Machine Learning Engineer Nanodegree

Capstone Project

Chest X-ray Pneumonia Prediction

Nimma Vishali

February 4th 2019

I. Definition

Project overview: Pneumonia is an infection in one or both lungs. It can be caused by bacteria, viruses, or fungi. Bacterial pneumonia is the most common type in adults. Pneumonia causes inflammation in the air sacs in your lungs, which are called alveoli. The alveoli fill with fluid or pus, making it difficult to breathe. According to National Institutes of Health (NIH), chest x-ray is the best test for pneumonia diagnosis. However, reading x-ray images can be tricky and requires domain expertise and experience. It would be nice if we can just ask a computer to read the images and tell us the results.

The more detailed information for pneumonia can be found in the following research paper https://www.papermasters.com/pneumonia.html.

Problem Statement:

The main aim of my project is to predict whether a person has Pneumonia by examining the X-ray report. For doing this I selected the dataset from Kaggle https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia Since my goal is to predict the presence of Pneumonia, Here I am using CNN Sequential model in Keras for classification models. My strategy is using this model adding the layers to increase the accuracy. Using convolution layers as needed,Dense layer to avoid overfitting,Flatten layer,MaxPooling layers.

***Metrics*:**

The metrics used is AUC. As my dataset is unbalanced, I would like to use auc as Evaluation metric for this categorical classifier. The auc function is defined as

```
def auc(y_true,y_pred):
   auc=tf.metrics.auc(y_true,y_pred)[1]
   K.get_session().run(tf.local_variables_initializer())
   return auc
```

During development, a validation set was used to evaluate the model. I want to use a small set of training images as my validation images. For validation I want to use “categorical_crossentropy” as loss metric for CNN, optimizer as “adam” and also metrics as “auc”.

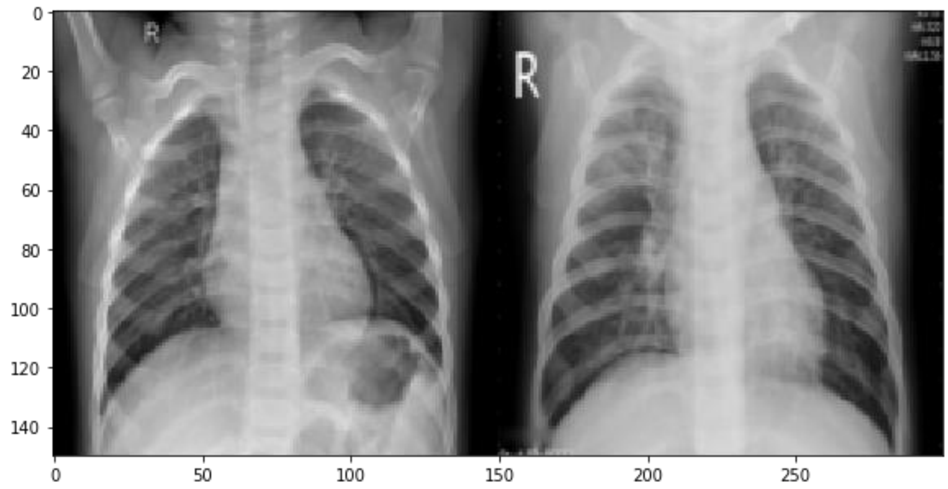
tf.metrics.auc has auc metrics the arguments used for this are auc(labels,predictions)where labels is Tensor whose shape matches predictions. Will be cast to bool. **predictions**: A floating point Tensor of arbitrary shape and whose values are in the range [0, 1].

Reference for this metrics
https://www.tensorflow.org/api_docs/python/tf/metrics/auc

II. Analysis

Data Exploration:

Classifying the X-ray images as Pneumonia and Normal Dataset contains totally 5856 images. There are two categories of the X-ray images



The first one id Normal X-ray and second one is Pneumonia X-ray.

The dataset that Iam working is downloaded from

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

This data set contains 5856 X-rays in which 1583 are normal images and 4273 are Pneumonia images. Photos are not high resolution. Photos are not reduced to a single size, they have different proportions. The total dataset is divided into validation set, testing set, training set.

Exploratory Visualization:

The dataset contains three subfolders namely test:

- NORMAL (234 images)
- PNEUMONIA (390 images)

train:

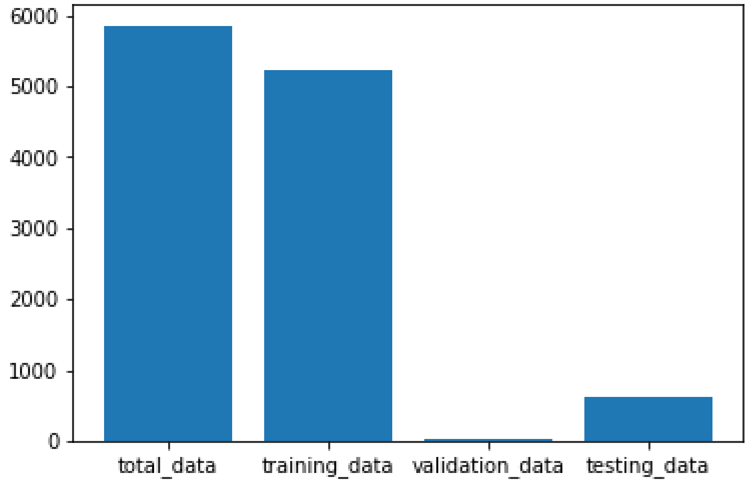
- NORMAL (1341 images)
- PNEUMONI (3875 images)

val:

- NORMAL (8 images)
- PNEUMONIA (8 images)

