

Udacity Machine Learning Nanodegree 2021

Capstone Project Report

Pneumonia Detection with Deep Learning

Ravee Rachakonda

Oct 2021

Problem Statement

The main goal of this capstone project is to utilize Deep Learning algorithms for detecting pneumonia in provided X-Ray. Build classifier which can score the X-Ray for detecting pneumonia and output as detected yes or no.

Datasets and Inputs

The dataset used for this capstone project is from the Kaggle project. The data is a set of chest X-Ray images in jpeg format. Dataset consists of 5956 x-ray images and is categorized into 3 categories:

- Train (5216)
 - NORMAL (1341)
 - PNEUMONIA (3875)
- Test (624)
 - NORMAL (234)
 - PNEUMONIA (390)
- Val (16)
 - NORMAL (8)
 - PNEUMONIA (8)

Solution Statement

The development of convolutional neural network layers has allowed for significant gains in the ability to classify images and detect objects in a picture. These are multiple processing layers to which image analysis filters, or convolutions, are applied. The abstracted representation of images within each layer is constructed by systematically convolving multiple filters across the image, producing a feature map that is used as input to the following layer. This architecture makes it possible to process images in the form of pixels as input and to give the desired classification as output. The image-to-classification approach in one classifier replaces the multiple steps of previous image analysis methods.

Benchmark Model

The benchmark is used from a research paper published in cell.com ([https://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5)). In the comparison of chest X-rays presenting as pneumonia versus normal, the research team

achieved an accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1% operating on the same data set. Research team used a transfer learning technique with CNN that was trained on various kinds of image data (Retinal, Chest X-Rays).

Evaluation Metrics

Loss, Accuracy, Precision and Recall are really good metrics for such kinds of problems as the dataset is 81.921% of Pneumonia positive. We will use the confusion matrix from the predictions and check what is the recall and precision of my model.

Project Design

Data Pre-processing

Data provided in the Kaggle project is split into train, test and validation sets. Each set contains folders for Normal and Pneumonia categories.

1. Image processing steps should be built to read in images
2. Leverage python libraries for convert image data to matrices

Kaggle input data containing the X-Ray images is pre-organized into train, val, test and classes: NORMAL, PNEUMONIA.

I have downloaded the kaggle data set, unzipped and uploaded the data to S3 bucket under data directory. As the data processing will be done with large instances using Sagemaker, it will be handy to have the data in S3 bucket. Images are grayscale and standard size of 224x224.

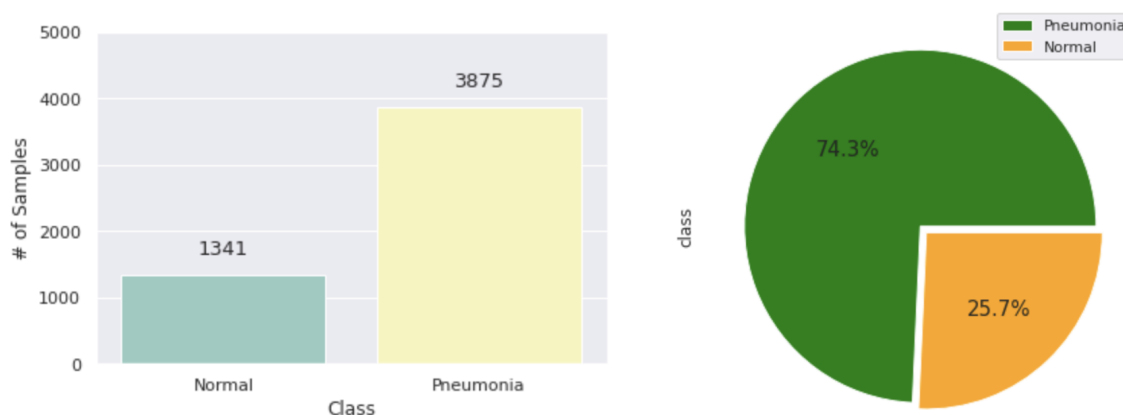


Fig 1: Training data distribution with $\sim 3\times$ of Pneumonia images over Normal

