Detecting Lung Disease using Deep Neural Network

1. Overview:

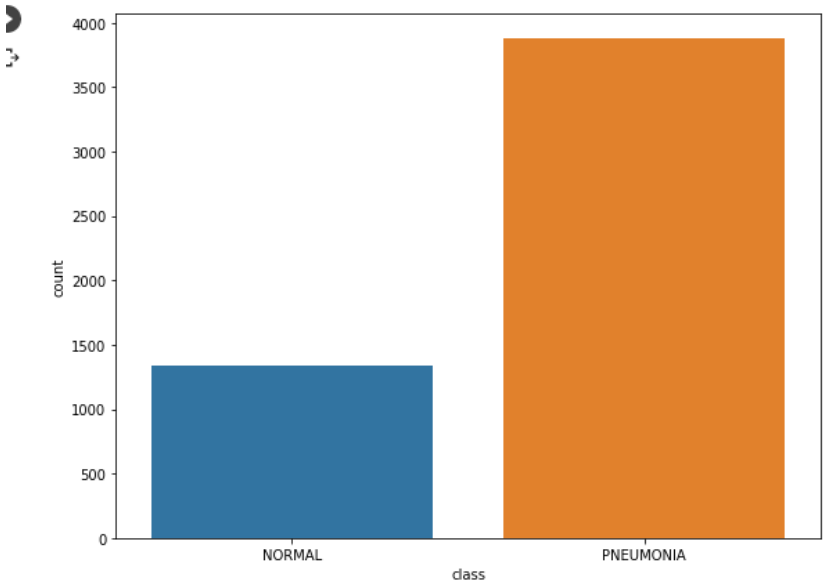
The problem statement is to detect whether a patient is having pneumonia or not based on their X-Ray images. The data is extracted from Kaggle which consist of a test dataset, train dataset and a val dataset. The model is trained using the train and test dataset and then use the model to predict the images in the val dataset to see if the model is able to predict the images correctly. Here I have used different networks to train the model. ResNet50, ResNet32, VGGN16 & VGGN19. We will be comparing the performance metrics of these models with the training accuracy and testing accuracy and try to predict the result for the data in the val dataset. The code is run in Google Colab as gpu is required to train the model with the huge dataset.

2. Dataset Description :

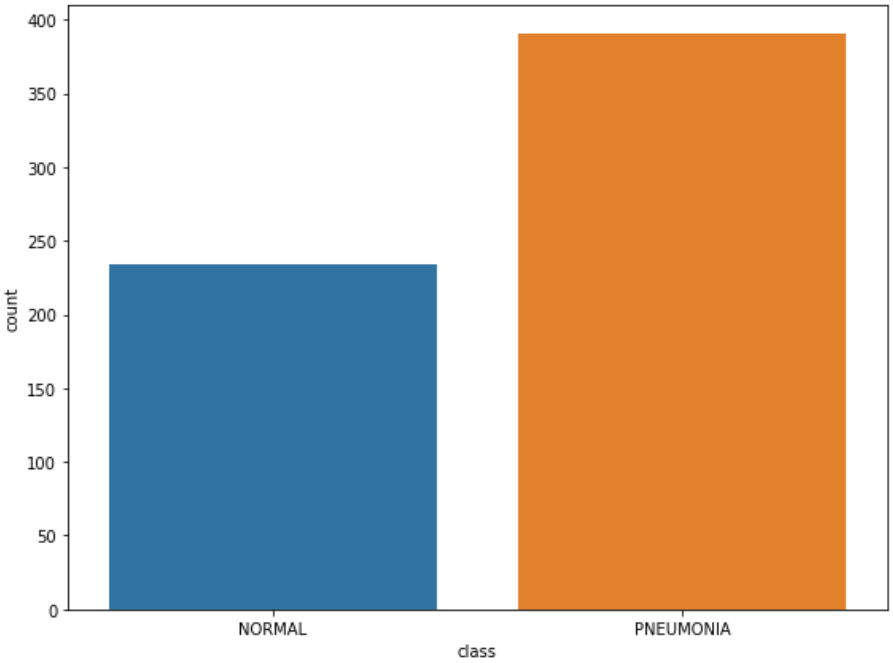
The dataset consist of test , train and val having images of normal and pneumonia X-Ray images.