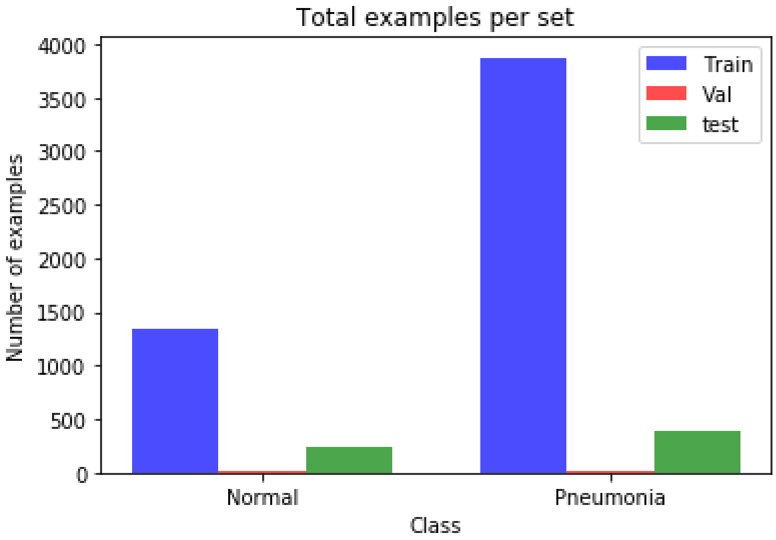
Total 1583 Normal images and 4273 Pneumonia images are present.

The visualization of total dataset divided in to train,test,val.



There are 5856 total images ,624 test set images,4236 test set images,16 validation set images.

Testing, Training, Validation categories



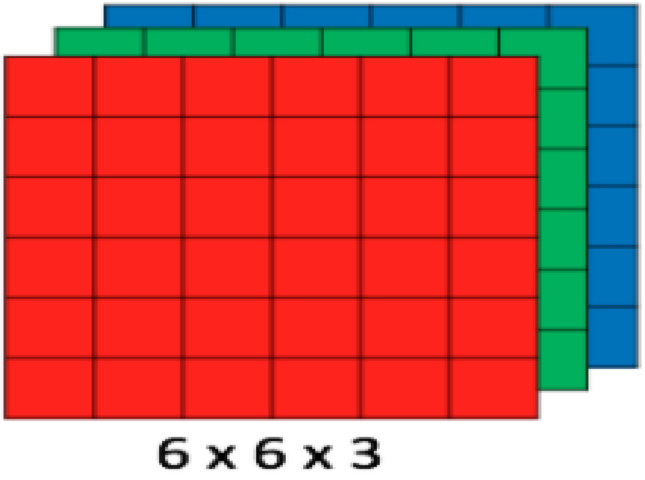
Algorithms and techniques:

The technique that I used is CNN Sequential model in Keras for classification. CNNs have wide applications in image and video recognition, recommender systems and natural language processing.

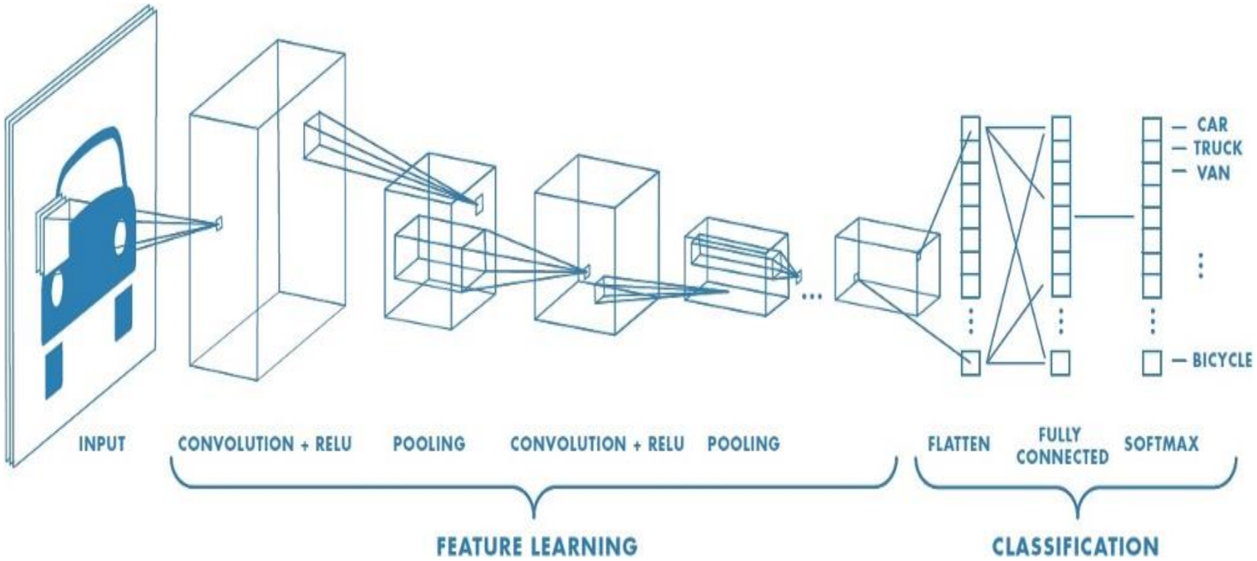
In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighted sum over them, pass it through an activation function and responds with an output. The whole network has a loss function and all the tips and tricks that we developed for neural networks still apply on CNNs.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d (h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1array of matrix of grayscale image



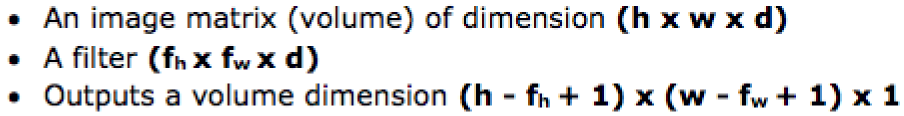
Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values.



