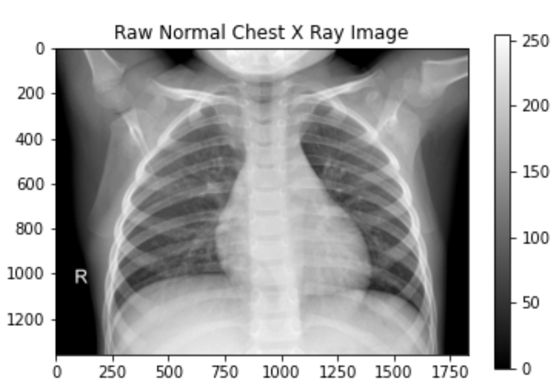
Project Report: Machine Learning Engineer Nanodegree Capstone Project

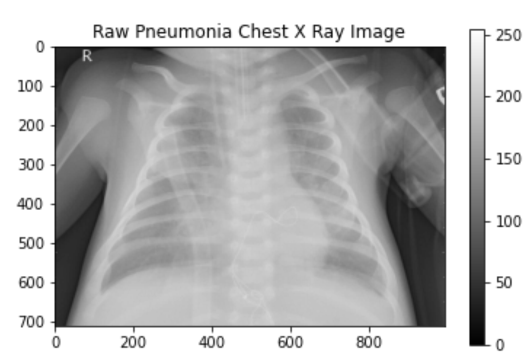
Submitted by: Anup Pandey
Date: 23 April, 2021

I. **Define:** In this project, chest X-ray images from the public have been used for the detection of pneumonia using convolutional neural network (CNN) and transfer learning frameworks of the machine-learning (ML) domain. Pneumonia is an inflammation of small air sacs in the lungs known as alveoli. Chest X-ray is one of the tools to help confirm the diagnosis. As a potential solution, we have applied CNN and transfer learning for extracting valuable features about pneumonia from medical images and instantly detect it in human chest X-rays. A convolutional neural network (CNN) will be the benchmark model for the problem. The performance of the transfer learning model based on previously trained models has been compared and evaluated against the base CNN model. The 'accuracy' and F1 score are taken as the evaluation metrics for the model's performance.

II. **Analyze:** The datasets have been taken from Kaggle's Chest X-Ray Images (https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia). The dataset comprises chest X-ray images each with pixel size 736 by 1048 and 3 color channels. The images are broadly classified into two main labels: 'PNEUMONIA' and 'NORMAL'. Each label has images separated into 'train', 'test', and 'val' directories for training, testing, and validation purpose.

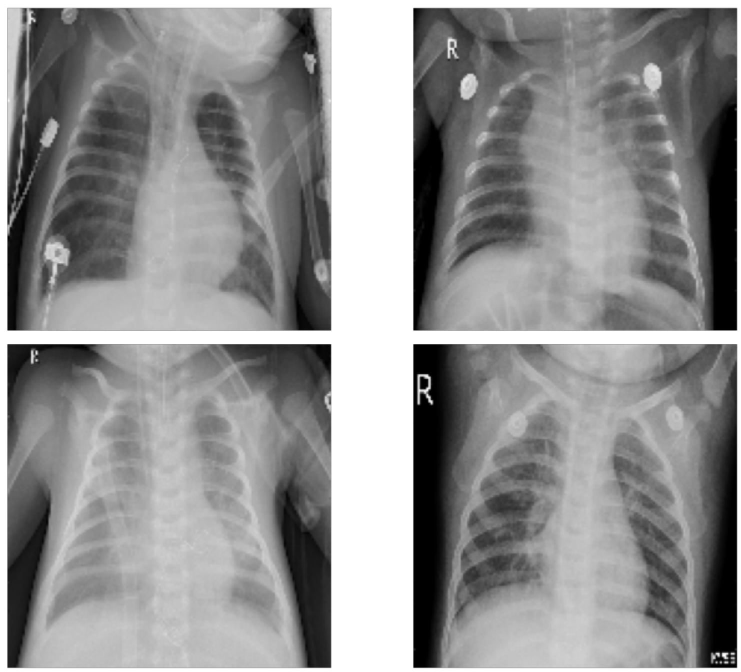
1. **Exploring Images:** Each X-ray image has varying pixel size and 3 color channels. The grayscale images for the normal and diseased (pneumonia) are shown in the figure below:



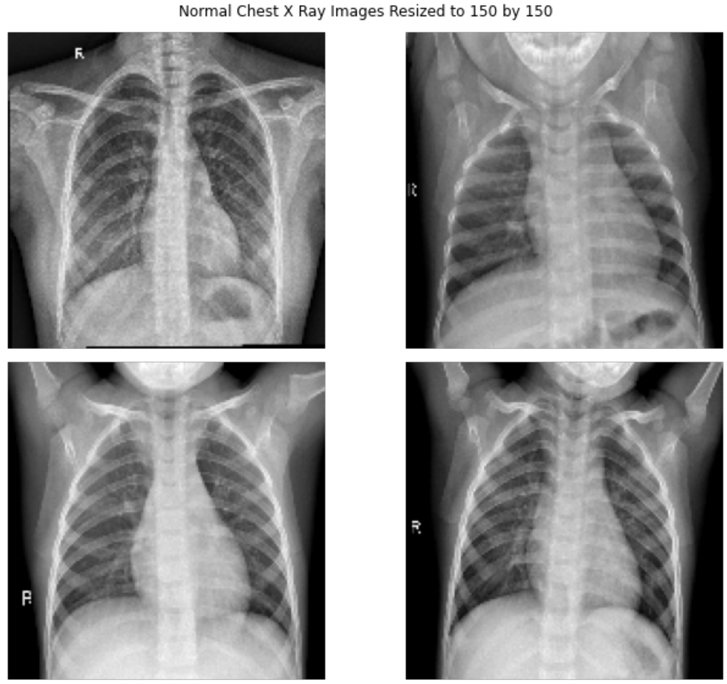


The pixel size of normal chest X-ray shown above is 1357 by 1828 and that of Pneumonia chest X-ray is 712 by 992. Therefore, all the images are prepressed to the same pixel size of 150 by 150. The resized images for normal and the one with pneumonia chest X-rays are shown in the figure below:

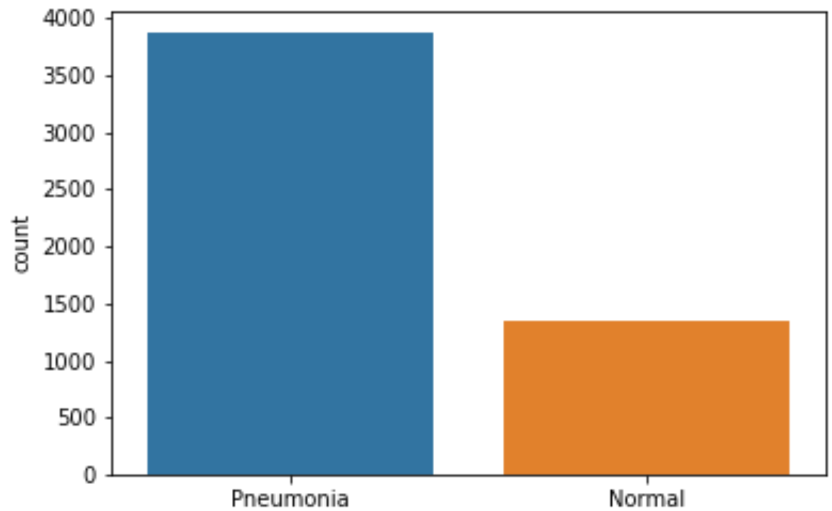
Pneumonia Chest X Ray Images Resized to 150 by 150



Normal Chest X Ray Images Resized to 150 by 150



2. **Check class balance/imbalance:** The data labels will be plotted to check the class balance/imbalance.



