PROJECT REPORT

INTERNSHIP PROJECT – Ai Council

By: Aravinda Raman J

E-mail: aravindaraman04@gmail.com

Problem Statement: Corona - COVID19 virus affects healthy people's respiratory systems, and chest X-Ray is one of the most significant imaging modalities for detecting the virus.

The objective of this project is to develop a Deep Learning Model to identify the X-Rays of healthy vs. Pneumonia (Corona) afflicted patients using the Chest X-Ray dataset, and use this model to power the AI application to test the Corona Virus in a faster phase.

Dataset Used: The dataset is a collection of Chest X-Ray images of people. It contains images of people who are healthy, those who are tested positive for COVID-19 or other viral and bacterial pneumonias such as SARS (Severe Acute Respiratory Syndrome), Streptococcus and ARDS (Acute Respiratory Distress Syndrome).

Dataset Link: https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset

There are 2 files present. One of them is the data and the other is the metadata. A sample of the data file is shown in Figure 1 and a sample of the metadata is shown in Figure 2.

	Unnamed:0	X_ray_image_name	Label	Dataset_type	Label_2_Virus_category	Label_1_Virus_category
0	0	IM-0128-0001.jpeg	Normal	TRAIN	NaN	NaN
1	1	IM-0127-0001.jpeg	Normal	TRAIN	NaN	NaN
2	2	IM-0125-0001.jpeg	Normal	TRAIN	NaN	NaN
3	3	IM-0122-0001.jpeg	Normal	TRAIN	NaN	NaN

Figure 1. Data file sample

	Unnamed: 0	Label	Label_1_Virus_category	Label_2_Virus_category	Image_Count
0	0	Normal	NaN	NaN	1576
1	1	Pneumonia	Stress-Smoking	ARDS	2
2	2	Pneumonia	Virus	NaN	1493
3	3	Pneumonia	Virus	COVID-19	58

Figure 2. Metadata file sample

Data Cleaning and preparation:

- As we can see from the data file sample, it has 2 categories in the 'Dataset_type' column: TRAIN and TEST. So, we separate the train and test images by filtering using the labels.
- Next, we fill all the places with NaN (Not a Number) with 'NA' string and we also append 'Label_2_Virus_category' column with the 'Label' column.
- We then check for all the label types like 'Normal/NA', 'Pneumonia/NA' and 'Pneumonia/COVID-19' in the train and test sets if they are present or not. We notice that 'Pneumonia/COVID-19' is not present in the test set. This is going to affect the prediction accuracy of the model. So, we take the last 600 examples of the train set and append it to the test set so that the data distribution with all the various labels is uniform and the overall model's accuracy is good.
- We then perform image data-augmentation on the train set to produce and add more images into the train set with varied orientations and other properties like zoom and brightness to improve the accuracy of the model.

Model Architecture:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 256, 256, 32)      896
_________________________________________________________________
activation (Activation)      (None, 256, 256, 32)      0
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 256, 32)      128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 256, 256, 32)      9248
_________________________________________________________________
activation_1 (Activation)    (None, 256, 256, 32)      0
_________________________________________________________________
batch_normalization_1 (Batch (None, 256, 256, 32)      128
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 128, 128, 32)      0
_________________________________________________________________
dropout (Dropout)            (None, 128, 128, 32)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 128, 64)      18496
_________________________________________________________________
activation_2 (Activation)    (None, 128, 128, 64)      0
_________________________________________________________________
batch_normalization_2 (Batch (None, 128, 128, 64)      256
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 128, 128, 64)      36928
_________________________________________________________________
activation_3 (Activation)    (None, 128, 128, 64)      0
_________________________________________________________________
batch_normalization_3 (Batch (None, 128, 128, 64)      256
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 64, 64, 64)        0
```