Neural network with many convolutional layers

Convolution Layer

Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernal



- An image matrix (volume) of dimension **(h x w x d)**
- A filter **(f_h x f_w x d)**
- Outputs a volume dimension **(h - f_h + 1) x (w - f_w + 1) x 1**

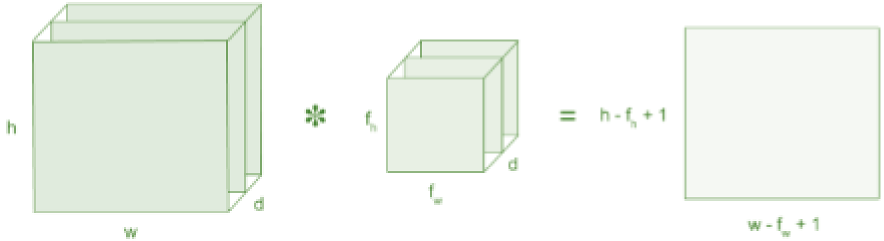
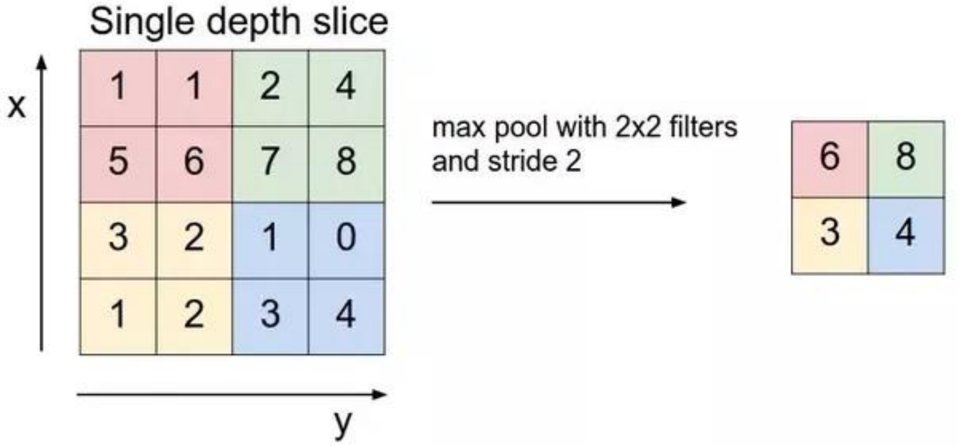


Image matrix multiplies kernel or filter matrix

Pooling Layer

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains the important information. MaxPooling Layer is used in this project.



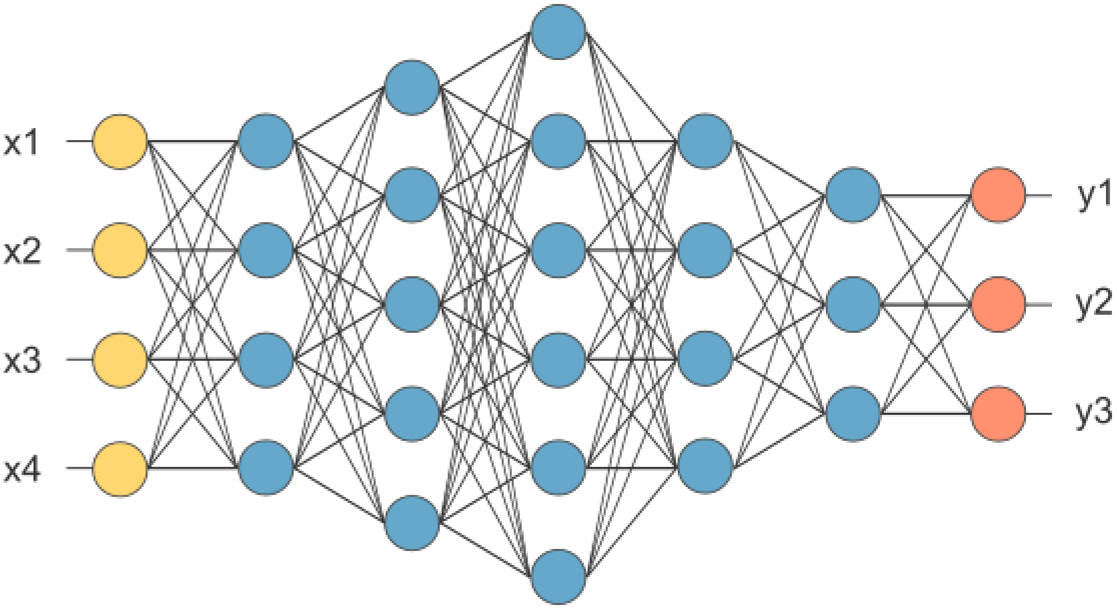
1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Exapmle of MaxPooling layer

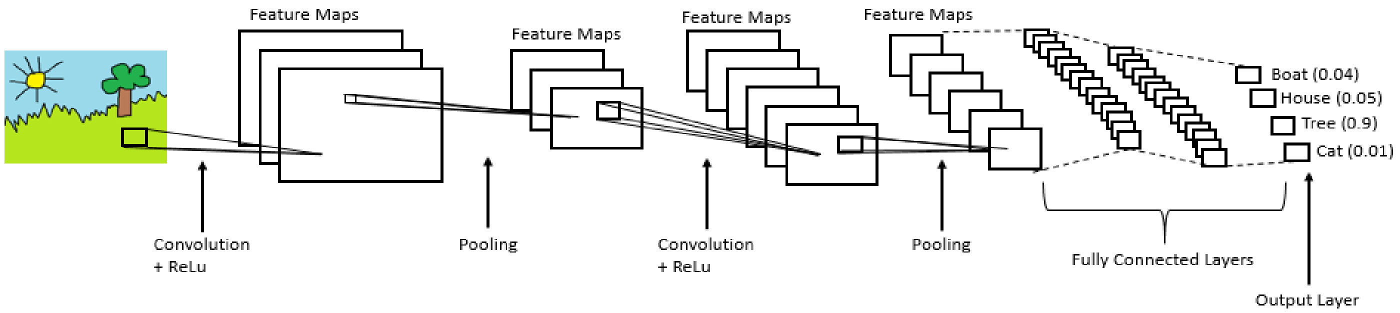
Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network. Considered an example



