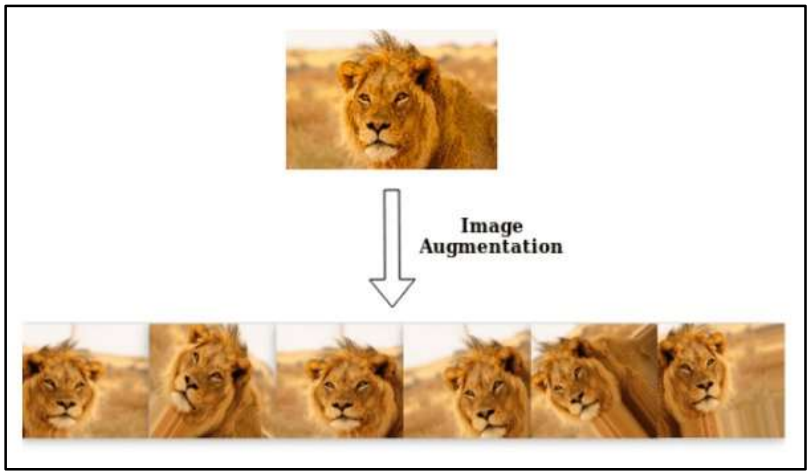


Figure 2.2

- ‘zoom_range’ is for randomly zooming inside pictures.

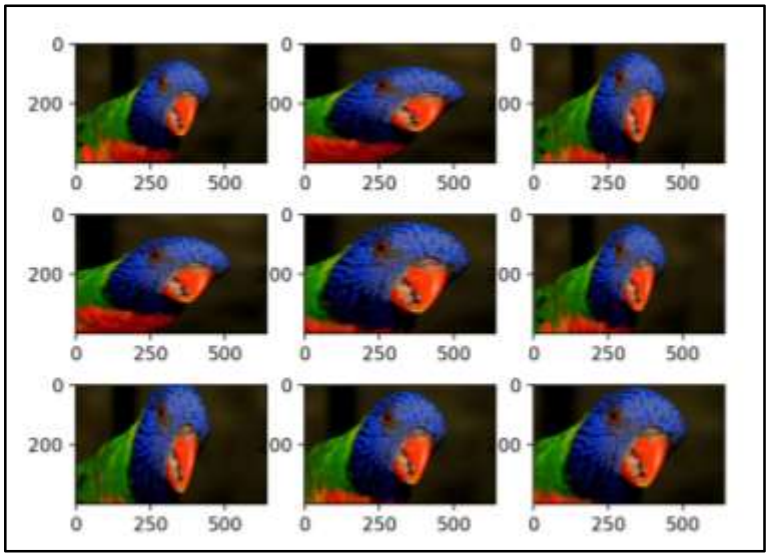
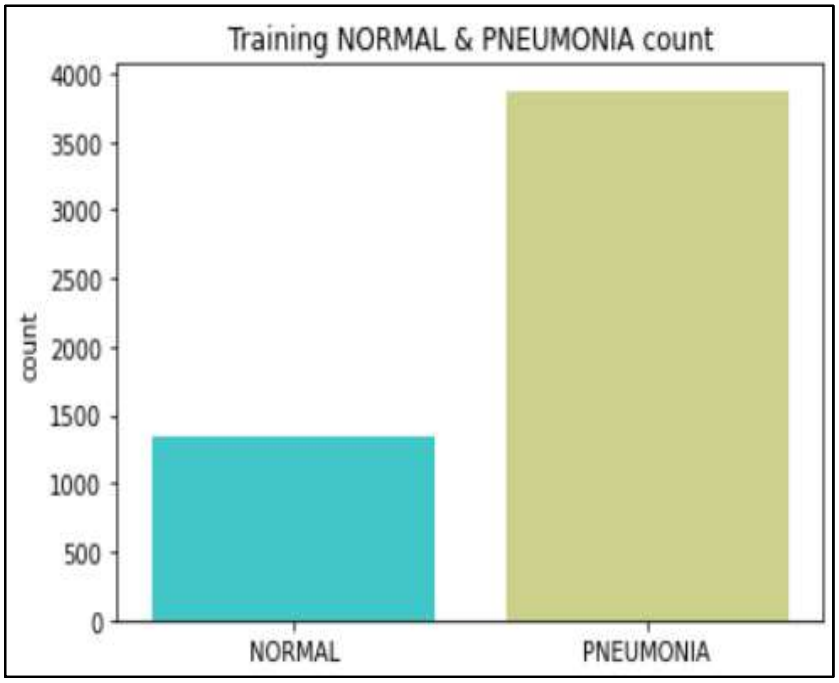


Figure 2.3

- ‘fill_mode’ is the strategy used for filling in newly created pixels, which can appear after a rotation or a width/height shift.

3. Data Visualization



The graph on the left represents the total number of Normal and Pneumonia patients in the training dataset. We can clearly see that there are more number of Data/X-Ray Images for Pneumonia patients than Normal patients for training the model, that results in the problem of Imbalanced Data.

Figure 3.1

The graph on the right represents the total number of Normal and Pneumonia patients in the test dataset to which the trained model is going to be applied. We can clearly see that there are more number of Data/X-Ray Images for Pneumonia patients than Normal patients.