Plotted here the number of Normal and Pneumonia images available in Train & Test data

Train data



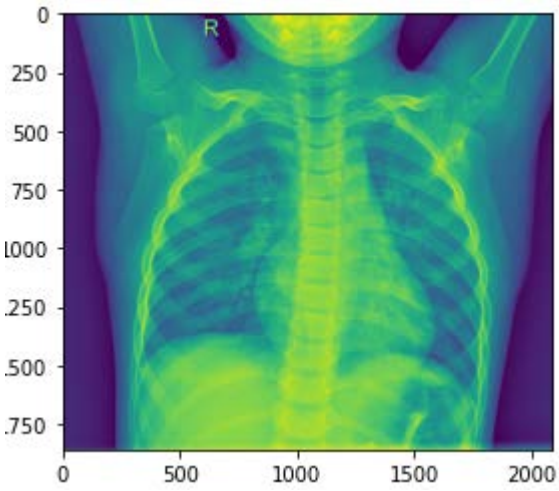
Test Data



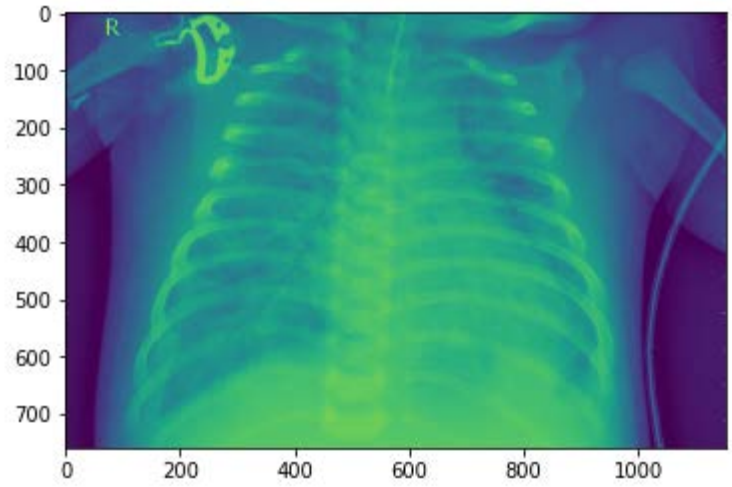
The train and test data image of normal and pneumonia X-rays are plotted here

Train data

Normal

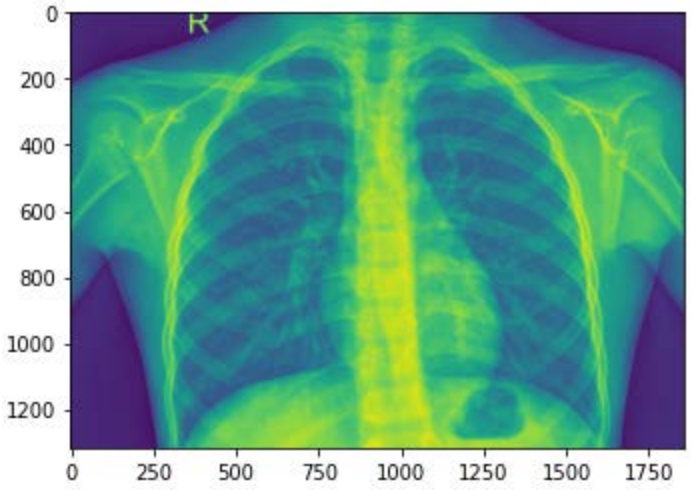


Pneumonia

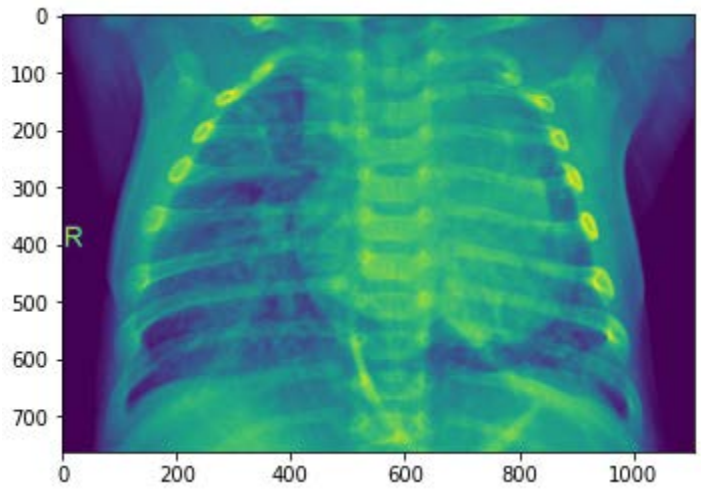


Test Data

Normal



Pneumonia



3. Data Preprocessing:

The train data has been passed through imagedata generator to produce more number of images by doing shear range , zoom range , horizontal flip . The images are also rescaled. The test data images are only rescaled . The target size of the image is (224,224)