After pooling layer, flattened as FC layer

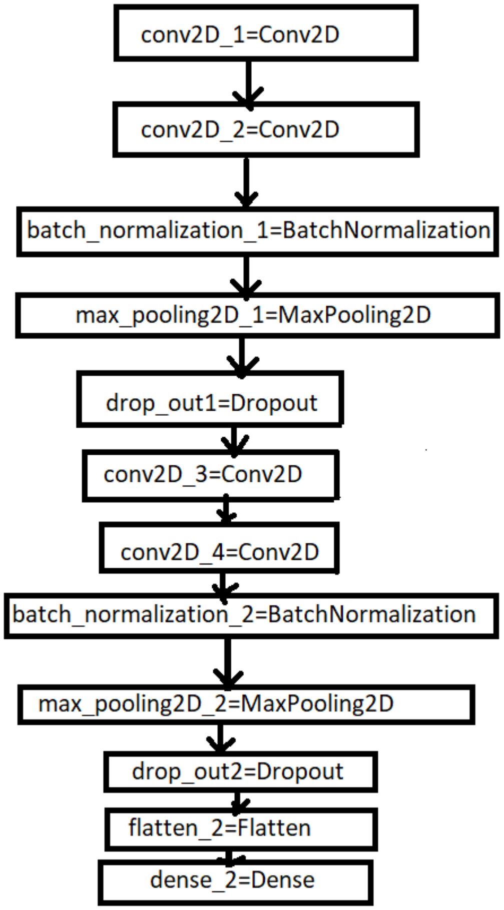
In the above diagram, feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax.



Complete CNN architecture

The above algorithm is used in the same manner to classify X-rays into Normal and Pneumonia.

The Convolution network graph for this project


conv2D_1=Conv2D
conv2D_2=Conv2D
batch_normalization_1=BatchNormalization
max_pooling2D_1=MaxPooling2D
drop_out1=Dropout
conv2D_3=Conv2D
conv2D_4=Conv2D
batch_normalization_2=BatchNormalization
max_pooling2D_2=MaxPooling2D
drop_out2=Dropout
flatten_2=Flatten
dense_2=Dense

Steps followed for design:

- Provided input image into convolution layer.
- Chosen parameters, apply filters with strides, padding if requires. Perform convolution on the image and apply ReLU activation to the matrix.
- Performed pooling to reduce dimensionality size.
- Added as many convolutional layers until satisfied.
- Flatten the output and feed into a fully connected layer (FC Layer)
- Configure the model for training.
- Using train function to train the model.
- Predicting the output I.e. classifying the images.

BENCHMARK MODEL:

Any CNN model that gives accuracy around 20% is my benchmark model. My created model Conv2D layer with 16 filters, kernel_size is equal to 2, padding is ‘same’ and input shape is (64, 64 ,1) and activation function as “relu”. Max Pooling layer with pooling size is equal to 2. flatten layer and Dense layers with 2 units and activation of "SoftMax" At compilation phase loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'] are used.For fit method in bench model batchsize is 16 and epchos are 4.

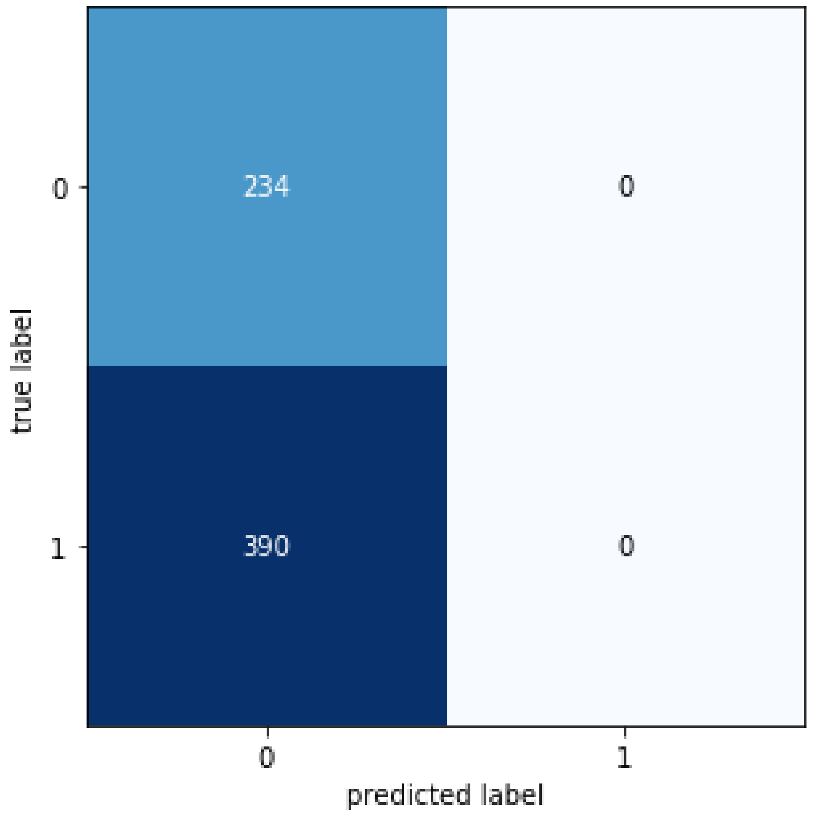
ARCHITECTURE:

```
_______________________________________________________________
Layer  (type)                         Output  Shape                       Param  #
=================================================================conv2d_1
(Conv2D)                    (None,   64,   64,   16)                           80
_________________________________________________________________max_pooling2d_1
(MaxPooling2    (None,     32,     32,     16)                                  0
_________________________________________________________________flatten_1
(Flatten)                     (None,   16384                                     0
_________________________________________________________________dense_1  (Dense)
(None,  2)                                           3277                        0
=================================================================Total    params:
32,850Trainable params: 32,850Non-trainable params: 0
```

Training accuracy:25.71% and Testing accuracy:50.00

CONFUSION MATRIX

The model not doing anything it is just classifying all the x-rays as Pneumonia.