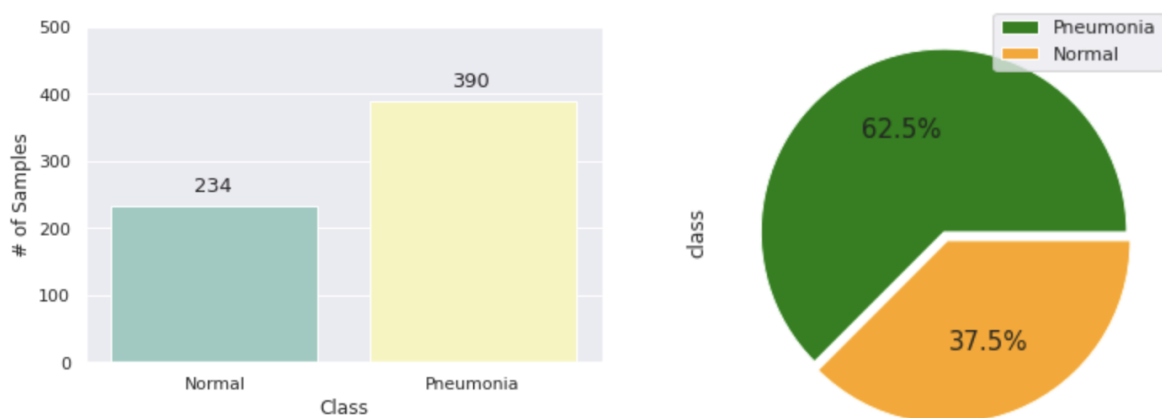
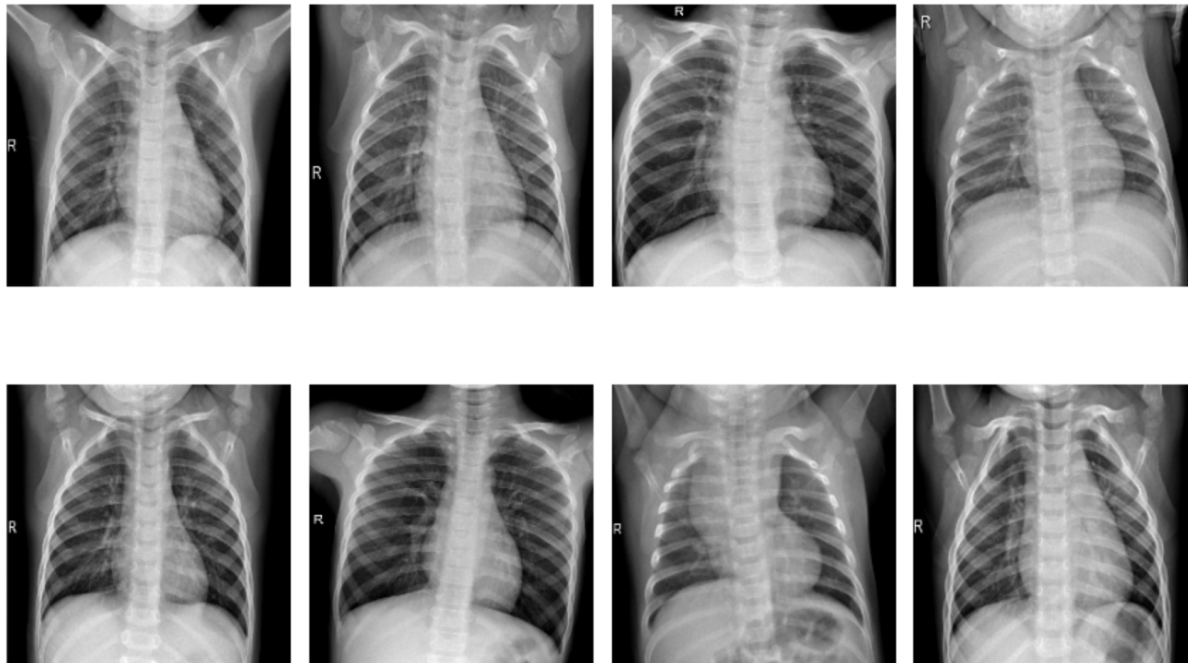


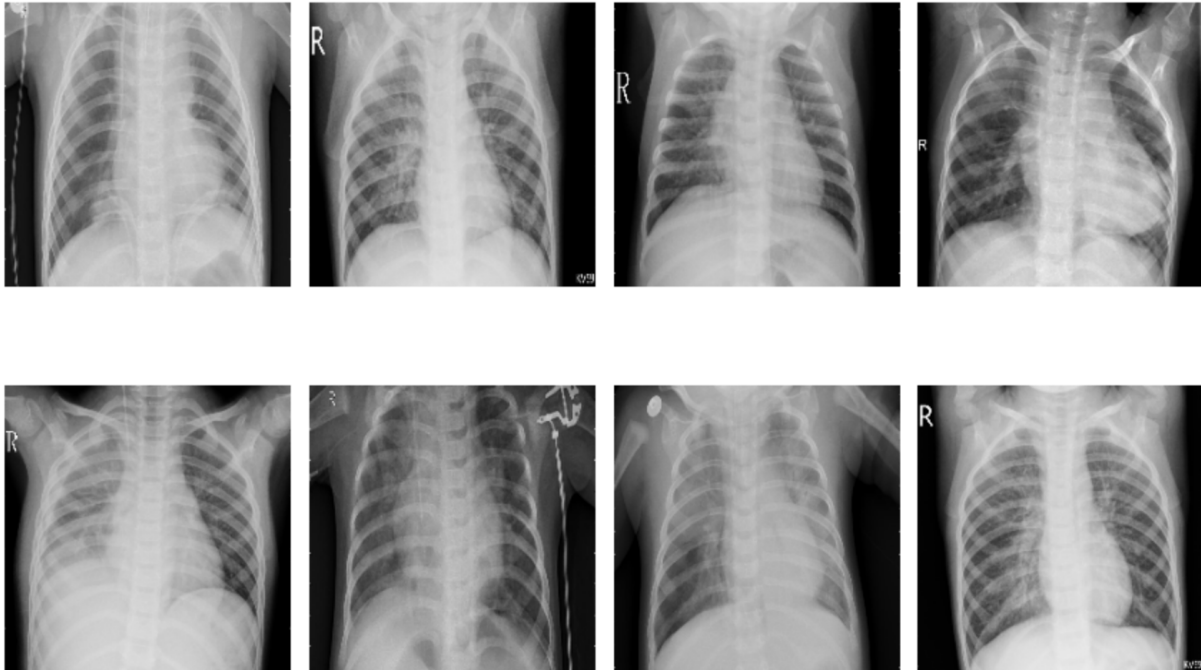
Fig 2: Training data distribution with $\sim 1.7\times$ of Pneumonia images over Normal

