COVID-19 Chest X-Ray Data

Ashray Saraf saraf.ash@northeastern.edu

Abstract

The world was not prepared for the kind of pandemic that came and created hotspots around the world in 2019. The transmission rate of the virus was very high and in a couple of months it took over the world. Medical team all over the world took major steps in the detection and controlling the virus. Chest X-rays came out as one of the most quick, non-invasive, and reasonably the cheapest way to detect covid virus inside the body. Chest X-rays are manually looked up and with prior knowledge they were segmented in their desired classes. Looking at 100's of X-rays in a day there are always a chance of human error that may lead to some serious problems. The data was collected from the world's first publicly available database. The link to the database is: https://darwin.v7labs.com/v7-labs/covid-19-chest-x-ray-dataset

1. Introduction

We as a civilization has progressed a lot in terms of medical field, scientific computations, and the ways to handle a pandemic since the last pandemic that was as harmful as coronavirus disease 2019. We say a lot of hotspots emerging across the planet since the first detection of the virus. One of the main problems that the medical community faced with COVID-19 was that the rate of transmissibility of the virus was much faster than the rate of transmissibility of information about the virus. This kind of virus was not new to the world as something similar had occurred in 2012. But the way this virus affected became much deadlier. The main detection tool that was used to detect if COVID-19 virus is present inside the body or not was examining the chest X-ray of the patient. With the increasing number of cases there was a pileup of X-rays that were waiting to get examined. This manual process took a lot of crucial time which was spent just in the detection phase.

To solve this problem usage of AI was studied to know how this process could be streamlined. In China and Italy, hospitals have implemented AI-driven computed tomography (CT) scan interpreters, as well as AI initiatives to improve COVID-19 patient triaging (i.e., discharge, general admission, or ICU care) and hospital resource allocation.

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning system that can take an input image, assign relevance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them. When compared to other classification methods, the amount of pre-processing required by a ConvNet is significantly less. While basic approaches require hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training.

The dataset that was used for experimentation and evaluation is taken from the Darvin dataset by V7 labs. The link for the dataset is https://darwin.v7labs.com/v7-labs/covid-19-chest-x-ray-dataset. Below you could see how images are present in the dataset:

There are 517 cases of COVID-19 among the 6500 images of AP/PA chest X-Rays in the collection. Professional doctors labelled the photos in the dataset so that they could be segregated into distinct classifications. Normal Lung, Pneumonic Lung (Bacterial/Viral/Fungal), and COVID-19 positive lung are the three main tags used to label the photos. For images with a COVID-19 positive tag, additional tags such as age, sex, temperature, location, intubation status, ICU admission, and patient outcome are also provided.

2. Background

Recent years have seen an increase in interest in ML applications on CXR data, such as lung segmentation, tuberculosis and cancer analysis, neural networks have even claimed performance around radiologist levels. However, the adoption has its own challenges. The main risk that comes with using neural networks on a dataset for prediction over an image of a chest X-ray of a lung is the case of overfitting and performance overestimation.

3. Approach

Using a CNN has been always a go to approach while dealing with image classification. In recent times it has also found its usefulness in medical imagining analysis and has helped researchers in formulating the desired results.

Getting the dataset from the Darwin dataset proved to be a challenging task as the images were presented in a form of different .json files which contained the link to the image and their annotations with the desired segmentations. Getting all the images in one place was the first task that was to be performed by iterating every .json file and saving the images on the local disk. A snippet of the layout of the format of .json file is shown below:

```
{
  "image": {
    "seq": 6448,
    "width": 907,
    "height": 689,
    "filename": "00006448.png",
    "original_filename": "1-s2.0-S1341321X20301124-gr3_lrg-c.png",
    "url": "https://darwin.v7labs.com/api/images/1676206/original?token=952dd84f-7ac8-4251-873f-3b13
    "thumbnail_url": "https://darwin.v7labs.com/api/images/1676206/thumbnail?token=952dd84f-7ac8-425
    "workview_url": "https://darwin.v7labs.com/workview?dataset=901&image=6448"
  },
  "dataset": "COVID-19 Chest X-Ray Dataset",
  "annotated": true,
  "annotations": [
    {
      "name": "Lung",
      "polygon": {
        "path": [
          {
            "x": 555,
            "y": 106
          },
          {
            "x": 554,
            "y": 107
          },
          {
            "x": 554,
            "y": 108
          },
          {
            "x": 553,
```


After getting all the images on the local disk the image was loaded on the notebook and transformation techniques were performed on to the images so that the images have same size. Images were labelled into three categories namely – 'Covid', 'Normal', and 'Viral Pneumonia'. Cross Validation was performed to avoid overfitting and underfitting which has been one of the main concerns in medical imaging.