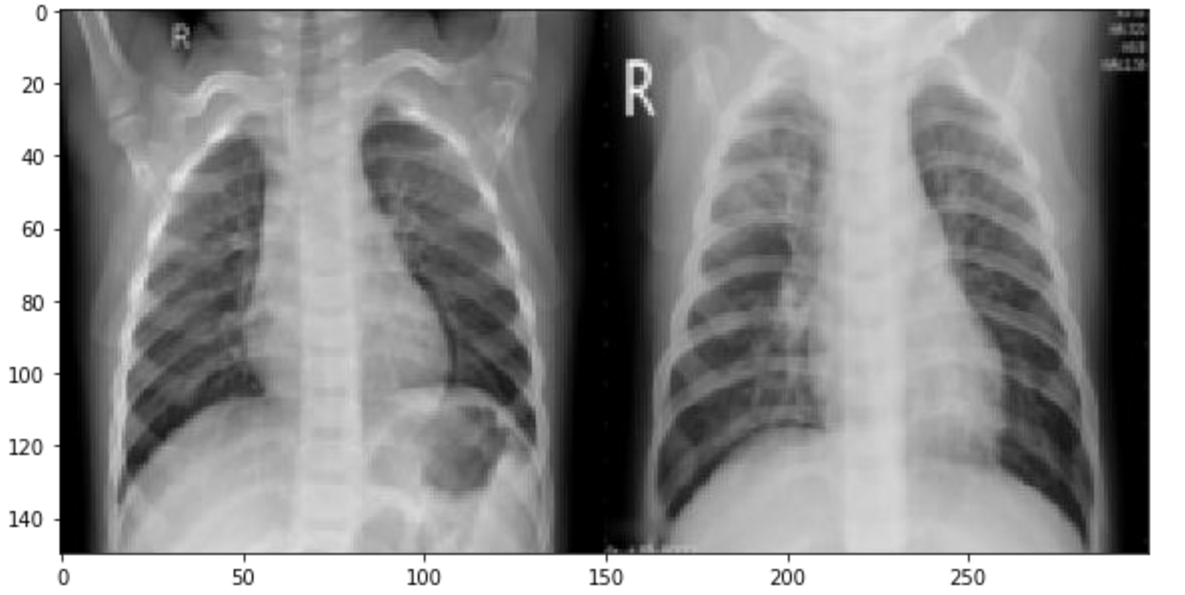


III. Methodology

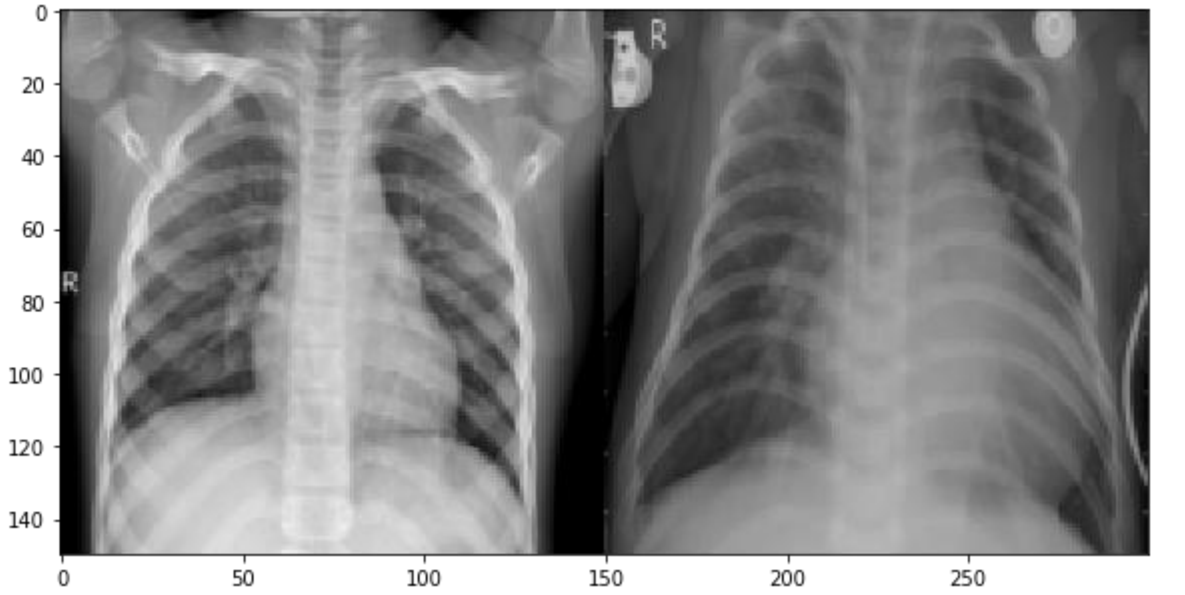
Data Preprocessing:

- Examining if there are any corrupted file in train or test or val dataset. As there are no corrupted images the datasets are left unchanged.
- The images are resized into 3D tensor with shape of width=150, height=150,channels=3 shaped array
- The dataset is visualized as below.

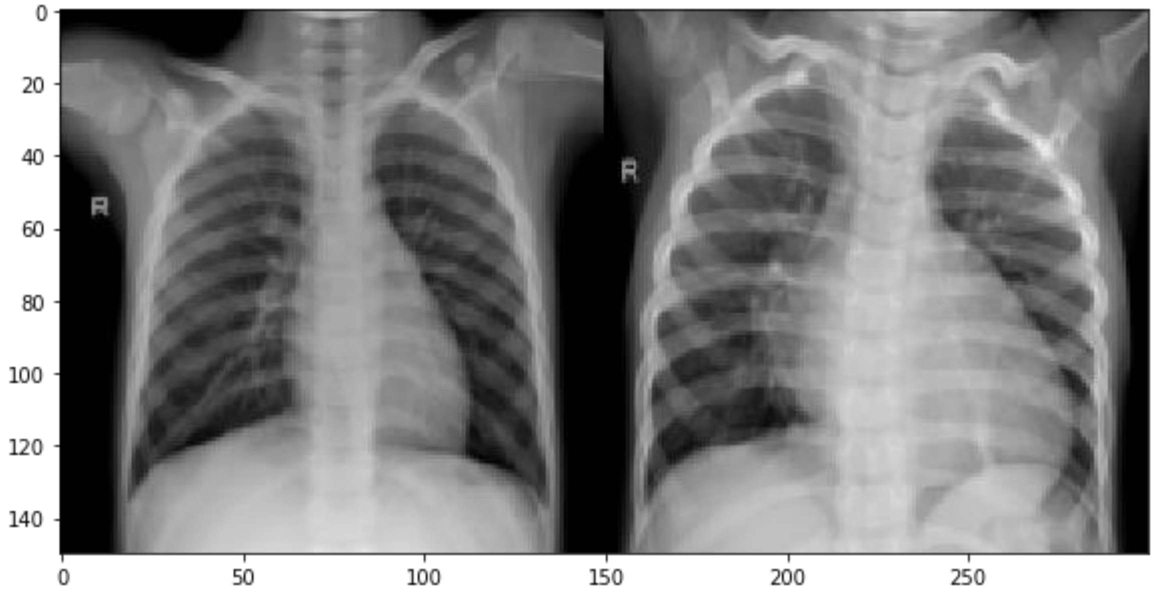
```
(Left) - No Pneumonia Vs (Right) - Pneumonia------------------------------------
-----------------------------------------------------------------------------
--------------------
```