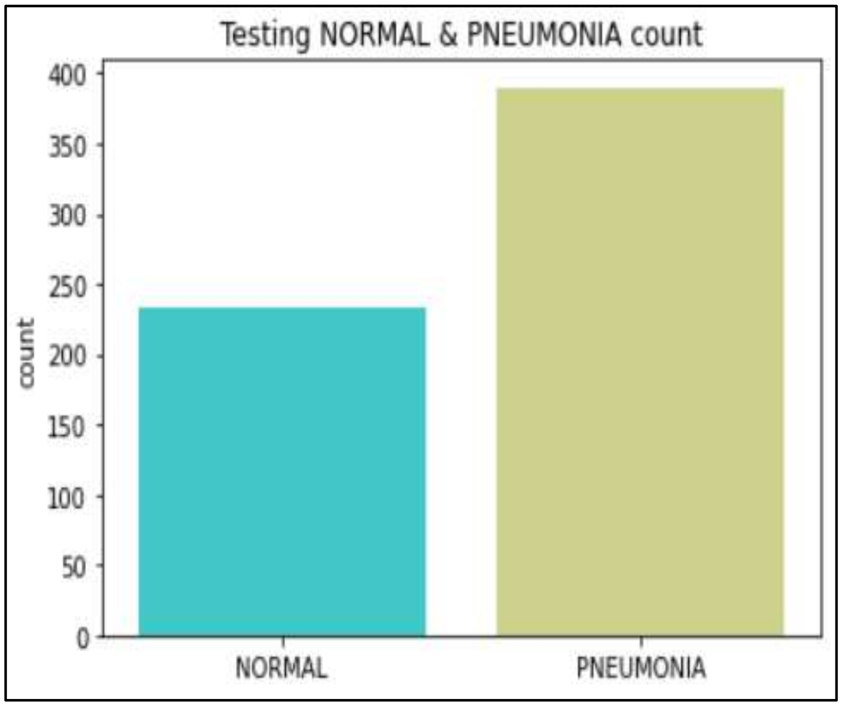


Figure 3.2

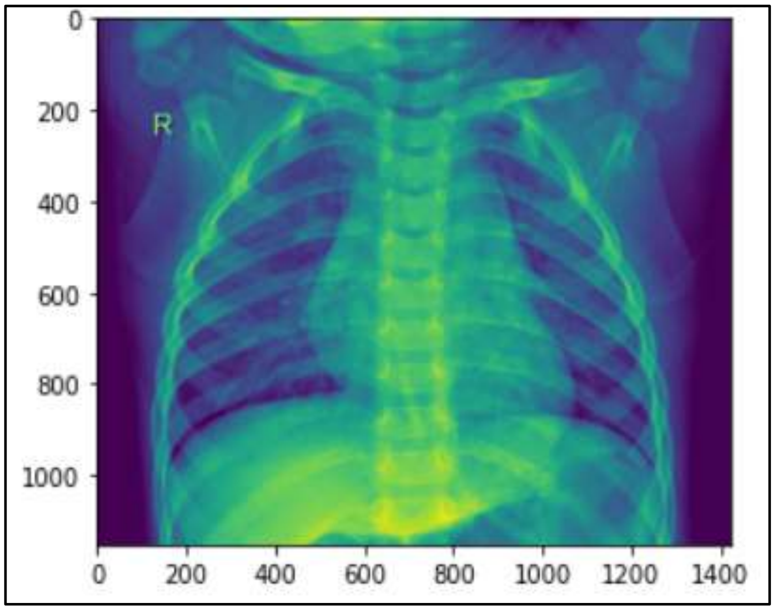


Figure 3.3

The image plot on the left represents X-ray Image for Normal patient. This normal chest X-ray depicts clear lungs without any areas of abnormal opacification in the image.

The image plot on the right represents X-ray Image for Pneumonia patient. We can see that the image manifests with a more diffuse "interstitial" pattern in both lungs.

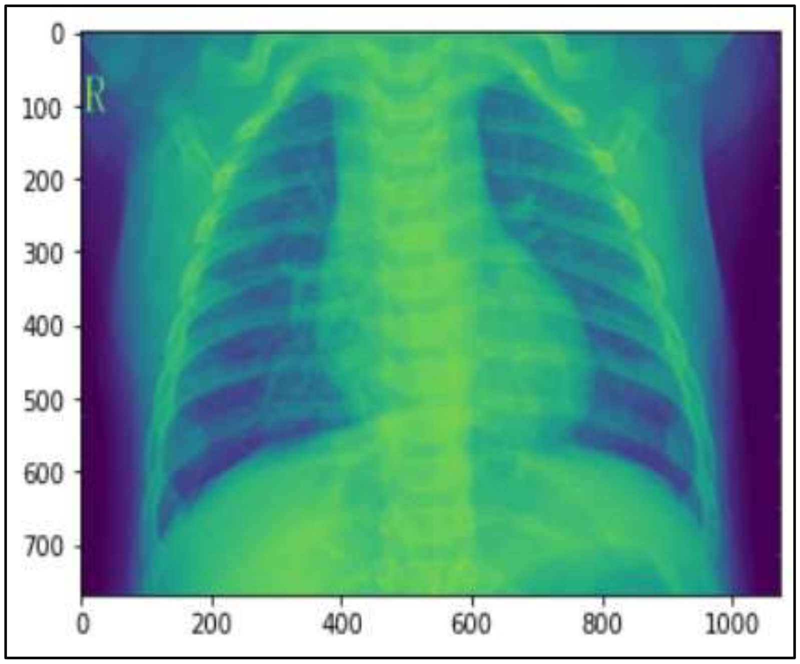


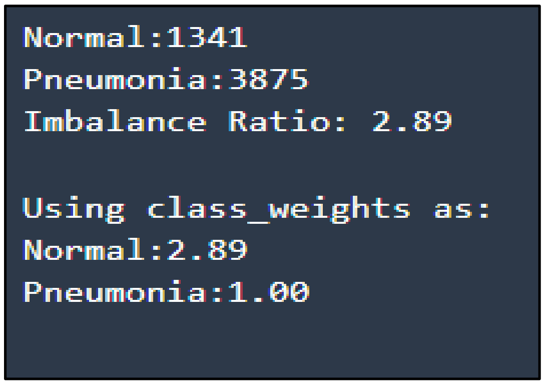
Figure 3.4

4. Handling Imbalanced Data

In a classification problem, a good model is the one that can predict or classify data well.

Imbalanced data is referred to a scenario where the dataset has unequal number of observations or data for each class or in other words difference in the number of observations or data for classes is very huge like we have seen in figure 3.1. This problem can lead to biasness for majority class and when the model is trained with such data it cannot generalize the new data well where bias refers to difference between average prediction of a model & the correct value which it is trying to predict. Therefore, when the optimization is applied while training it basically tries to reduce the misclassification error & gives more priority to the majority class data so that the total error is minimum & the accuracy is maximum, but for minority class data the performance is not so good.