Image normalization is performed to ensure that every input pixel has a similar data distribution, doing so makes convergence faster while training the network. Since every image contains pixel value ranging from 0 to 255. We take every pixel from the image and divide that pixel with 255.

A basic architecture of how a CNN layer works can be seen below:

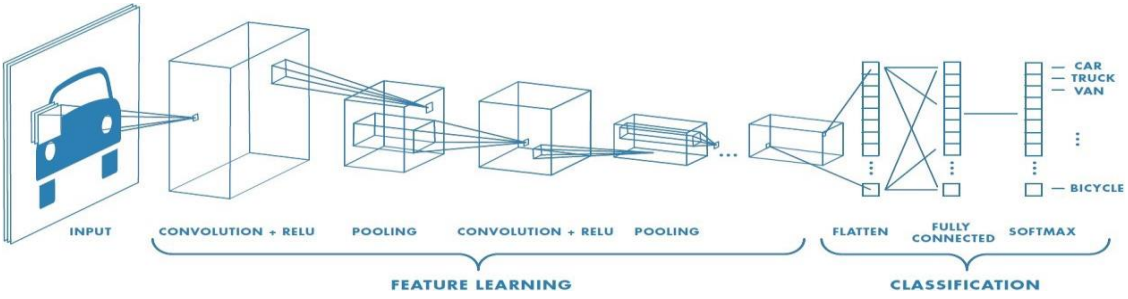


Figure 1: Basic CNN Architecture

Series of Convolutional layers followed by pooling layer were created. A convolution neuron layer is a very important and basic need of making a CNN. When performing an image classification task, a convolutional layer takes a series on metrices as input and multiple 2D metrices are generated as an output. The inputs and outputs of a convolutional layer can have different number of metrices. A CNN layer uses the formula below to compute a single output matrix –

$$A_j = f(\sum_{i=1}^{N} I_i * K_{i,j} + B_j)$$

Each input matrix *I* is multiplied with its corresponding kernel matrix, a bias value *B* is then added to each value of the convoluted matrix. The output of the resultant is then passed through a non-linear activation function *f* which converts all the metrices to a single

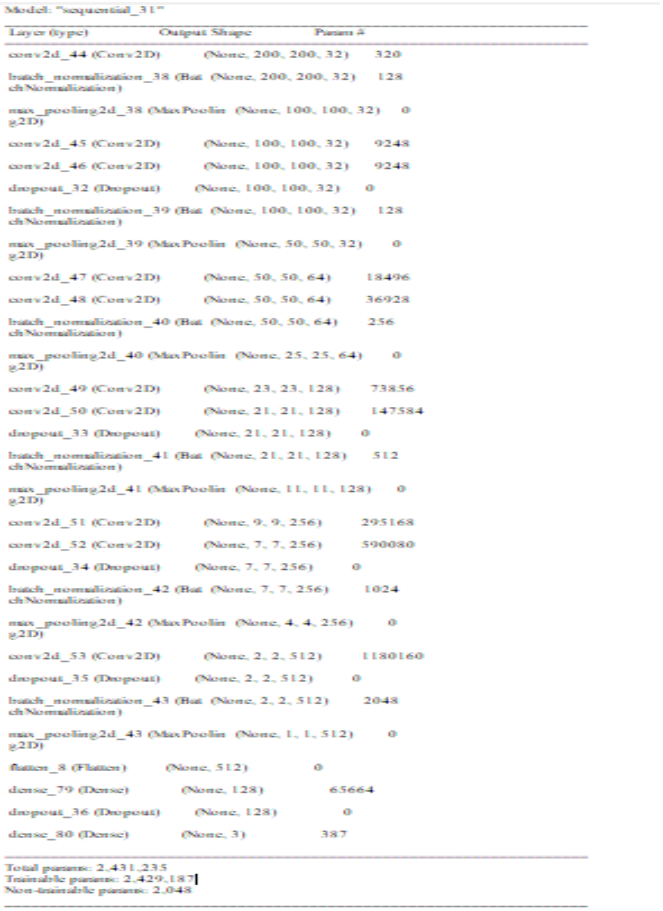
matrix. The size of the convolutional layer that was used in the project was (3,3), with a ReLU activation function. A dropout layer was also added, the function of which is to remove a certain sets of input units. The usage of a dropout layer is to make sure that we don't overfit the data. In this experimentation we used a dropout value of 0.2.