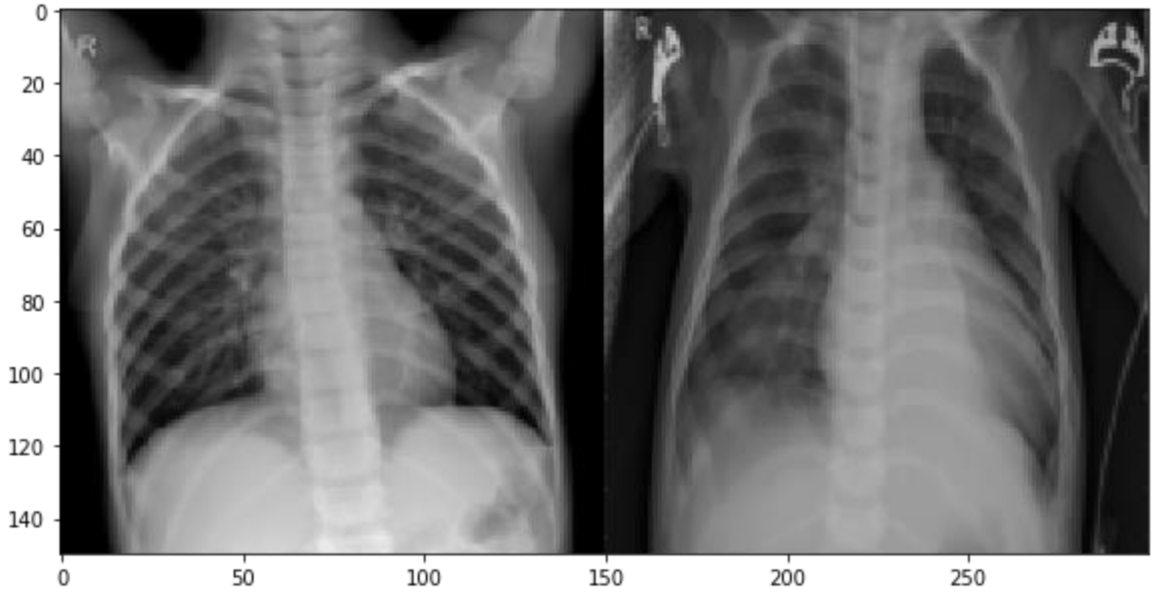
- (Left) - No Pneumonia Vs (Right) - Pneumonia---



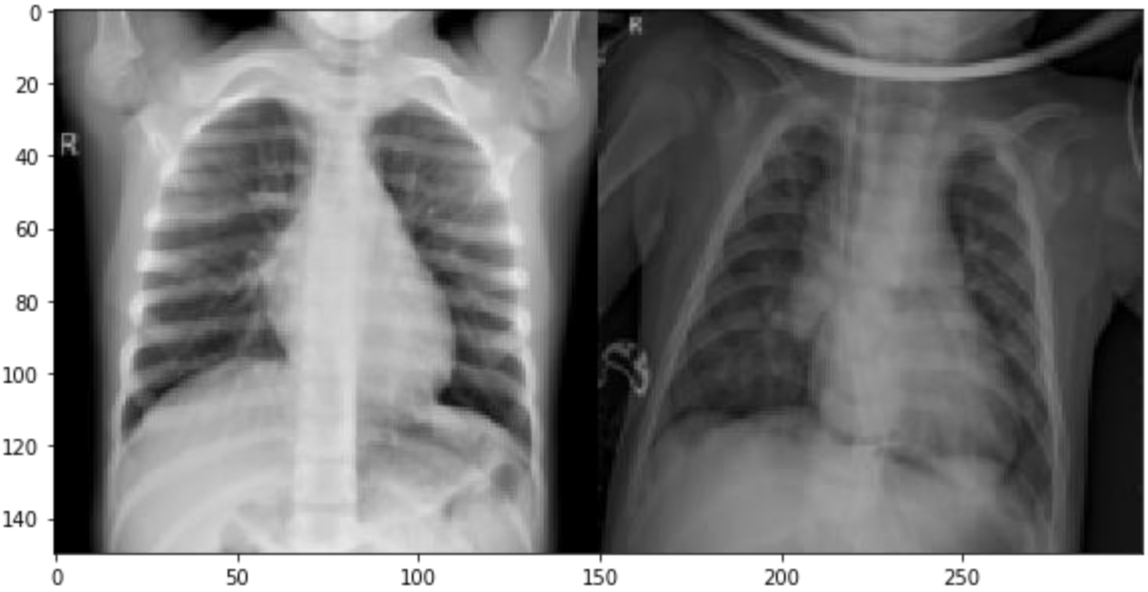
- (Left) - No Pneumonia Vs (Right) - Pneumonia---



- (Left) - No Pneumonia Vs (Right) - Pneumonia--



- (Left) - No Pneumonia Vs (Right) - Pneumonia--



As already the dataset is available as train set, test set, val set there is no to split the dataset.

Implementation:

As in implementation model I have created this model using CNN Sequential model.

TRAINING PHASE:

In this model four convolution layers are used with filters=32,32,64,64 and activation as “relu”, padding as “same”, and two maxpooling layers are used with pool_size=(2,2),one flatten layer, two batchnormalization layers, two dropouts with 0.25 order to avoid overfitting, Dense layer with 2 units and activation function as “softmax”.