Fig 3: Visualizing the a sample of 12 x-ray images for Normal and Pneumonia

Train Set - Normal



Train Set - Pneumonia



Data Splitting

Data is already split for train, test and validation sets. We will need to randomly load the corresponding matrix data for normal and pneumonia images and create a training data set. Similarly, for test and Validation sets.

DataFrame object is created by loading all the normal and pneumonia images for training into df_train. Total of 5216 rows with class [Normal, Pneumonia] and Image path.

Training (df_train): 5126 x 2 columns

Test (df_test): 624 x 2 columns

	class	image
0	Normal	./chest-xray-pneumonia/chest_xray/train/NORMAL/NORMAL2-IM-0571-0001.jpeg
1	Normal	./chest-xray-pneumonia/chest_xray/train/NORMAL/NORMAL2-IM-0799-0001.jpeg
2	Normal	./chest-xray-pneumonia/chest_xray/train/NORMAL/NORMAL2-IM-0995-0001-0002.jpeg
3	Normal	./chest-xray-pneumonia/chest_xray/train/NORMAL/IM-0704-0001.jpeg
4	Normal	./chest-xray-pneumonia/chest_xray/train/NORMAL/NORMAL2-IM-0986-0001.jpeg
...
5211	Pneumonia	./chest-xray-pneumonia/chest_xray/train/PNEUMONIA/person967_bacteria_2892.jpeg
5212	Pneumonia	./chest-xray-pneumonia/chest_xray/train/PNEUMONIA/person1340_virus_2312.jpeg
5213	Pneumonia	./chest-xray-pneumonia/chest_xray/train/PNEUMONIA/person1679_bacteria_4450.jpeg
5214	Pneumonia	./chest-xray-pneumonia/chest_xray/train/PNEUMONIA/person585_bacteria_2414.jpeg
5215	Pneumonia	./chest-xray-pneumonia/chest_xray/train/PNEUMONIA/person1343_bacteria_3414.jpeg

5216 rows x 2 columns

Using the train_test_split, I have split the training data into 80% and validation 20%. Keras ImageDataGenerator is used for processing the image data and creating the data set. Had issues with directly reading the files from S3 using flow_from_directory so leveraged the flow_from_dataframe by reading the images from the local directory on the sagemaker server.

Model Training

We will use Deep Learning algorithms using Convoluted Neural Network (CNN) to train and build a model. I have used Tensorflow with Sagemaker for building the model. Kernel conda_tensorflow2_36 (Tensorflow 2 with Python 3.6) is used for the execution. Along with the Sequential model, I have tried transfer learning models (ResNet152V2, DenseNet121) to compare performance.

A Sequential Model with 6 layers total. Image size of 224 x 224 with a depth of 3 is passed to the inputs_input layer.

Notebook initialized the Tensorflow function with calling entry_point of train.py which calls model.py for building the model.