4. Results

The original dataset contained 517 cases of Covid-19 and 6500 images of AP/PA chest X-rays. Due to hardware limitations accessing these many images on the machine was not possible. So, the number of images were reduced to 137 cases of Covid-19. Images were labelled with the help of professional doctors so that they could be segregated in distinct classification. The dataset was further split into 80% – 20% for training and testing respectively. The proposed method consisted of 10 Convolutional layers with a low learning rate and a max epoch to 100.

The summary of the model can be seen in the fig.2 which shows all the layers that were added and the number of parameters that were computed.

The network has 18 layers: 10 convolutional layers, 6 pooling layers, 1 dense layer and 1 output layer. We tried to maintain the convolution block with 2 CNN's and 1 pooling layer, followed by a dropout layer characterized by a dropout rate of 20%. The CNN layer with a size of 3 x 3 kernels is used for feature extraction with an activation function of ReLU on top of it. The max pooling layer with a size of 2 x 2 kernels is then later used to reduce the dimension of an input image. After which the inputs are parsed thorough a series of densely connected network, at the end which is then used to predict whether they belong under any of the three categories (Covid, Normal, Viral Pneumonia).



```
Model: "sequential_31"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_44 (Conv2D)           (None, 200, 200, 32)      320
batch_normalization_38 (Bat  (None, 200, 200, 32)      128
ch_normalization)
max_pooling2d_38 (MaxPoolin  (None, 100, 100, 32)      0
g2D)
conv2d_45 (Conv2D)           (None, 100, 100, 32)      9248
conv2d_46 (Conv2D)           (None, 100, 100, 32)      9248
dropout_32 (Dropout)         (None, 100, 100, 32)      0
batch_normalization_39 (Bat  (None, 100, 100, 32)      128
ch_normalization)
max_pooling2d_39 (MaxPoolin  (None, 50, 50, 32)        0
g2D)
conv2d_47 (Conv2D)           (None, 50, 50, 64)        18496
conv2d_48 (Conv2D)           (None, 50, 50, 64)        36928
batch_normalization_40 (Bat  (None, 50, 50, 64)        256
ch_normalization)
max_pooling2d_40 (MaxPoolin  (None, 25, 25, 64)        0
g2D)
conv2d_49 (Conv2D)           (None, 23, 23, 128)       73856
conv2d_50 (Conv2D)           (None, 21, 21, 128)       147584
dropout_33 (Dropout)         (None, 21, 21, 128)       0
batch_normalization_41 (Bat  (None, 21, 21, 128)       512
ch_normalization)
max_pooling2d_41 (MaxPoolin  (None, 11, 11, 128)       0
g2D)
conv2d_51 (Conv2D)           (None, 9, 9, 256)         295168
conv2d_52 (Conv2D)           (None, 7, 7, 256)         590080
dropout_34 (Dropout)         (None, 7, 7, 256)         0
batch_normalization_42 (Bat  (None, 7, 7, 256)         1024
ch_normalization)
max_pooling2d_42 (MaxPoolin  (None, 4, 4, 256)         0
g2D)
conv2d_53 (Conv2D)           (None, 2, 2, 512)         1180160
dropout_35 (Dropout)         (None, 2, 2, 512)         0
batch_normalization_43 (Bat  (None, 2, 2, 512)         2048
ch_normalization)
max_pooling2d_43 (MaxPoolin  (None, 1, 1, 512)         0
g2D)
flatten_8 (Flatten)          (None, 512)               0
dense_79 (Dense)             (None, 128)               65664
dropout_36 (Dropout)         (None, 128)               0
dense_80 (Dense)             (None, 3)                 387
=================================================================
Total params: 2,431,235
Trainable params: 2,429,187
Non-trainable params: 2,048
_________________________________________________________________
```