ARCHITECTURE:

```
Layer(type)                                       Output  Shape                          Param#
===============================================================================conv2d_2
(Conv2D)                                     (None,64,64,32)                                320
_______________________________________________________________________________conv2d_3
(Conv2D)                                 (None,   64,   64,   32)                          9248
_______________________________________________________________________________batch_normalizat
ion_1     (Batch     (None,     64,     64,     32)                                         128
_________________________________________________________________max_pooling2d_2
(MaxPooling2    (None,     32,     32,     32)                                  0
_________________________________________________________________dropout_1(Dropou
t)                               (None,   32,   32,   32)                       0
_________________________________________________________________conv2d_4
(Conv2D)                                (None,   32,   32,   64)            18496
_________________________________________________________________conv2d_5
(Conv2D)                              (None,   32,   32,   64)              36928
_________________________________________________________________batch_normalizat
ion_2     (Batch     (None,     32,     32,     64)                           256
_________________________________________________________________max_pooling2d_3
(MaxPooling2    (None,     16,     16,     64)                                  0
_________________________________________________________________dropout_2
(Dropout)                           (None,   16,   16,   64)                    0
_________________________________________________________________flatten_2
(Flatten)                        (None,   16384)                                0
_________________________________________________________________dense_2  (Dense)
(None,    2)                                                                32770
=================================================================Total    params:
98,146Trainable params: 97,954Non-trainable params: 192
```

ARCHITECTURE:

COMPILATION PHASE:

- The trained model was compiled with “adam” as optimizer, loss function is “categorical_crossentrophy”, metrics as “auc”.
- The trained model uses fit method with batchsize=16 and epchos=4.

The dataset in train folder, test folder, val folder is used for training, testing, validation respectively.

Classification of X-rays are done finally using the predict method on Test dataset.

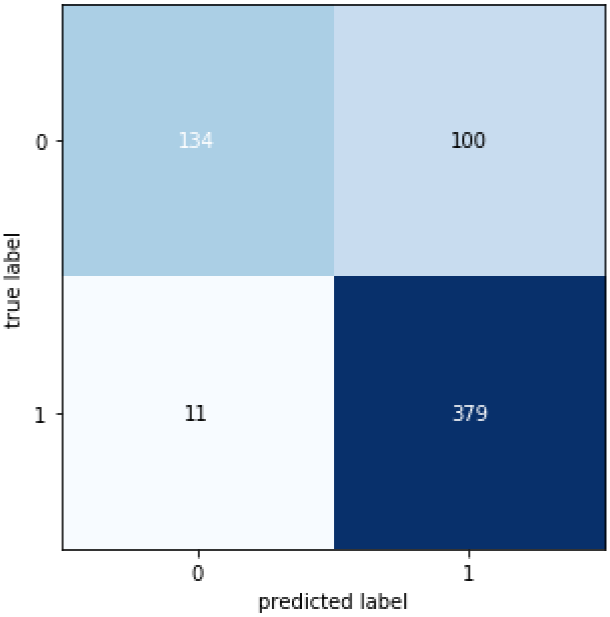
Refinement:

I have no complications occurred during the coding process because I used keras library to build CNN. Since it is high Level library for deep learning there is no complication during coding process. The main trouble was occurred when I am trying to tune the number of Convolution layers.

IV. Result

Model evaluation and validation:

CONFUSION MATRIX:

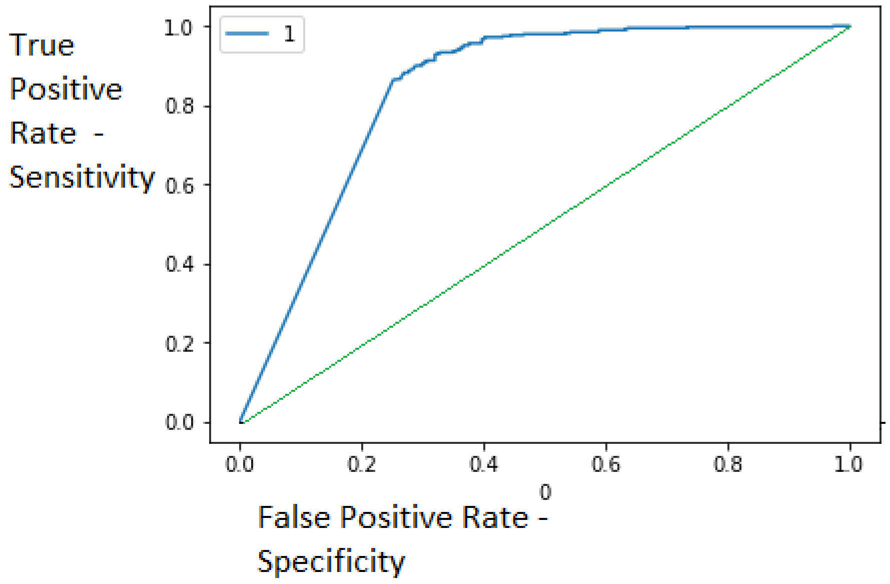


CLASSIFICATION REPORT:

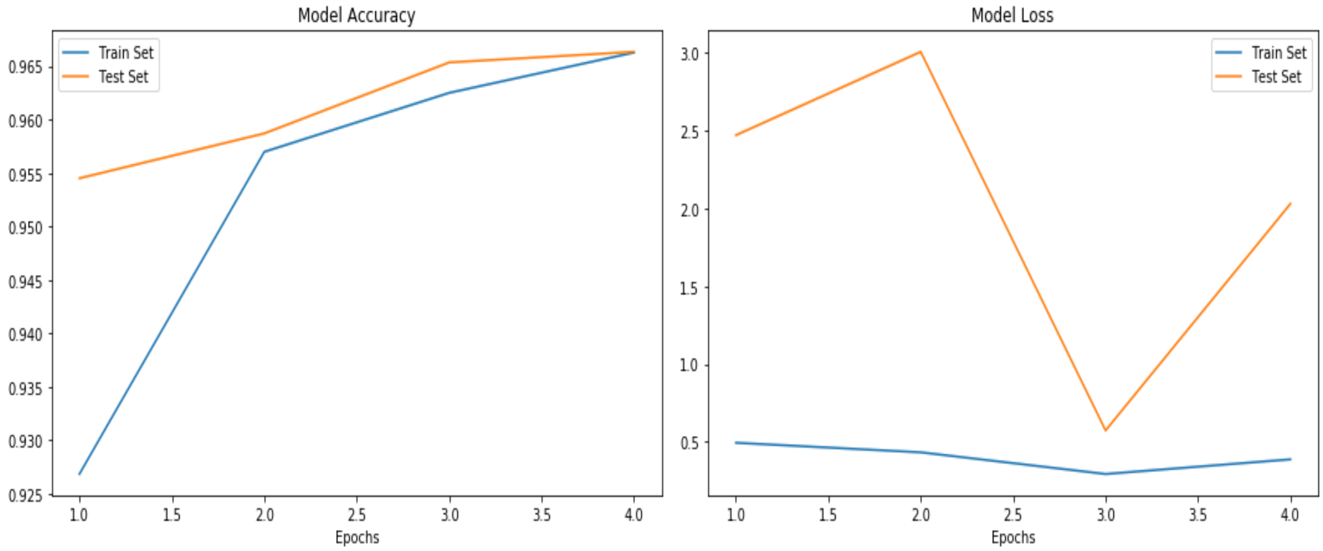
```
          precision   recall    f1-score    support      Normal      0.92       0.57
0.71          234   Pneumonia    0.79        0.97     0.87           390   micro avg
0.82       0.82    0.82           624

macro avg   0.86       0.77    0.79         624Weighted avg 0.84      0.82    0.81
624
```

ROC curve:



History of loss and accuracy for sequential CNN model:

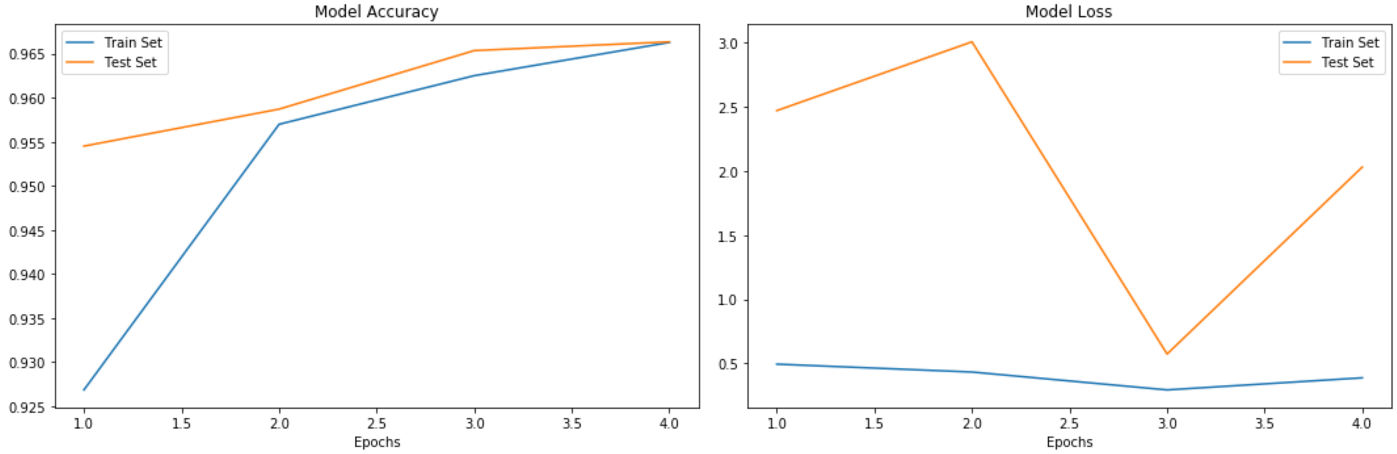


<u>Justification:</u>

When compared with my benchmark model, my model gives training accuracy up to 96.66% Whereas my testing accuracy is around 96.68%. The results obtained from my model above satisfactory as both training and testing have accuracy scores are above 95% which is pretty good. Now I can confidently say that my sequential cnn model solution significant to solve the problem.

V. Conclusion:

History of accuracy and loss for this model.



Reflection:

In this capstone project I have taken image classification as my thought of interest inspired from dog breed classifier. In this process I have learned many things.

- First thing I have learned about data retrieval processes. When I am doing research about retrieval process, I have come across a lot of surprising methods try. Out of all I decided to use load files method in cv2
- I have also learn how to use kaggle kernals and how to commit it and reproduce my work.
- Then I used my skills on representation of overall number of images, and also category wise in both training and testing images by using Matplotlib library. I have realized that its most informative in beginning in understanding size of your project.

- When I started loading images, I have found many useful methods like image. load_img in keras. preprocessing, cv2.load_img, etc., there are lots of them.
- I am also learn how to handle the corrupted images in dataset
- I learned that Image resizing is more important in image classification because we can't expect every image of same size.
- I came to know how to resize the image size and also how to add layers to the model to increase accuracy.I came to know about how numbers in image array changes with brightness.
- Here comes the heart of project, creating a CNN model, fitting it to training data and testing on test data evaluating validation curves, learn from confusion matrix doing modifications on models and there's more going on.
- Last but not least without visualizing results we can't trust the robustness of a model

<u>Improvement:</u>

Few things that can be improved from my model is we can use grid search for tuning hyper parameters but it takes so much time. And also data augmentation can be applied.

REFERENCE LINKS:

- https://keras.io/models/sequential/
- https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8
- https://machinelearningmastery.com/keras-functional-api-deep-learning/
- Dataset link https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

- https://www.med-ed.virginia.edu/courses/rad/cxr/pathology3chest.html
- https://www.drbeen.com/blog/the-basics-of-chest-x-ray-interpretation/
- https://keras.io/getting-started/sequential-model-guide/
- https://machinelearningmastery.com/keras-functional-api-deep-learning/
- https://brohrer.github.io/how_convolutional_neural_networks_work.html