0	/ Pneumonia-Detection-DL-Project	Name	Last Modified	File size
	..		seconds ago	
	chest-xray-pneumonia		8 months ago	
	train_model		6 minutes ago	
	image-classification-fulltraining.ipynb		8 days ago	27.6 kB
	Pneumonia-Detection-v2.ipynb	Running	2 minutes ago	3.43 MB

Evaluation:

Testing the machine learning models with 100 epochs:

1. Sequential
2. Resnet152v2 (pretrained model)
3. DenseNet121 (pretrained model)

Performance Metrics

Model: Sequential

Metrics:

Test loss	0.6046411311
Test accuracy	0.8445512652

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.39	0.24	0.3	234
Normal (Class 1)	0.63	0.77	0.69	390
accuracy			0.57	624
macro avg	0.51	0.51	0.5	624
weighted avg	0.57	0.54	0.57	624

ROC_AUC	0.4945923734
---------	---------------------

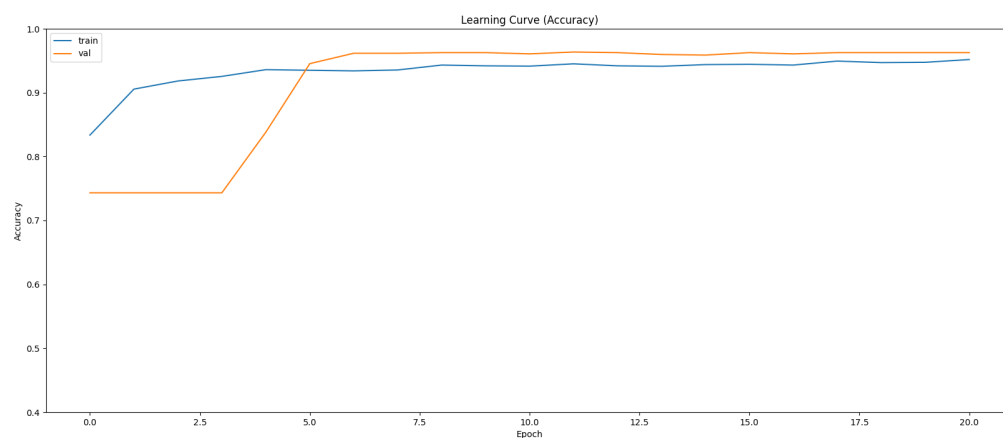


Fig 4: Learning Curve (Accuracy) - Sequential Model

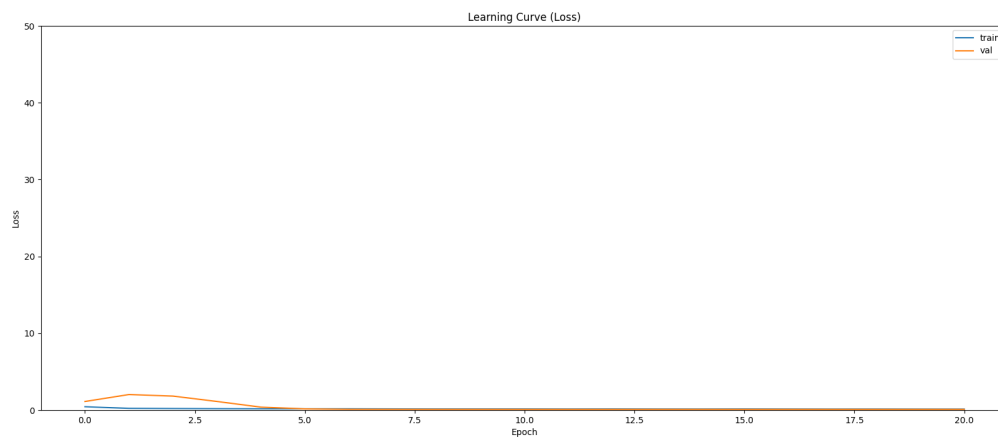


Fig 5: Learning Curve (Loss) - Sequential Model

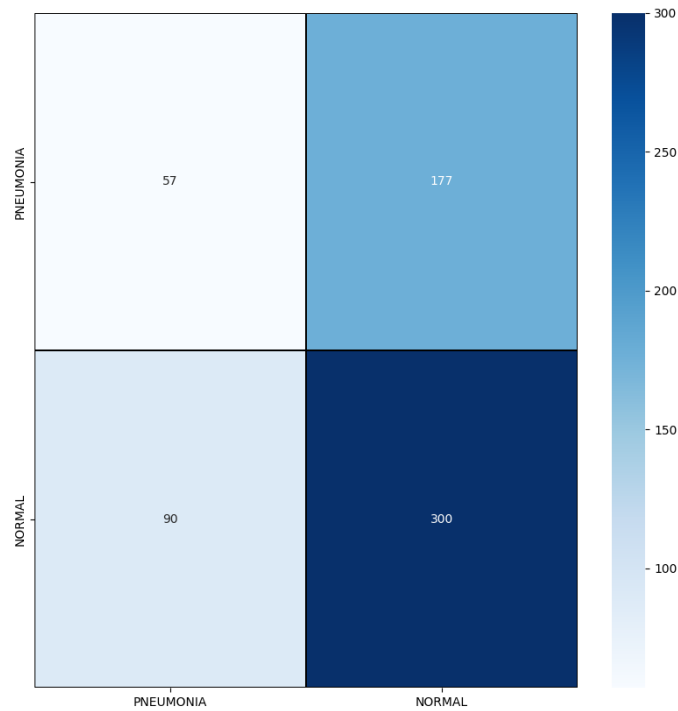


Fig 6: Confusion Matrix - Sequential Model

Sequential model, while triggered for 100 epochs, stopped at 20 epochs as the learning rate change has not been changing for a few epochs.