The above plot shows the training data is imbalanced. In such a case, using normal loss function will result in a model that biases towards the dominating class, which is Pneumonia in our case. One way to address this issue is to use a weighted loss function that imbalances the contribution of each class in the loss function. The weights for each label are calculated in the following ways:

weight_0 = number of Pneumonia in train set/ Total train sets
weight_1 = number of Normal in train set/ Total train sets

3. **Data Preprocessing and Data Augmentation:** One of the ways to expand the training data set is by data augmentation. The training data set can be artificially expanded with small transformations in the images to reproduce the variations. For the data augmentation and preprocessing, the Keras ImageDataGenerator function has been used in this project. The images are augmented in the following four ways:
    i. Rescale the image
    ii. Randomly rotate the images by 20 degrees.
    iii. Randomly zoomed the images by 10%
    iv. Randomly shifted the width by 10%

From the above exploratory data analysis, we can conclude that CNN models are appropriate for solving the problem.

III. **Implement:** The training process has been implemented in the following two steps:

1. **CNN base model:** The base CNN model based on Keras sequential architecture has been defined to solve a binary classification problem. The hyperparameters are optimized to get the best accuracy from the model.
2. **Transfer Learning:** The top-performing pre-trained models for image recognition known as VGG19 and ResNet152 has been used to build a transfer learning model. The weights of the pre-trained models have been optimized which are customized with few outer layers.

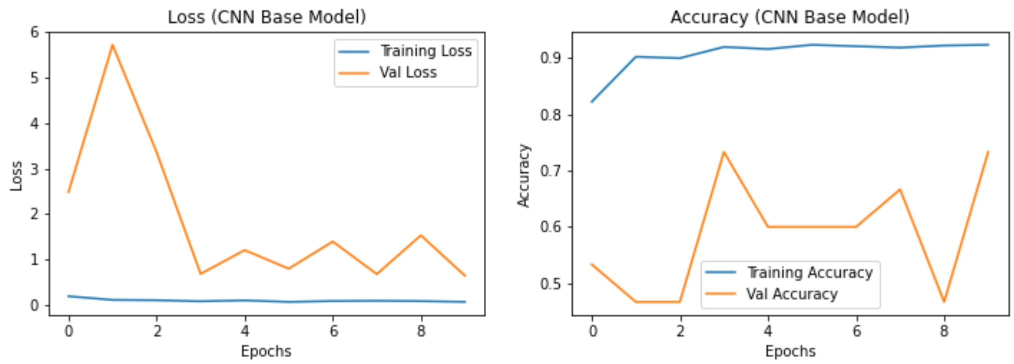
The model's performance has been evaluated by the model's predictions against the test images that are not used during the training. The model's accuracy and F1 score have been used as the evaluation metric.

Model architecture of CNN base model.