4. Libraries Used:

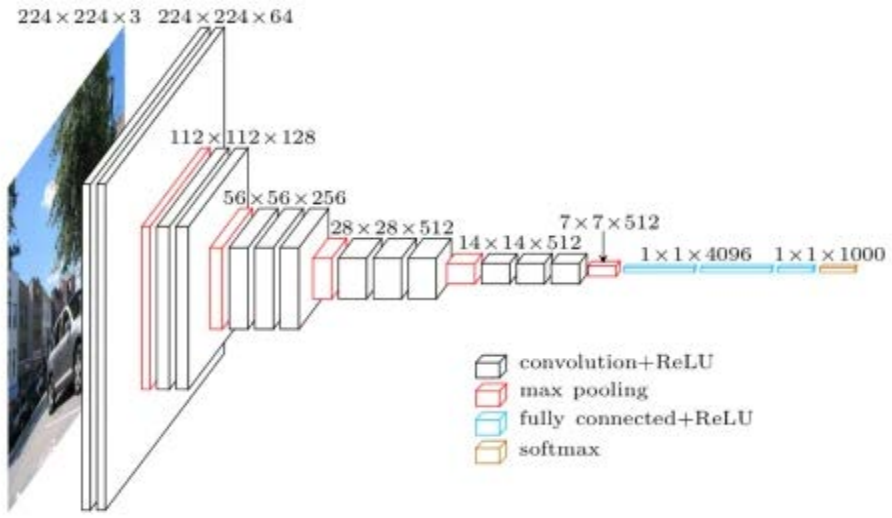
- NumPy- For working with arrays
- Pandas – For reading/writing data
- Matplotlib – to display images
- TensorFlow Keras models
- TensorFlow Keras layers

5. Model Architecture

I have tried using ResNet50, ResNet32,VGGN16 & VGGN19. Among these models VGGN19 gave the best training accuracy and testing accuracy when trained with the image size of 224. The model architecture of VGGN19 is detailed below.

The idea behind using VGGN19 is to make use of the transfer learning technique to overcome the huge computation resource used by deep convolutional neural network. Here, we will reuse the model weights from pre-trained models that were developed for standard computer vision benchmark datasets like ImageNet. The last layer with a categorical class of 1000 will be replaced with our fully connected layer that provides a two categorical class.

VGG19 in short consists of 19 layers. It consist of input with a shape of (224,224,3) , 16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer, and a classification layer with two classes.



VGG Neural Network Architecture – Source

6. Model Compilation and Training

During the model compilation, the optimizer algorithm, loss function and the list of metrics are parameters which are to be taken care of. Adam is used as the optimization algorithm, binary cross entropy is used as the loss function and accuracy is the only metric used. Model is trained for 5 epochs .The sigmoid activation is used at the output layer which provide two categories of classes.

```
model.summary()
```