In order to solve this problem we can apply various techniques like undersampling the majority class data, oversampling the minority class data and the technique we used is setting the class weights where we assign higher weights for minority class using the imbalanced ratio while training our model with train dataset, and this is the most efficient way where we can keep the data as it is without losing or adding extra data.


```
Normal:1341
Pneumonia:3875
Imbalance Ratio: 2.89

Using class_weights as:
Normal:2.89
Pneumonia:1.00
```

Figure 4.1

5. Convolutional Neural Network (CNN)

Neural Network is basically a type of machine learning algorithm that is inspired by human biological brain. The goal is to solve the problem in a similar way as the human brain does. A Convolutional Neural Network (CNN) is a type of neural network that specializes in Image Recognition & Computer Vision tasks.

CNN has several types of layers:

- **Input Layer:** This layer consists of Image in the form of pixels of shape (height, width, color channels (RGB)).

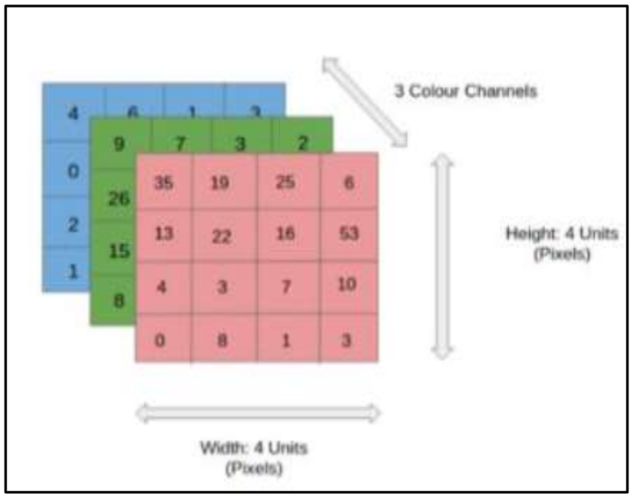


Figure 5.1

- **Convolutional Layer:** A filter/kernel that is in the form of matrix scans few pixels at a time creates a feature map that predicts the class that each feature belongs to.

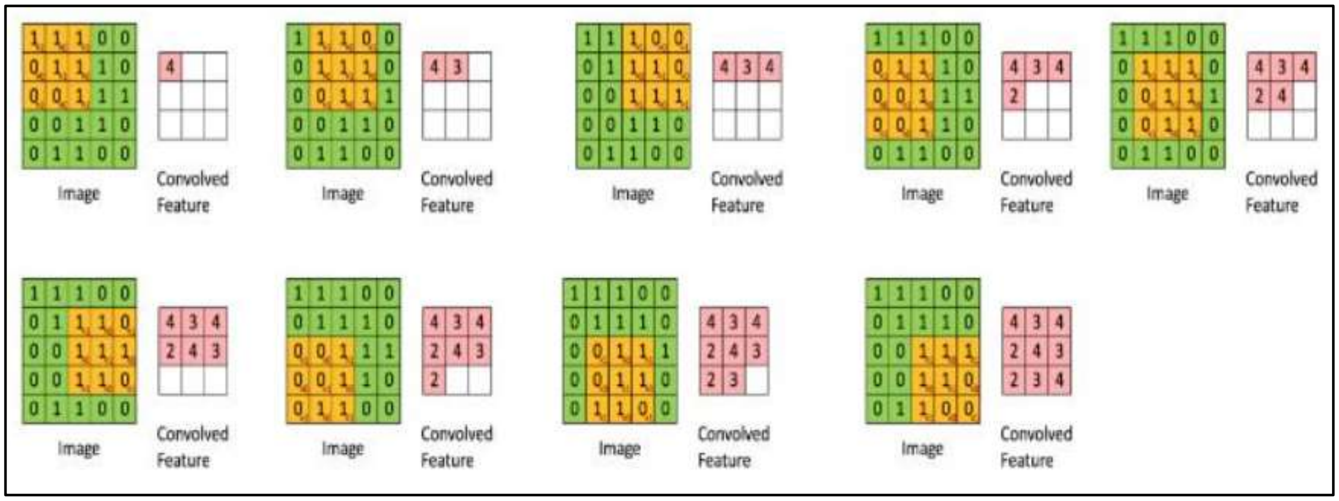
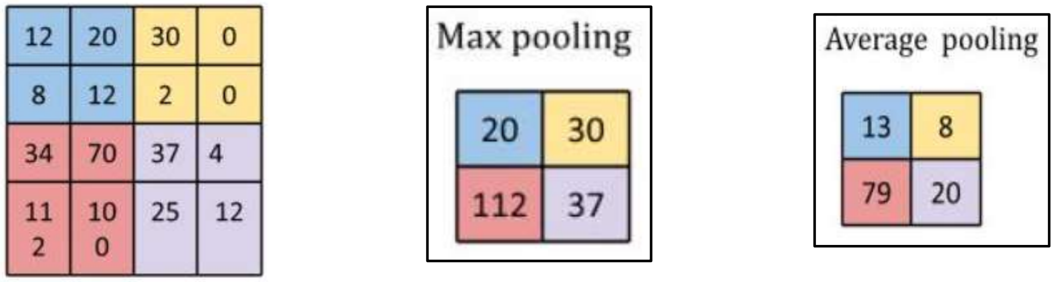


Figure 5.2

- **Pooling Layer:** This layer reduces the amount of information in each feature that was obtained in the convolutional layer while maintaining the most important information. Several types of pooling that are used are, Max pooling that calculates the maximum value for each patch of the feature map, Average pooling that calculates the average value for each patch of the feature map & Global Max pooling that calculates the overall maximum value for feature map.



- **Batch Normalization Layer:** This layer helps to coordinate the update of multiple layers in the model. It scales the output of the layer, specifically by standardizing the activations of each input variable per mini-batch, such as the activations a node from the previous layer which means that assumptions the subsequent layer makes about the distribution of inputs during the weight update will not change, at least not dramatically. This has the effect of stabilizing and speeding-up the training process of deep neural networks.

  Activations or activations functions are mathematical gates between input feeding the current neuron and output going to the next layer. Some of the activations are ReLU or rectified linear activation function is a linear function that will output the input directly if it is positive, otherwise, it will output zero and in this project it is used because pixels are always positive, Sigmoid activation function which is the logistic function used for binary classification problems.

- **Dropout Layer:** This layer is used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

- **Flatten Layer:** This layer converts the multi-dimensional output to 1D vector of pixels.

- **Dense Layer:** A dense layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer.