```
_________________________________________________________________
dropout_1 (Dropout)          (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 64, 128)       73856
_________________________________________________________________
activation_4 (Activation)    (None, 64, 64, 128)       0
_________________________________________________________________
batch_normalization_4 (Batch (None, 64, 64, 128)       512
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 64, 128)       147584
_________________________________________________________________
activation_5 (Activation)    (None, 64, 64, 128)       0
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 64, 128)       512
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 32, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 32, 32, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 131072)            0
_________________________________________________________________
dense (Dense)                (None, 3)                 393219
=================================================================
Total params: 682,019
Trainable params: 681,123
Non-trainable params: 896
_________________________________________________________________
```

Figure 3. Model Architecture

Algorithm Used: As we have to **classify** the data into 3 categories of outputs 'Normal/NA', 'Pneumonia/NA' and 'Pneumonia/COVID-19', I have chosen **CNN (Convolutional Neural Network).**

As we can see from Figure 3, we have the input shape as (256,256,3) and we have various Convolutional layers with a different number of filters and padding set to 'same'. With padding set to 'same', the image dimension remains the same after every convolutional layer which gives the model more scope to learn features along the edges of the image.

We have used L2 regularization with value 1e-4 which is a **technique used for tuning the function by adding an additional penalty term in the error function**. The additional term controls the excessively fluctuating function such that the coefficients don't take extreme values. We have the activation function as ReLU and we also have 'Batch Normalization' after every convolutional layer so that the model doesn't overfit and it also reduces the total number of epochs to train the model.

In the prefinal layer, we flatten the image into a feature vector and feed it to a Dense layer with 3 outputs which correspond to the 3 outputs 'Normal/NA', 'Pneumonia/NA' and 'Pneumonia/COVID-19' and use 'softmax' activation function which gives the probability to which the input image may belong among the 3 classes.

For optimizing the model, we use Adam optimizer with 0.0004 learning rate, 'categorical cross entropy' as loss function because it is a multi-class classification model.

We then train the model for 40 epochs with 3740/32 steps per epoch. The model yields a training accuracy of 94.76% and validation accuracy of 77.54%. The validation accuracy is a little low and can be improved. So, we perform 2 more epochs after changing the learning rate from 0.0004 to 0.0002 and we end up with a training accuracy of 95.28% and validation accuracy of 89.52%. Figure 4 and Figure 5 represent 'Training accuracy and validation accuracy' and 'Training loss and validation loss' respectively.

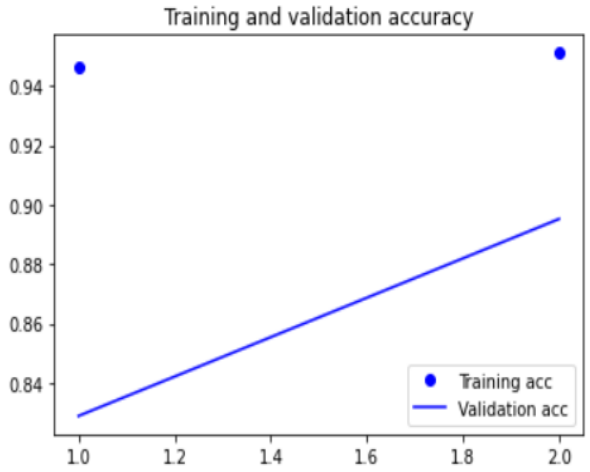

Figure 4

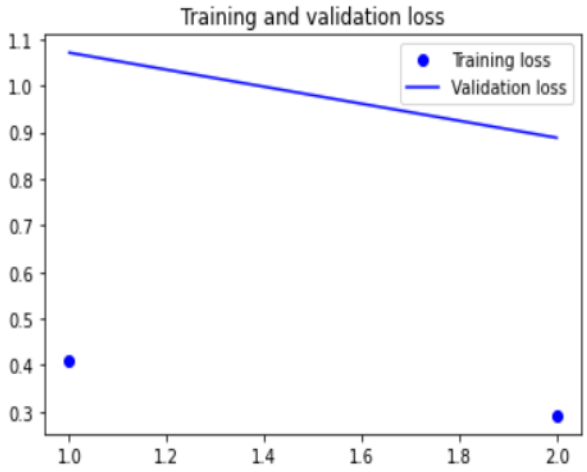

Figure 5

Software Package Used:

- Numpy 1.21.1
- Pandas v1.3.1
- Matplotlib 3.3.4
- TensorFlow 2.0
- Keras 2.3.0

Industrial scope and advantage of this project: Methods for detecting and classifying human illnesses from medical pictures that are automated using novel Machine Learning and Deep Learning Algorithms enable the doctor in driving the consultation in a better way, reducing the time it takes to diagnose the Corona Virus. This would also give physicians an edge and allow them to act with more confidence while they wait for the analysis of a radiologist by having a digital second opinion confirm their assessment of a patient's condition. Also, these tools can provide quantitative scores to consider and use in studies.

Code and its related files: https://github.com/aravinda-1402/Covid-Detection-model-using-Chest-X-ray.git