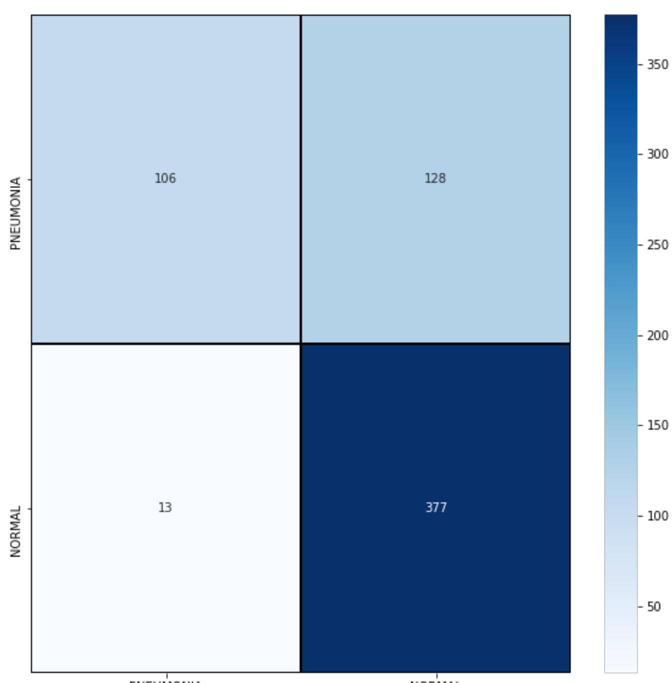
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
batch_normalization (BatchNo (None, 148, 148, 32)      128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 146, 146, 32)      9248
_________________________________________________________________
batch_normalization_1 (Batch (None, 146, 146, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 73, 73, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 71, 71, 64)        18496
_________________________________________________________________
batch_normalization_2 (Batch (None, 71, 71, 64)        256
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 69, 69, 64)        36928
_________________________________________________________________
batch_normalization_3 (Batch (None, 69, 69, 64)        256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 34, 34, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 32, 32, 128)       73856
_________________________________________________________________
batch_normalization_4 (Batch (None, 32, 32, 128)       512
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 30, 30, 128)       147584
_________________________________________________________________
batch_normalization_5 (Batch (None, 30, 30, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 15, 15, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 28800)             0
_________________________________________________________________
dense (Dense)                (None, 128)               3686528
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 3,975,457
Trainable params: 3,974,561
Non-trainable params: 896
```

IV. Results:

1. **Benchmark Model (CNN base model):** The CNN model is trained with the binary crossentropy as a loss, a learning rate of 0.0001 for Adam optimizer, and 10 epochs. The training accuracy is 95.07% while the validation accuracy is 64.42 %. The accuracy on test set images is 77.40%.



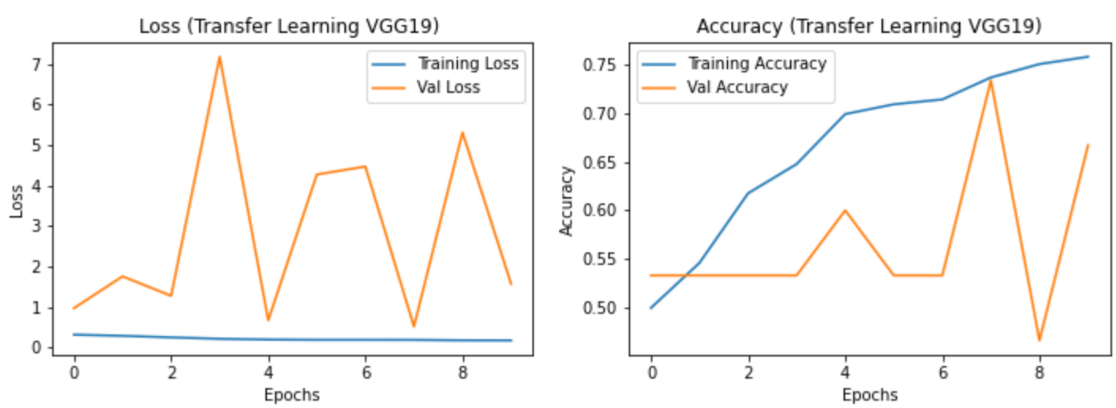
As a second metric to evaluate the model's performance, confusion metric and classification report are calculated. The confusion metric shows that for the label '0' (Pneumonia), 106 are true positives and 128 are false positives. For the label '1' (Normal), 13 is false positive while 377 is true positive.



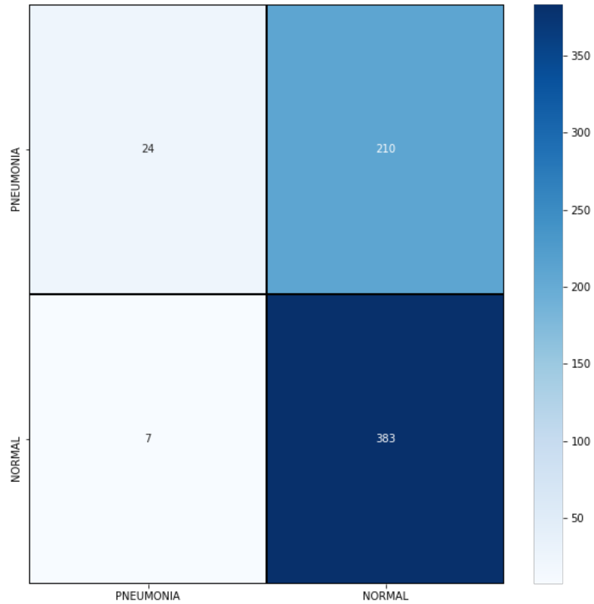