100 Epoch - (Resnet152v2) Transfer Learning

Resnet152v2 pretrained model is used as part of transfer learning with layers locked and run for 100 epochs. The execution was very fast and the algorithm performed better than the sequential model.

Metrics

Test loss	0.3989337858
Test accuracy	0.8653846383

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.34	0.24	0.28	234

Normal (Class 1)	0.61	0.72	0.66	390
accuracy			0.54	624
macro avg	0.48	0.48	0.47	624
weighted avg	0.51	0.54	0.52	624

ROC_AUC	0.4984056542
---------	--------------

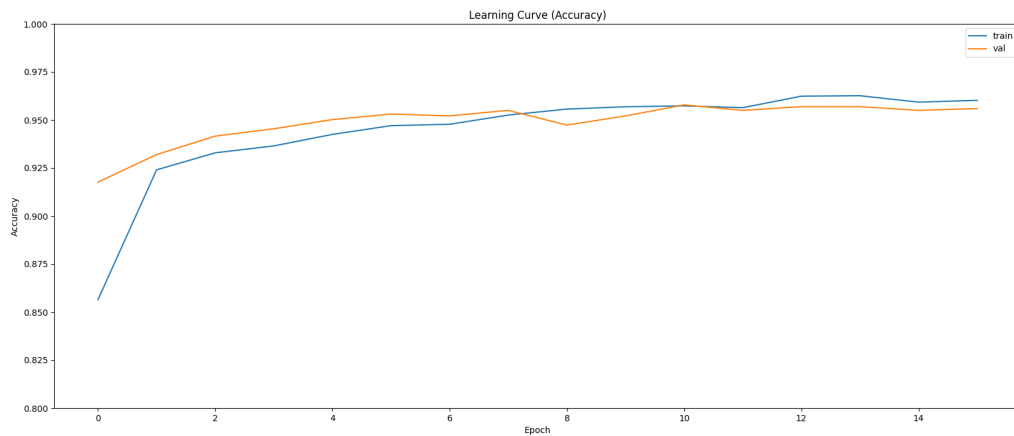


Fig 7: Learning Curve (Accuracy) - Resnet152v2 Model

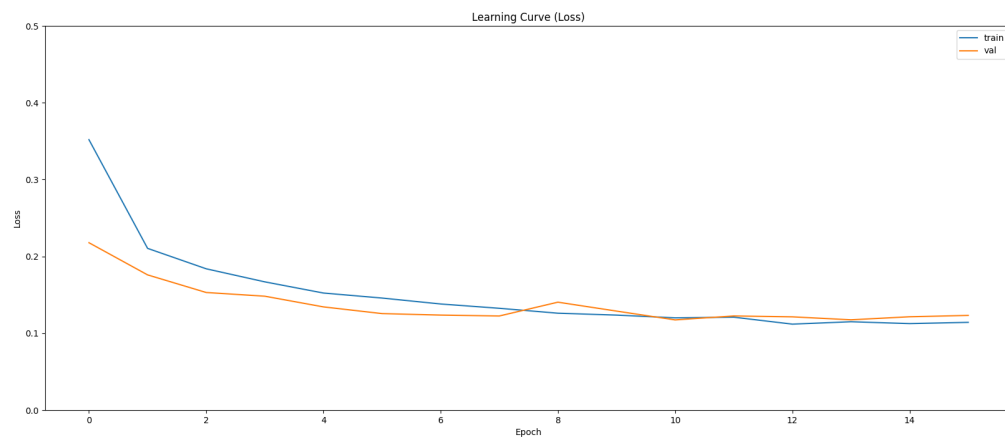


Fig 8: Learning Curve (Loss) - Resnet152v2 Model

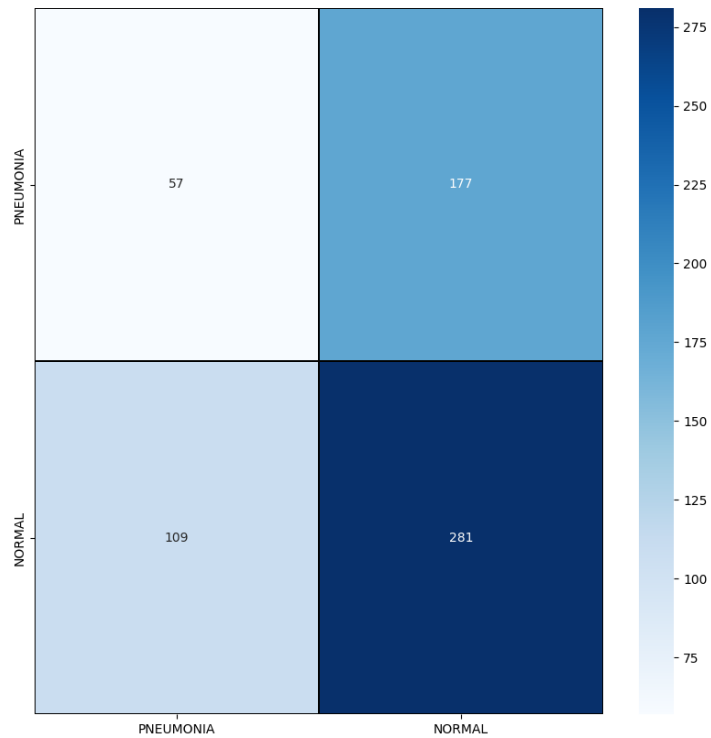


Fig 9: Confusion Matrix - Resnet152v2 Model

100 Epoch - DenseNet121

Densenet121 pretrained model is used as part of transfer learning with layers locked and run for 100 epochs. The execution was very fast and the algorithm performed better than the sequential model but a bit lesser than Resnet152v2.

Metrics

Test loss	0.3881112001
Test accuracy	0.8541666865

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.32	0.22	0.26	234
Normal (Class 1)	0.61	0.72	0.66	390