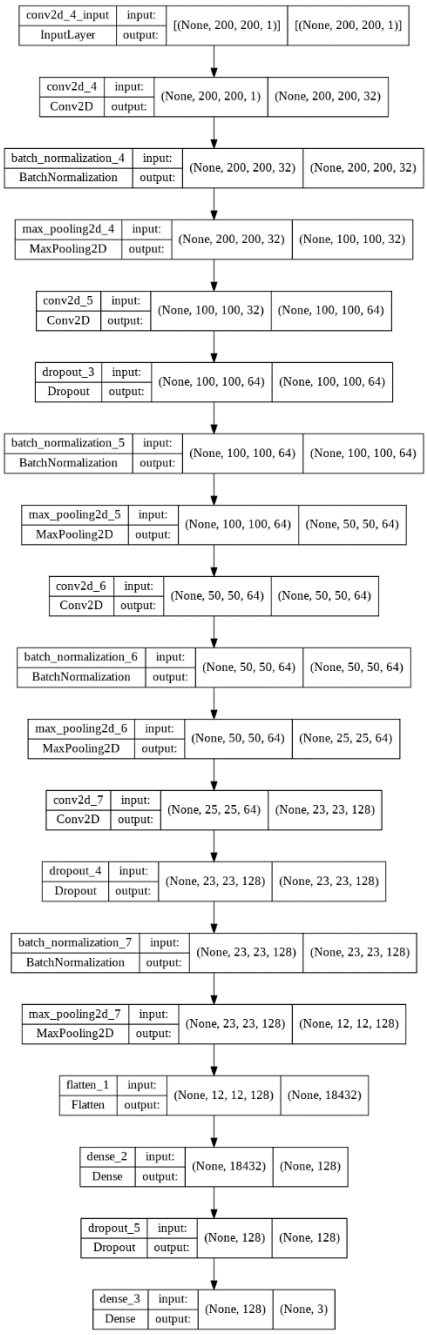


Figure 2: Model Summary

Different metrices were used to calculate the Accuracy of the model – in which TP(or True Positive) denoted the correctly predicted COVID-19 case, FP(or False Positive) denoted the misclassified Normal or Viral Pneumonia case that is classified as COVID-19 by the model, TN(or True Negative) denoted the correctly predicted Normal or Viral Pneumonia case, and lastly FN(or False Negative) denoted the misclassified COVID-19 case that is classified as Normal or Viral Pneumonia.

Accuracy is calculated as – (TP + TN) / (TN + FP + TP + FN)

In fig 3, we can see the confusion matrix that was obtained from the model. Where 0 represents Covid19, 1 represents Normal, and 2 represents Viral Pneumonia