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0
```

```
block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

flatten (Flatten)           (None, 25088)             0

dense (Dense)               (None, 2)                 50178

=================================================================
=================================================================
Total params: 20,074,562
Trainable params: 50,178
Non-trainable params: 20,024,384
_________________________________________________________________
```

7. Model Evaluation

The model is trained for 5 epochs and the evaluation is done based on the training accuracy, training loss , validation accuracy & validation loss. The model is able to give a training accuracy of .96 and validation accuracy of .92.
The images from the val dataset are taken and the predicted results gave good accuracy with only 4 images misclassified in normal images and all others predicted correctly.

Following are the softmax output for the two classes for the images in the Normal and Pneumonia group.

NORMAL2-IM-1427-0001

	0	1
0	0.999388	0.000611737

NORMAL2-IM-1430-0001

	0	1
0	1	0

NORMAL2-IM-1431-0001

	0	1
0	1.1831e-14	1

NORMAL2-IM-1436-0001

	0	1
0	1	2.83729e-14

NORMAL2-IM-1437-0001

	0	1
0	2.49418e-10	1

NORMAL2-IM-1438-0001

	0	1
0	2.05203e-17	1

NORMAL2-IM-1440-0001

	0	1
0	1	9.09357e-25

NORMAL2-IM-1442-0001

	0	1
0	4.51061e-21	1

person1946_bacteria_4874

	0	1
0	0	1

person1946_bacteria_4875

	0	1
0	0	1

person1947_bacteria_4876

	0	1
0	0	1

person1949_bacteria_4880

	0	1
0	2.4765e-33	1

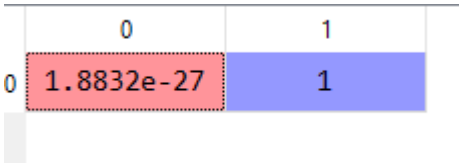
person1950_bacteria_4881

	0	1
0	0	1

person1951_bacteria_4882

person1952_bacteria_4883

person1954_bacteria_4886



Using the above result plotted a confusion matrix as below.

Confusion matrix

		Actual	
Predicted		(Positive) Pneumonia	(Negative) Normal
	(Positive) Pneumonia	8	4
	(Negative) Normal	0	4

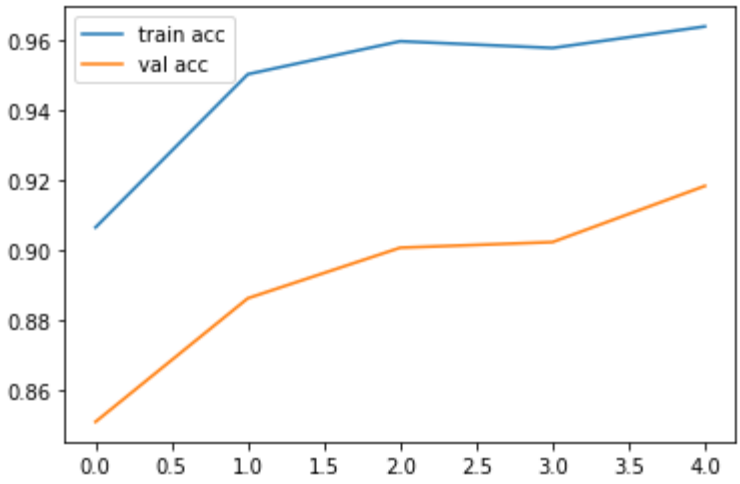
TP	FP	TN	FN	Precision	Recall
8	4	4	0	TP/TP+FP = .67	TP/TP +FN = 1

Recall and precision score plays a vital role in deciding the right model. Here we can see that the precision score is just 67% but have a 100 % recall score which is vital for this problem statement. It is necessary to have a model that gives minimum false negative as there should not be any patient with pneumonia predicted as normal. Our model is able to achieve the task at hand by giving a good recall score.

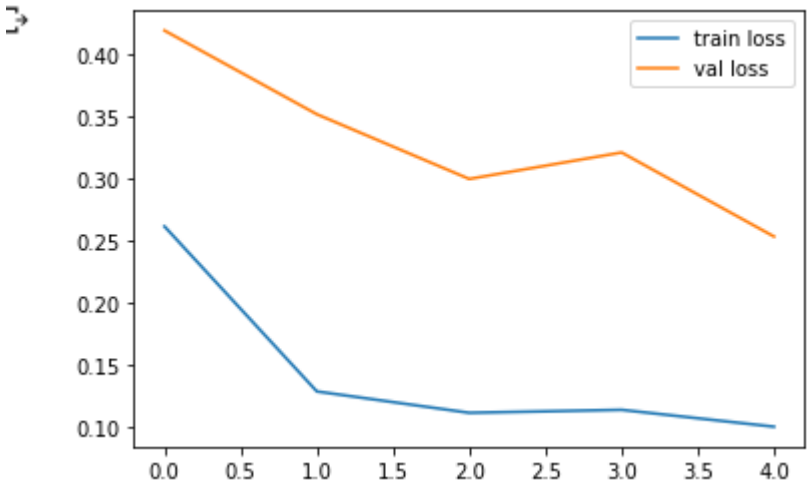
Training result per epoch