6. Optimization

Optimization is a technique that finds the values of parameter/weights such that is minimizes of maximizes the objective function of interest. In this project, the objective function is an error/cost function that is needed to be minimized.

Several optimization techniques that is used are:

- **Stochastic Gradient Descent:**
  In this technique, weights are initialized with random values, a single sample is selected in random from the dataset, then for each sample the algorithms selects optimized weights such that it reduces error/cost function, at last weights are updated & the model is evaluated.

- **Adam (Adaptive Moment Estimation):**
  This optimzation technique basically uses momentum which pushes out our algorithm or model from locally optimal solution or local minima along with RMSProp which is an adaptive learning algorithm that tries to improve AdaGrad which is a gradient-based optimization technique that adapts the learning rate to the parameters(weights), performing smaller updates (i.e. low learning rates) for parameters(weights) associated with frequently occurring features, and larger updates (i.e. high learning rates) for parameters associated with infrequent features. Here, learning rate refers to how quickly an algorithm updates its parameters(weights) or size of the step taken by the algorithm to reach global minima (overall minimum value(error) of data points in the curve).

7. Steps Performed

1. Installing the necessary packages.

- matplotlib
- seaborn
- tensorflow
- numpy
- pandas

2. Importing the necessary libraries.

3. Create an Image data generator.

- Generator will normalize/scale Image data pixels & augment images.

4. Loading image data.

- There are 5,863 X-Ray images are organized into 2 folders (train, test) and contains subfolders for each image category (Pneumonia/Normal).
- The images are loaded from the train & test directories with fixed (width, height) pixels.
- Image data generator is applied on these image data.

5. Developing various plots for exploratory data analysis.

6. Create custom function to handle imbalanced training data.

- Finding imbalanced ratio and setting higher class weights for minority class data while training.

7. Building models.

- *Basic Convolutional Neural Network (CNN)*: CNN is a type of neural network dealing with computer vision tasks. Built a Sequential Model where output of one layer is fed as an input to the another layer, where 3 blocks of Conv2D, MaxPool2D, BatchNormalization, Dropout layers are added along with Flatten & Dense Layers.
- *VGG16*: VGG16 is a pre-trained deep learning vision model architecture with 16 layers and 5 blocks of Conv2D & MaxPool2D layers along with Flatten & Dense layers.
- *ResNet50*: ResNet or Residual Network is a pre-trained deep learning model architecture which was built to solve the problem of the vanishing/exploding gradient. ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer.
- *InceptionV3:* Inception v3 is a deep learning model architecture assisting in image analysis and object detection consisting 42 layers of Conv2D, BatchNormalization, MaxPooling, AveragePooling, Mixed layers.

8. Predicting values

The trained models are applied on the generated test image data that return probabaility as output. The probabilities are converted to 0 - Normal & 1 - Pneumonia by setting a threshold of greater than 50% as 1 and less than equal to 50% as 0.

9. Evaluating model performance based on predictions.

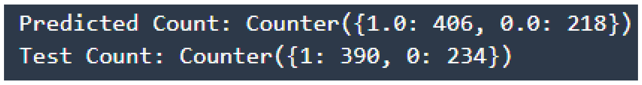
Comparing counts of classes for predicted and actual values, metrics such as accuracy, precision, recall & f1-score is used for evaluating models performance.

10. Interpreting results.

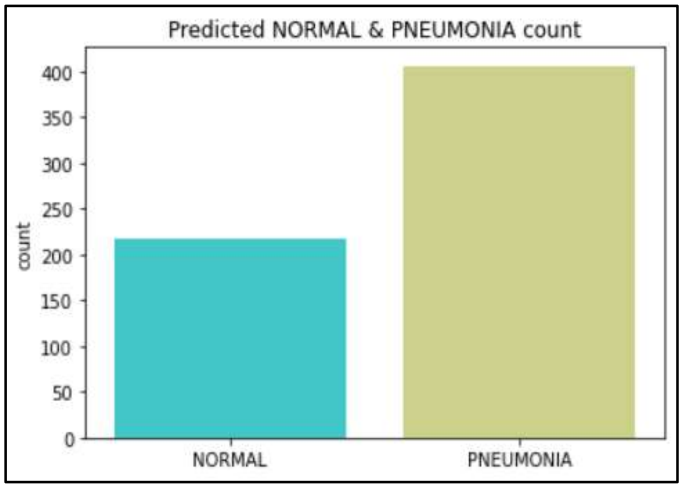
8. Results

Basic Concolutional Neural Network (CNN):

Figure 8a



```
Predicted Count: Counter({1.0: 406, 0.0: 218})
Test Count: Counter({1: 390, 0: 234})
```



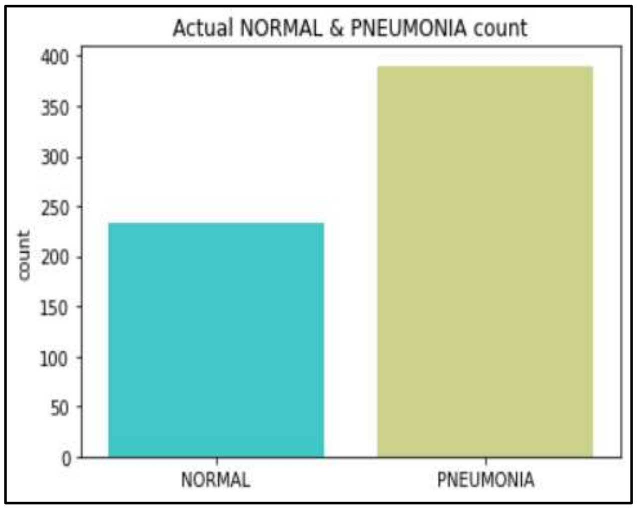


Figure 8b

From the count we can see that around 96.05% of Pneumonia patients were predicted correctly and 93.59% of Normal patients were predicted correctly.