The F1-score for class 0 (Pneumonia) is 0.60 and for class 1 (Normal) is 0.84. This shows that the CNN model's prediction is better for Normal images compared to the diseased ones.

	Pneumonia (Class 0)	Normal (Class 1)	accuracy	macro avg	weighted avg
precision	0.890756	0.746535	0.774038	0.818645	0.800618
recall	0.452991	0.966667	0.774038	0.709829	0.774038
f1-score	0.600567	0.842458	0.774038	0.721512	0.751749
support	234.000000	390.000000	0.774038	624.000000	624.000000

2. **Transfer Learning using VGG19 pre-trained model:** The top-performing pre-trained model for image recognition known as VGG19 has been used to build a transfer learning model. The weights of the pre-trained model will be used which will be customized with few outer layers. The model has been trained with the binary crossentropy as a loss, a learning rate of 0.0001 for Adam optimizer, and 10 epochs. At the end of the training, the training accuracy is 84.22% while the validation accuracy is 66.67%. The accuracy on test set images is 65.22%.



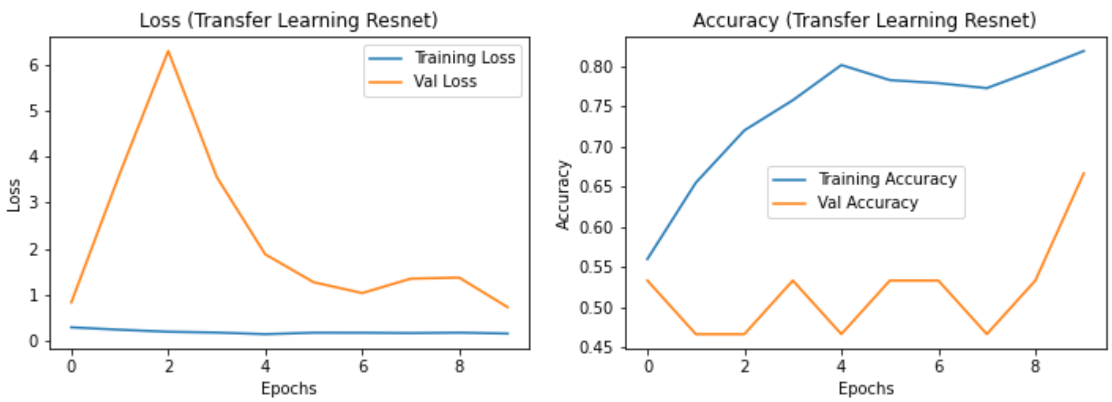
As a second metric to evaluate the model's performance, the confusion matrix and classification report are calculated. The confusion matrix shows that for the class '0' (Pneumonia), 24 are true positives and 210 are false positives. For the class '1' (Normal), 7 is false positive while 383 is true positive.



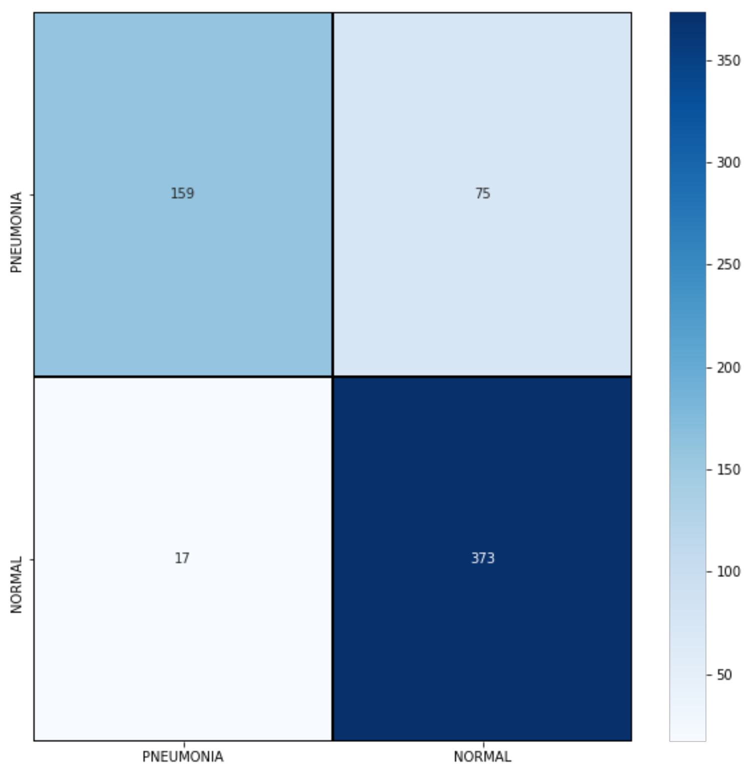
The F1-score for class 0 (Pneumonia) is 0.18 and for class 1 (Normal) is 0.78. This shows that the CNN model's prediction is better for Normal images compared to the diseased ones.

	Pneumonia (Class 0)	Normal (Class 1)	accuracy	macro avg	weighted avg
precision	0.774194	0.645868	0.652244	0.710031	0.693990
recall	0.102564	0.982051	0.652244	0.542308	0.652244
f1-score	0.181132	0.779247	0.652244	0.480190	0.554954
support	234.000000	390.000000	0.652244	624.000000	624.000000

3. **Transfer Learning using ResNet152 pre-trained model:** Another top-performing pre-trained model for image recognition known as ResNet152 has been used to build a transfer learning model. The weights of the pre-trained model will be used which will be customized with few outer layers. The model has been trained with the binary crossentropy as a loss, a learning rate of 0.0001 for Adam optimizer, and 10 epochs. At the end of the training, the training accuracy is 92.14% while the validation accuracy is 66.67%. The accuracy of test set images is 85.26%.