accuracy			0.53	624
macro avg	0.46	0.47	0.46	624
weighted avg	0.5	0.53	0.51	624

ROC_AUC	0.4945923734
---------	--------------

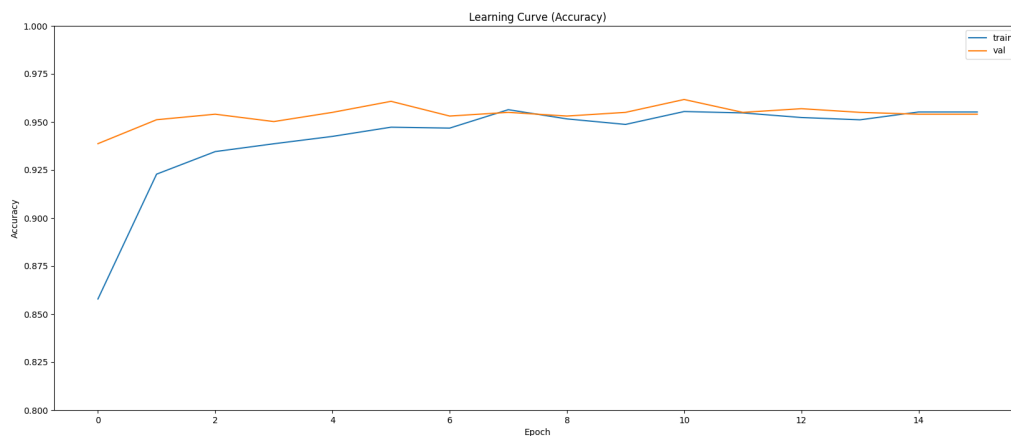


Fig 10: Learning Curve (Accuracy) - Densenet121 Model

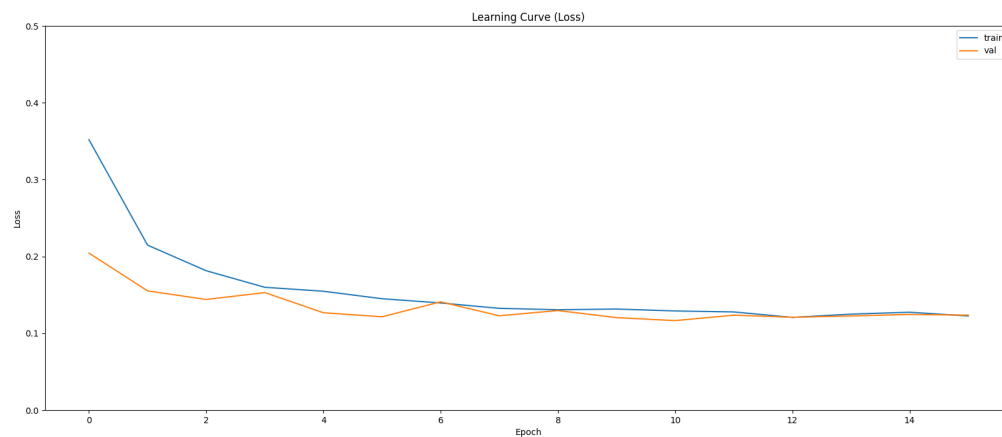


Fig 11: Learning Curve (Loss) - Densenet121 Model

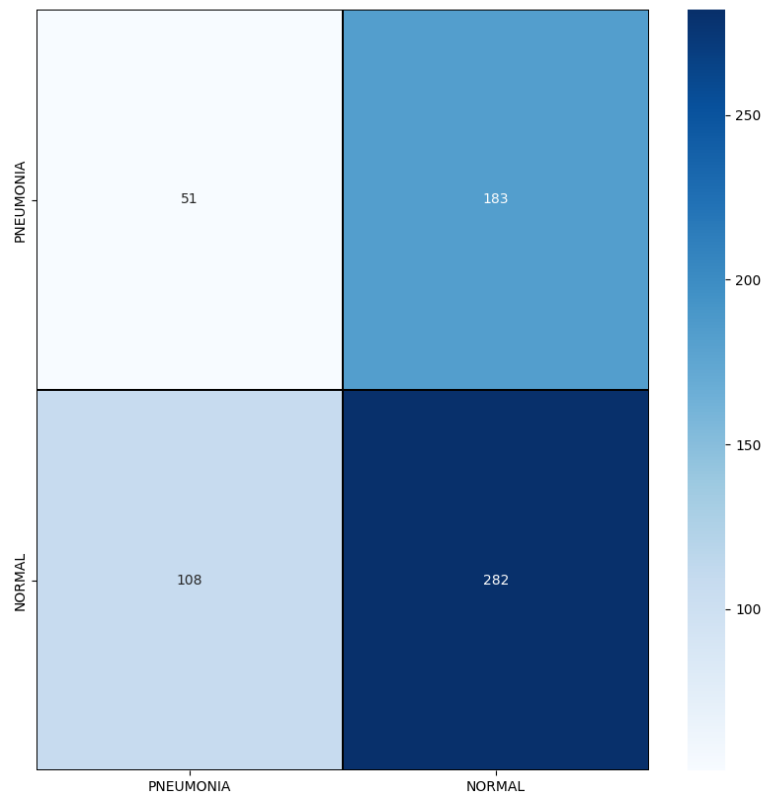


Fig 12: Confusion Matrix - Densenet121 Model

Comparison of test accuracy between the 3 models