Plotting the training & validation loss and accuracy results for

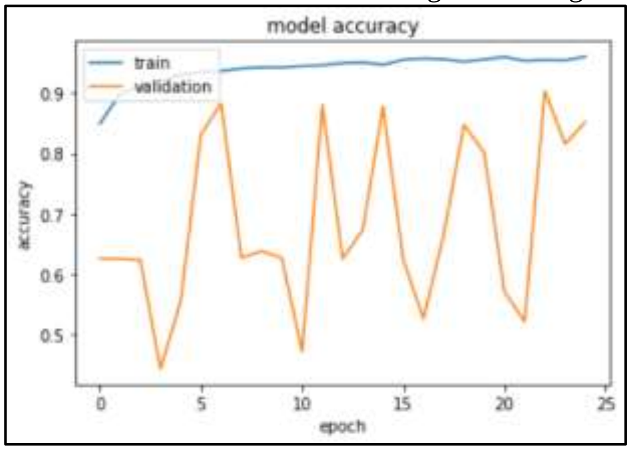


Figure 8.1

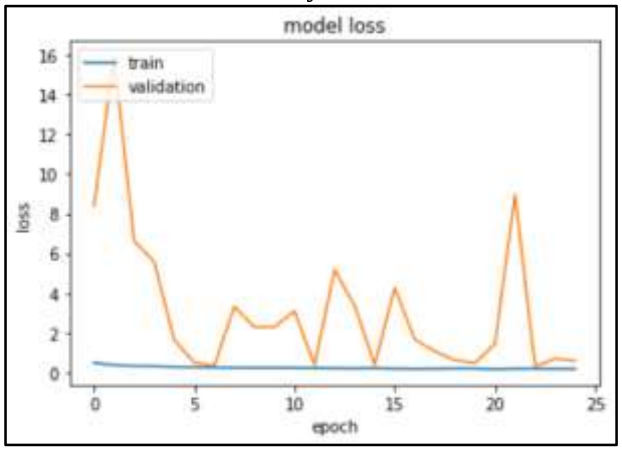
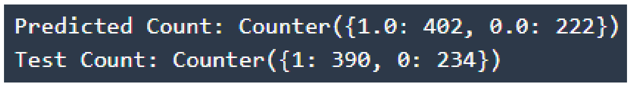


Figure 8.2

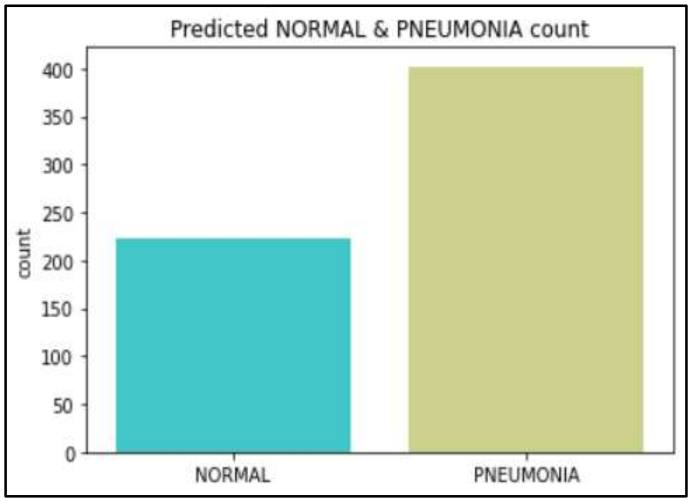
From the above line plots, it can be seen that model is not stable and is overfitting at some epochs.

VGG16:



```
Predicted Count: Counter({1.0: 402, 0.0: 222})
Test Count: Counter({1: 390, 0: 234})
```

Figure 8c



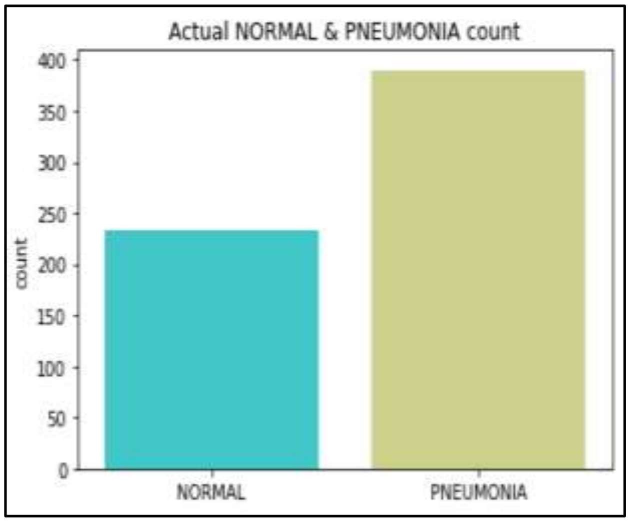
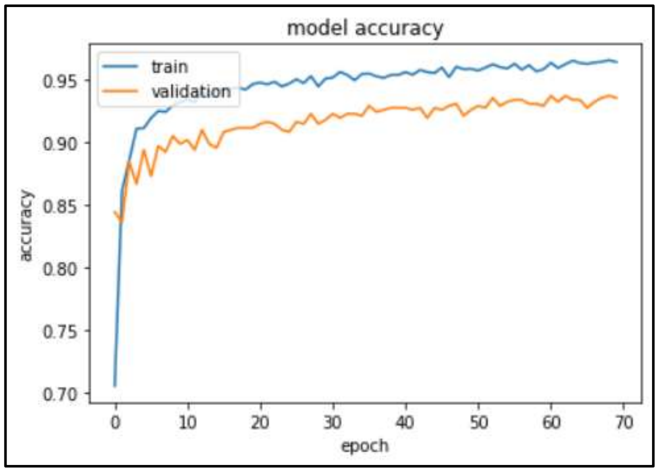


Figure 8d

From the count we can see that around 97% of Pneumonia patients were predicted correctly and 94.87% of Normal patients were predicted correctly.