As a second metric to evaluate the transfer learning model's performance, the confusion matrix and classification report are calculated. The confusion matrix shows that for the class '0' (Pneumonia), 159 are true positives and 75 are false positives. For the class '1' (Normal), 17 is false positive while 373 is true positive.



The F1-score for class 0 (Pneumonia) is 0.78 and for class 1 (Normal) is 0.89. This shows that the CNN model's prediction is better for Normal images compared to the diseased ones.

	Pneumonia (Class 0)	Normal (Class 1)	accuracy	macro avg	weighted avg
precision	0.903409	0.832589	0.852564	0.867999	0.859147
recall	0.679487	0.956410	0.852564	0.817949	0.852564
f1-score	0.775610	0.890215	0.852564	0.832912	0.847238
support	234.000000	390.000000	0.852564	624.000000	624.000000

Results Summary: The results for accuracy and F1-score are summarized in the table below:

Model	Train Accuracy (%)	Test Accuracy (%)	Val Accuracy (%)	F1-score (Class 0)	F1-score (Class 1)
CNN (Benchmark)	95.07	77.04	73.33	0.60	0.84
VGG19 (Transfer Learning)	84.22	65.22	66.67	0.18	0.78
ResNet152 (Transfer Learning)	92.14	85.26	66.67	0.78	0.89

Confusion Matrix for all three models is summarized below:

Model	Class 0 (True Positive)	Class 0 (False Positive)	Class 1 (True Positive)	Class 1 (False Positive)
CNN (Benchmark)	106	128	377	13
VGG19 (Transfer Learning)	24	210	383	7
ResNet152 (Transfer Learning)	159	75	373	17

4. Conclusions:

From the above results, it can be concluded that the transfer learning model using the ResNet152 pre-trained model performs better than both the CNN benchmark model and the transfer learning model using VGG19. Although the CNN benchmark model has the highest accuracy of 95.07, the overall performance of the model is lower than that of the transfer learned (TL) ResNet152 model. This can be seen from the total number of false positives predicted by the CNN model for Class '0' (Pneumonia) is almost twice (128 versus 75) than that of the TL ResNet152 model. The TL VGG19 model's performance is not up to par for addressing our problem. Thus, we can conclude that although using the weighted loss, data augmentation, and transfer learning can improve the model's performance significantly, it is necessary to expand the training data sets to get a robust model.