	Sequential	Resnet152v2	Densenet121
Test loss	0.6046411311	0.3989337858	0.3881112001
Test accuracy	0.8445512652	0.8653846383	0.8541666865

Resnet152v2 has performed better than the Densenet121 and Sequential models.

300 Epoch Transfer Learning - Resnet152v2

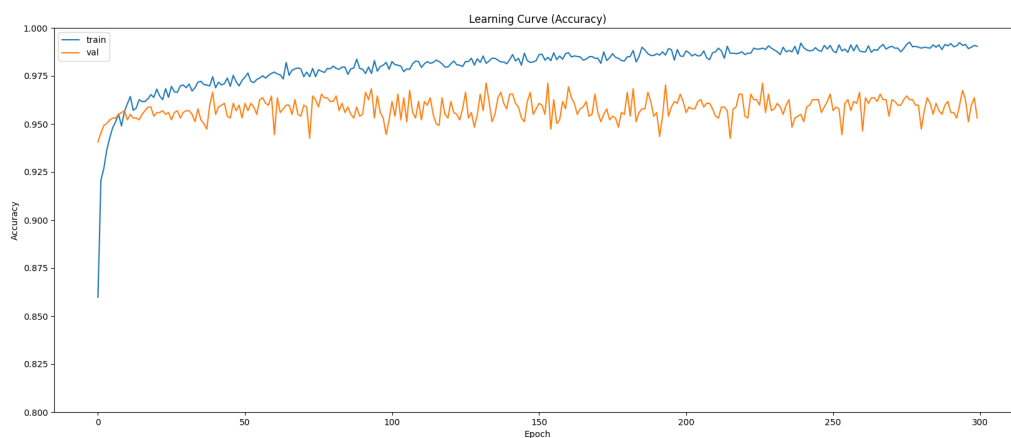
Tested with Resnet152v2 for 300 epochs the model performance has improved tremendously with accuracy of 92%.