Plotting the training & validation loss and accuracy results for



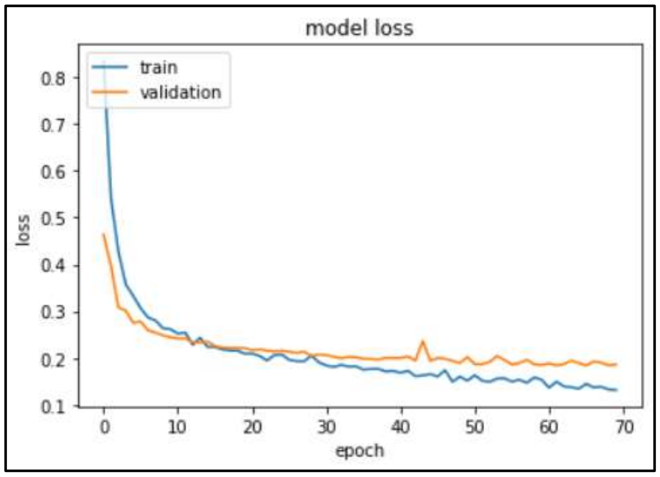
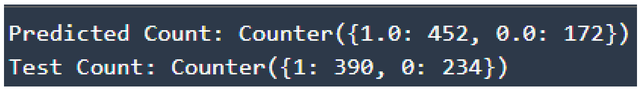


Figure 8.3

Figure 8.4

From the above line plots, it can be seen that model is stable, there is not much difference between training & validation accuracy & loss.

ResNet50:



```
Predicted Count: Counter({1.0: 452, 0.0: 172})
Test Count: Counter({1: 390, 0: 234})
```

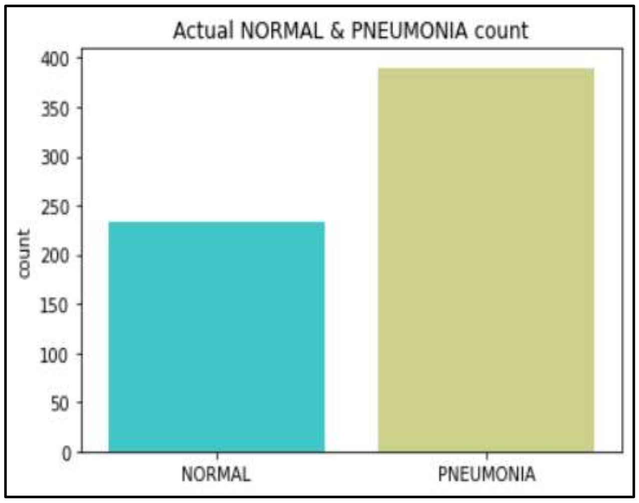
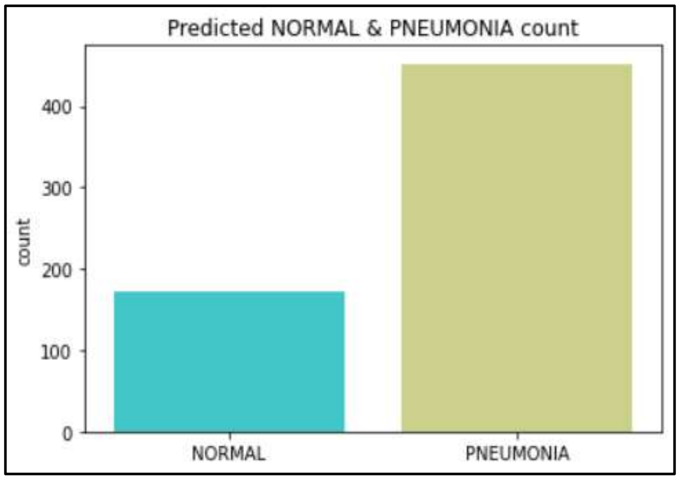


Figure 8f

Figure 8e



From the count we can see that around 86.28% of Pneumonia patients were predicted correctly and 73.04% of Normal patients were predicted correctly.

Plotting the training & validation loss and accuracy results for

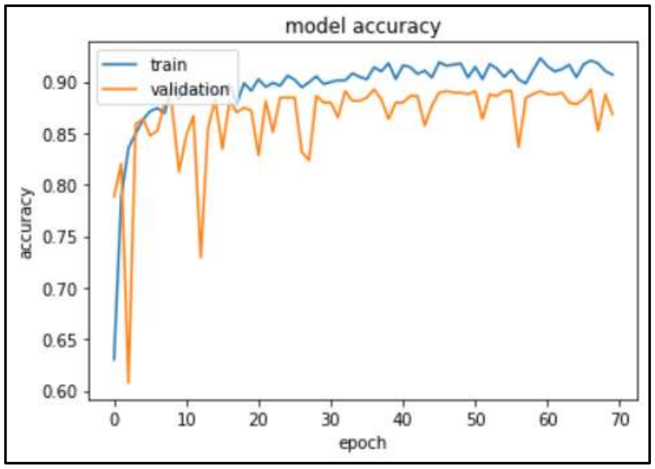


Figure 8.5

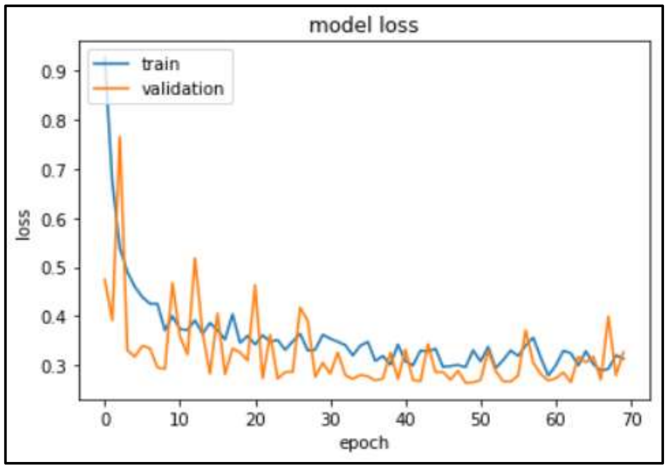
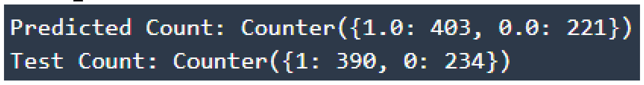


Figure 8.6

From the above line plots, it can be seen that model is not overfitting much but model is not so stable as well.