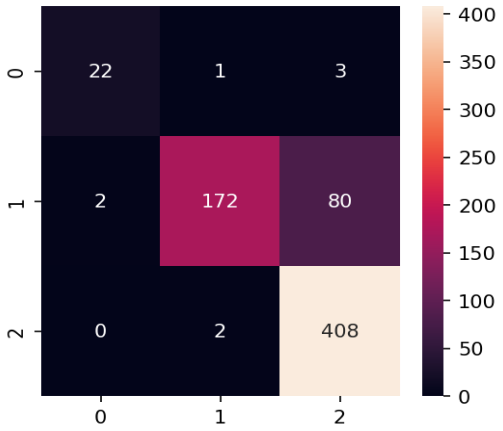
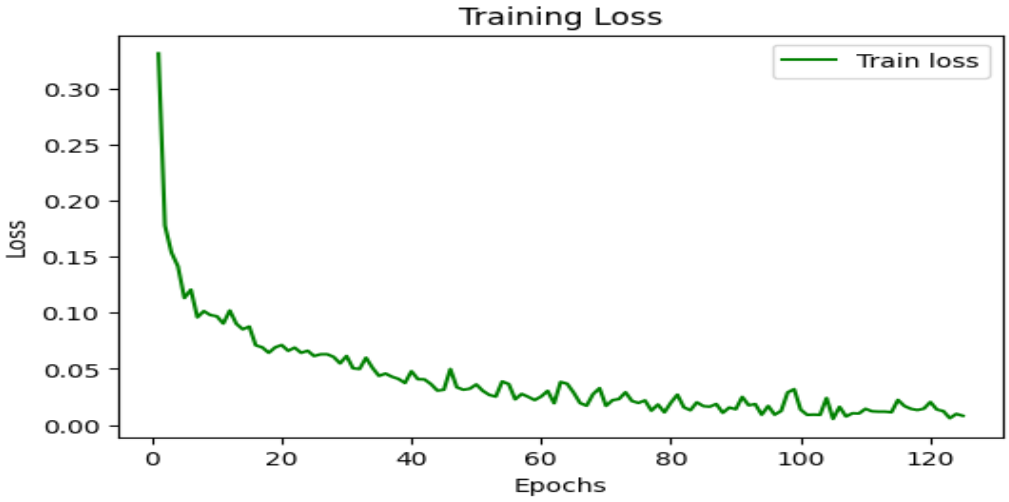
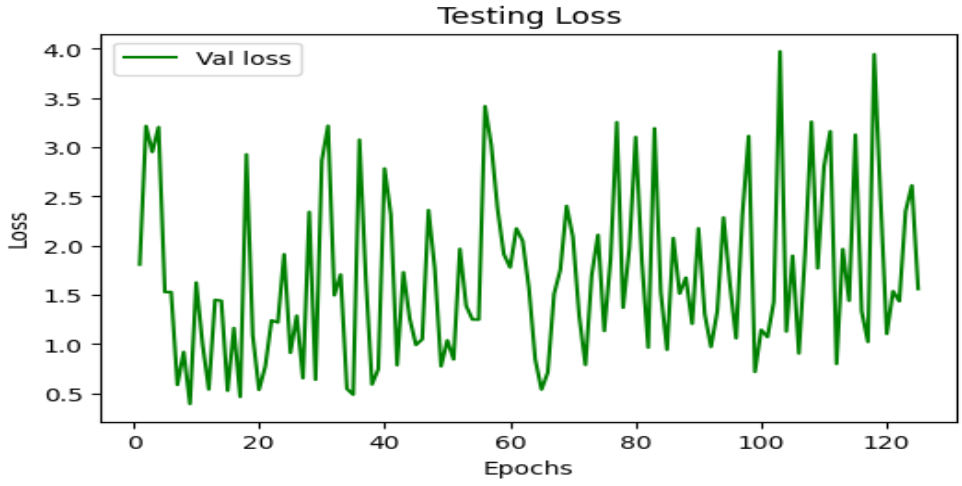


Figure 3: Confusion Matrix

We received an Accuracy Score of 87.24%, Precision Score of 89.01%, Recall Score of 75.5%, and F1 Score of 87.24%. An accuracy score of 86.6% tells us that the model classifies around 75% of images correctly as COVID-19. Training and Testing Loss over 125 epochs can be seen in the figures below.





5. Conclusion

In conclusion we can say that our model did perform at an accuracy score of 87.24%. With increasing cases of COVID-19 many countries are still facing shortage of resources during the pandemic. The main aim of the project was to make the model as accurate such that it identifies every single positive case. The model was built on a series of convolutional layers with a Nadam optimizer.

The proposed model also has a series of limitations such as the dataset was relatively small which needs to be increased for testing a generalized dataset. Our data consisted of only the posterior view of the X-ray, so our model learned only a particular view and it cannot differentiate among other views. Our model also cannot differentiate with other lung problems, and it would fail if any other kind of data is fed. Also, the accuracy scores as compared with the radiologists still needs to be worked upon.

6. References

Joseph Paul Cohen and Paul Morrison and Lan Dao COVID-19 image data collection, arXiv:2003.11597, 2020

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D.D. and Chen, M., 2014, December. Medical image classification with convolutional neural network. In 2014 13th international conference on control automation robotics & vision (ICARCV) (pp. 844-848). IEEE.

M.Z. Islam, M.M. Islam, A. Asaraf, A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images, Inform. Med. Unlock. 20(2020) 100412

Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labelled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, v2