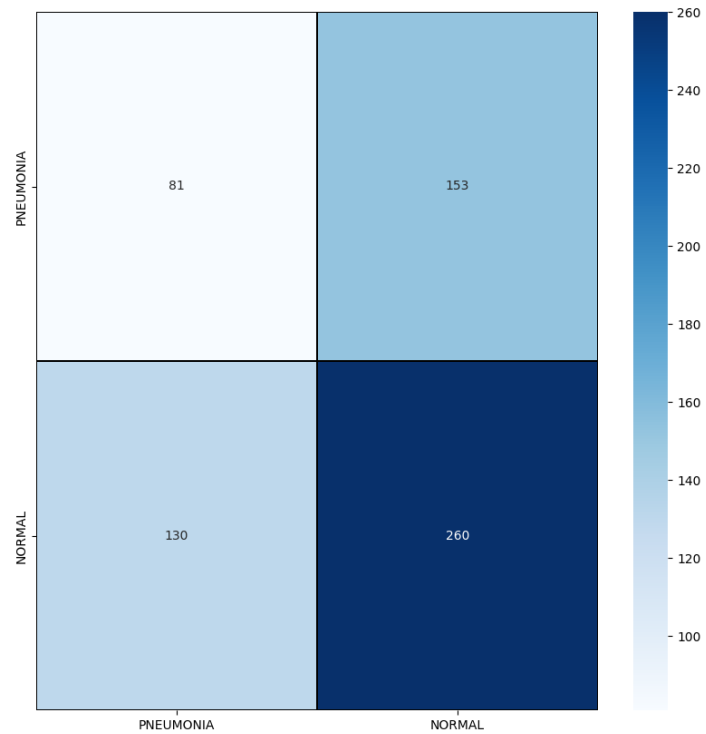
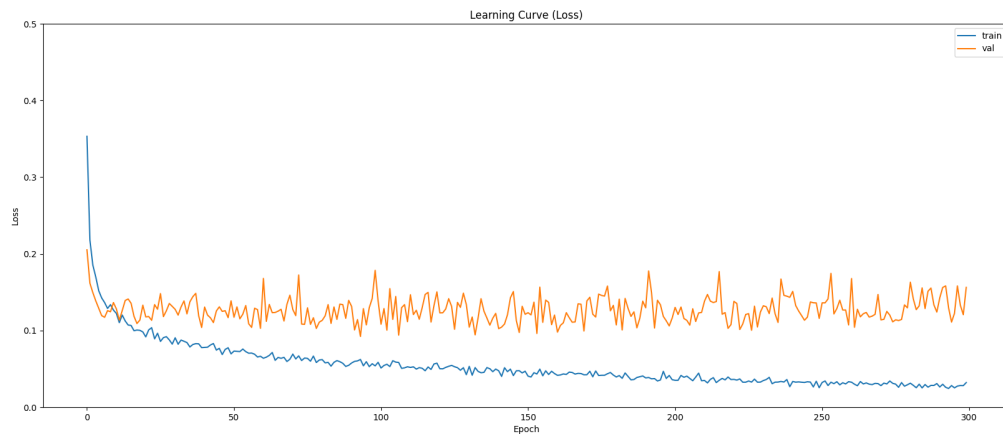
Metrics

Test loss	0.2732946018
Test accuracy	0.9214743376

ROC_AUC	0.5423460443
---------	--------------

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.38	0.35	0.36	234
Normal (Class 1)	0.63	0.67	0.65	390
accuracy			0.55	624
macro avg	0.51	0.51	0.51	624
weighted avg	0.54	0.55	0.54	624





$$\text{Sensitivity} = \frac{81}{81 + 130} = 0.3838$$

$$\text{Specificity} = \frac{260}{260 + 153} = 0.6295$$

The model has a high accuracy rate when tested on test images. Sensitivity and Specificity are weak.

References

- [1] Daniel S. Kermany, Michael Goldbaum, Wenjia Cai, Carolina C.S. Valentim, Huiying Liang, Sally L. Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, Justin Dong, Made K. Prasadha, Jacqueline Pei, Magdalene Y.L. Ting, Jie Zhu, Christina Li, Sierra Hewett, Jason Dong, Ian Ziyar, Alexander Shi, Runze Zhang, Lianghong Zheng, Rui Hou, William Shi, Xin Fu, Yaou Duan, Viet A.N. Huu, Cindy Wen, Edward D. Zhang, Charlotte L. Zhang, Oulan Li, Xiaobo Wang, Michael A. Singer, Xiaodong Sun, Jie Xu, Ali Tafreshi, M. Anthony Lewis, Huimin Xia, Kang Zhang
"Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning" [https://www.cell.com/cell/fulltext/S0092-8674\(18\)30154-5](https://www.cell.com/cell/fulltext/S0092-8674(18)30154-5)
- [2] Paul Mooney <https://www.kaggle.com/paultimothymooney/chest-xraypneumonia>
- [3] <https://en.wikipedia.org/wiki/Pneumonia>
- [4] <https://www.hopkinsmedicine.org/health/conditions-and-diseases/pneumonia>
- [5] <https://www.kaggle.com/jonaspalucibarbosa/chest-x-ray-pneumonia-cnn-transfer-learning/notebook>
- [6] <https://www.kaggle.com/saran27/pneumonia-detection-using-cnn-91-3-accuracy>
- [7] https://medium.com/@robparkin_38642/running-keras-models-on-amazon-sagemaker-b4059843f767
- [8] <https://towardsdatascience.com/using-tensorboard-in-an-amazon-sagemaker-pytorch-training-job-a-step-by-step-tutorial-19b2b9eb4d1c>
- [9] <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7345724/>
- [10] <https://en.wikipedia.org/wiki/Pneumonia>