InceptionV3:



```
Predicted Count: Counter({1.0: 403, 0.0: 221})
Test Count: Counter({1: 390, 0: 234})
```

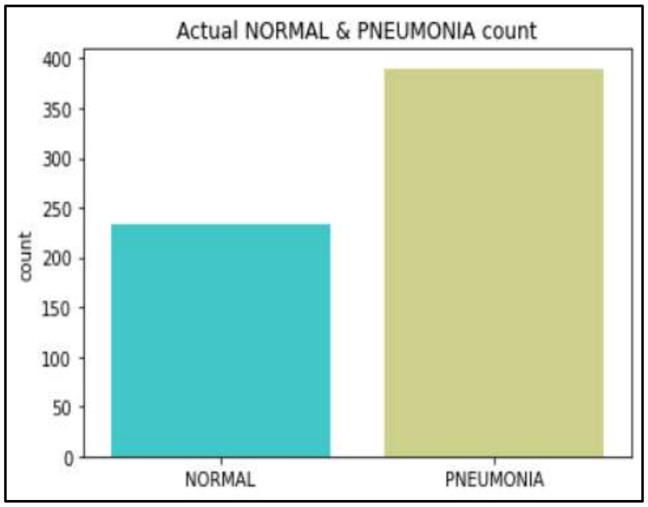
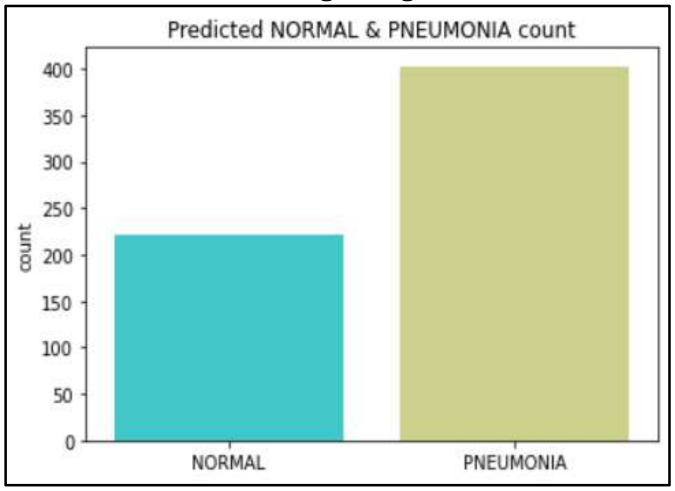


Figure 8h

Figure 8g



From the count we can see that around 96.77% of Pneumonia patients were predicted correctly and 94.44% of Normal patients were predicted correctly.

Plotting the training & validation loss and accuracy results for

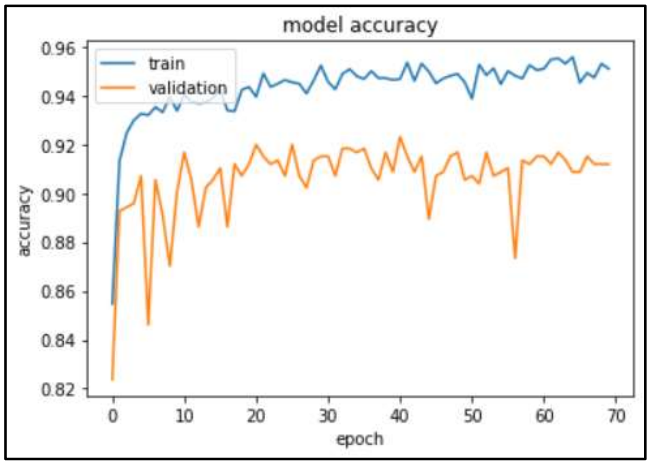


Figure 8.7

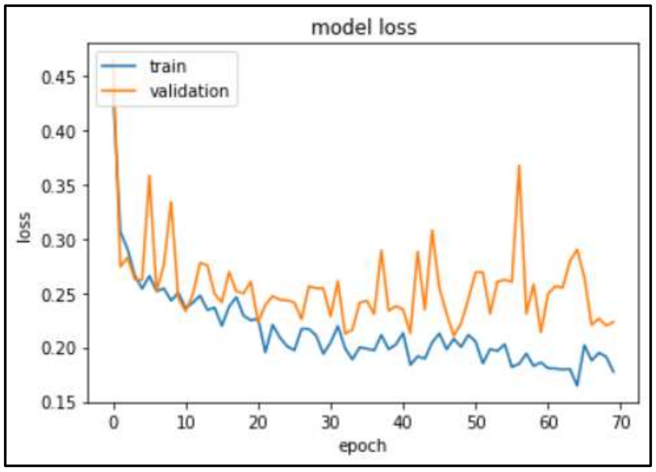
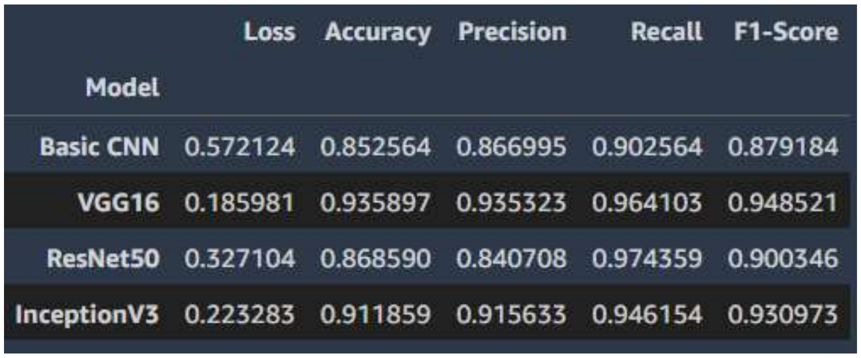


Figure 8.8

From the above line plots, it can be seen that model is not so stable and is overfitting at some epochs.

Evaluation:



Model	Loss	Accuracy	Precision	Recall	F1-Score
Basic CNN	0.572124	0.852564	0.866995	0.902564	0.879184
VGG16	0.185981	0.935897	0.935323	0.964103	0.948521
ResNet50	0.327104	0.868590	0.840708	0.974359	0.900346
InceptionV3	0.223283	0.911859	0.915633	0.946154	0.930973

Plotting the performance measures Loss – Loss function tells how good our model is in predictions & it is minimum if predictions are closer to the actual values, Binary cross entropy used compares each of the predicted probabilities to actual class output which can be either 0 or 1, calculates the score that penalizes the probabilities based on the distance from the expected value i.e. how close or far from the actual value. Accuracy – Percent of correct predictions, Recall – Percent of positive (Pneumonia) predictions we were able to catch, Precision – Percent of correct positive predictions, F1-Score – Test accuracy.