```
Epoch 1/5
163/163 [==============================] - 1258s 7s/step - loss: 0.2617 - accuracy: 0.9064 - val_loss: 0.4196 - val_accuracy: 0.8510
Epoch 2/5
163/163 [==============================] - 151s 928ms/step - loss: 0.1287 - accuracy: 0.9502 - val_loss: 0.3521 - val_accuracy: 0.8862
Epoch 3/5
163/163 [==============================] - 151s 925ms/step - loss: 0.1115 - accuracy: 0.9595 - val_loss: 0.3000 - val_accuracy: 0.9006
Epoch 4/5
163/163 [==============================] - 147s 903ms/step - loss: 0.1139 - accuracy: 0.9576 - val_loss: 0.3213 - val_accuracy: 0.9022
Epoch 5/5
163/163 [==============================] - 147s 902ms/step - loss: 0.1003 - accuracy: 0.9638 - val_loss: 0.2536 - val_accuracy: 0.9183
```

Training Accuracy VS Validation Accuracy



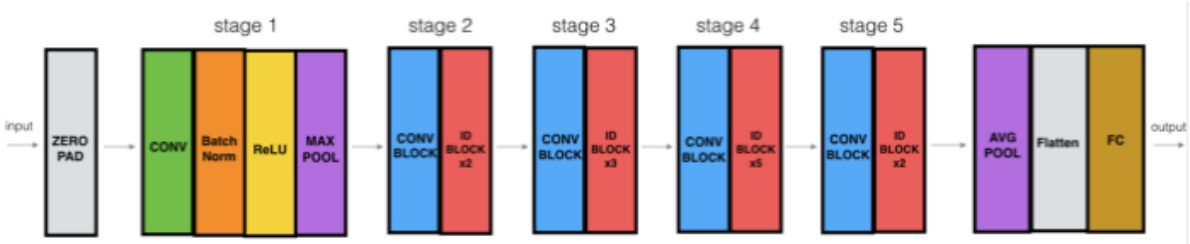
Training loss VS Validation loss



8. Comparison with other models

Following models are trained with the chest X-Ray image dataset.

1) ResNet50 - ResNet-50 is a convolutional neural network that is 50 layers deep. The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.
   Convolutional Neural Networks have a major disadvantage - Vanishing Gradient Problem. During backpropagation, the value of gradient decreases significantly, thus hardly any change comes to weights. To overcome this, ResNet is used. It makes use of skip connection**.**



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

2) ResNet32 - ResNet-32 is a convolution neural network backbone that is based off alternative ResNet networks such as ResNet-34, ResNet-50, and ResNet-101. As its name implies, ResNet-32 has 32 layers. It addresses the problem of vanishing gradient with the identity shortcut connection that skips one or more layers.

3) VGG16 - To train the model I have used VGG neural network that consist of 16 layers.VGG16 has 13 convolutional layers and 3 fully connected layers. The pretrained weigths of the convolutional layers are used . The input shape provided in (224,224,3). Removed the fully connected layer of the network in order to have a fully connected layer with output layer having softmax activation and 2 nodes. Basically removed the default 1000 classification of the VGG network with two classifications.

4) VGG19 - VGG19 is a variant of VGG model which in short consists of 19 layers. The input shape of (224,224,3) , 16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer, and a classification layer with two classes.

Given below a comparison table with respect to training accuracy and validation accuracy

Model	Training Accuracy	Validation Accuracy
ResNet50	0.95	0.63
ResNet32	0.93	0.57
VGGN19	0.96	0.92
VGGN16	0.97	0.90

9. Conclusion

Tried various architectures to train the model. I have used already trained networks ResNet50,ResNet32,VGGN16,VGGN19 for training with the chest X-Ray dataset. The weights of the pre-trained model are used and the output layer is replaced with our fully connected layer having a softmax activation and two classification nodes. Comparing the training accuracy and validation accuracy VGGN19 model with image size 224 gave the best result. Predicted the classification of images in the val dataset and it is found that only 4 images were misclassified in the normal category with rest all predicted correctly. Plotted a confusion matrix and calculated the precision and recall score. The model is able to give a 100% recall score with the val dataset. A recall score of 100% shows that all patients with pneumonia are predicted to have pneumonia which is the most important criteria for this model.

10. Reference

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