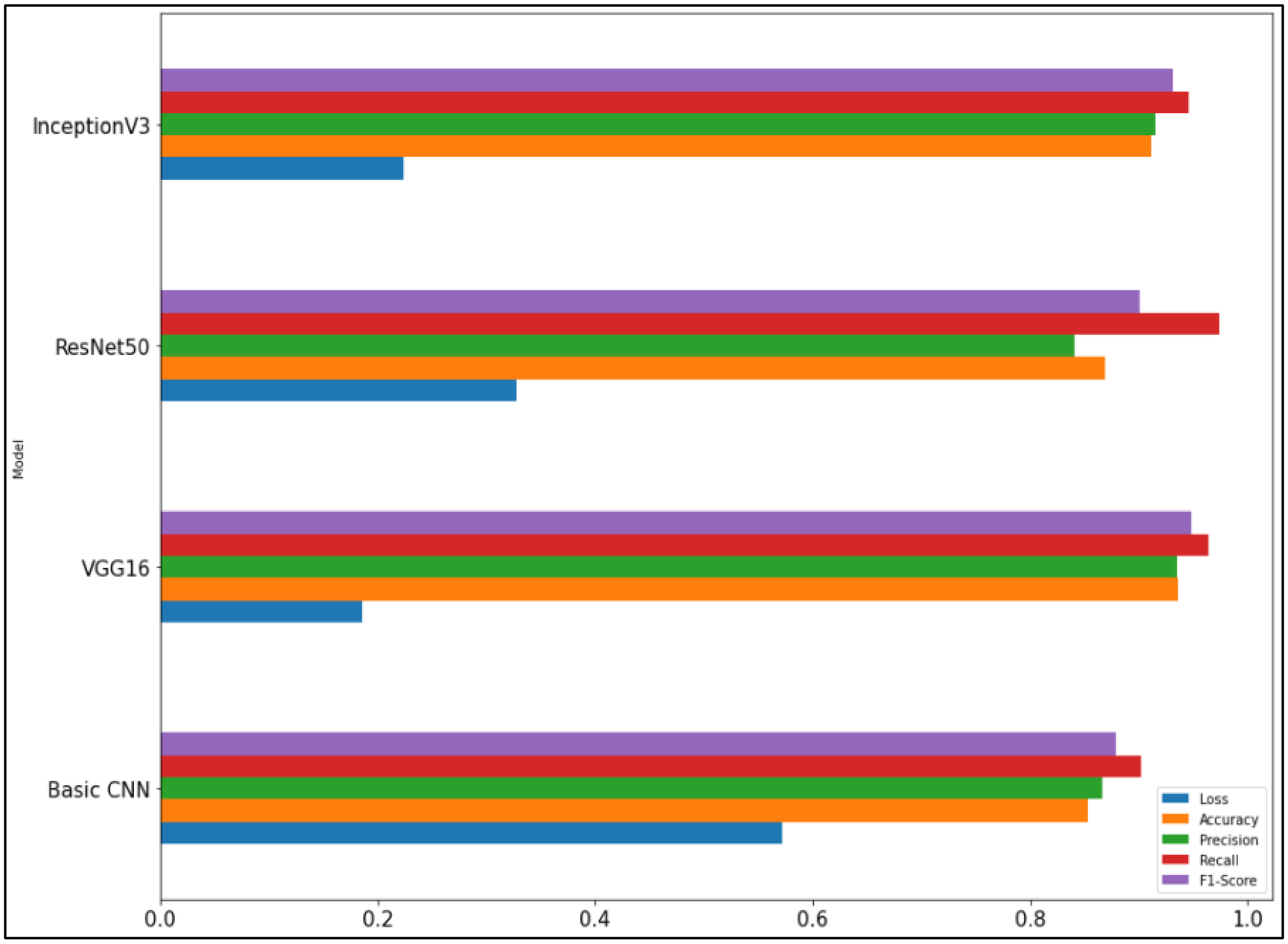


Figure 8.9

From the above results, we can conclude that VGG16 is the most stable and better performing model as compared to Basic Convolutional Neural Network (CNN), RestNet50, InceptionV3 models. Hence, VGG16 is the preferred model for the prediction of Pneumonia patients.

9. Conclusion

Pneumonia is an infection in one or both lungs caused by bacteria, viruses, or fungi. The infection leads to inflammation in the air sacs of the lungs, which are called alveoli. The alveoli fill with fluid or pus, making it difficult to breathe. Both viral and bacterial pneumonia are contagious. This means they can spread from person to person through inhalation of airborne droplets from a sneeze or cough.

The X-Ray of Pneumonia patients exhibits a focal lobar consolidation, in this case in the right upper lobe (white arrows), or manifests with a more diffuse "interstitial" pattern in both lungs like we have seen in Figure 3.4.

With the help of the preferred model that recognizes patterns in the X-Ray images, we can predict probability of a patient having Pneumonia or not, with around 94% accuracy using their lungs X-Ray Image. The application of computer-assisted diagnostic models to help radiologists to more quickly and accurately interpret chest X-ray images, to screen and classify Pneumonia patients, is highly required. The integration of this algorithm into the clinical system could help health institutions to advance patient care by reducing the time to diagnosis and increasing access to chest radiograph interpretation, as well as mitigating the shortage of expert radiologists in remote areas.

10. References

Theory References:

a) https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

b) https://datascience.stackexchange.com/questions/45165/how-to-get-accuracy-f1-precision-and-recall-for-a-keras-model

c) https://www.healthline.com/health/pneumonia#what-is-pneumonia

d) https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

e) https://iq.opengenus.org/inception-v3-model-architecture/

f) https://iq.opengenus.org/resnet50-architecture

g) https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks

h) https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/

i) https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/

j) https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/

k) https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

l) https://www.geeksforgeeks.org/intuition-of-adam-optimizer/

m) https://datascience.stackexchange.com/questions/37987/why-imbalanced-data-set-will-bias-the-prediction-model-towards-the-more-common-c

n) https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

o) https://learning.imarticus.org/

p) https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/

Coding References:

a) https://keras.io/api/applications/vgg/

b) https://keras.io/api/applications/inceptionv3/

c) https://keras.io/api/applications/resnet/

d) https://keras.io/api/