SPOTLE.AI

CORONA HACKATHON

Detecting the Presence of COVID-19 in Patients From Chest X-Ray Images Using CNN-Deep Learning.

Group Member:

PRIYANK JHA
VIDHYA SUBRAMANIUM
DEVLEENA BANERJEE
KARTHIK
MODABBIR

PROBLEM STATEMENT

Develop an advanced machine learning based classifier that can scan chest X- rays and classify COVID19 positive cases and negative cases.

In this project, we are going to explore X-ray images as doctors frequently use X- rays and CT scans to diagnose pneumonia, lung inflammation, abscesses, and/or enlarged lymph nodes and try to detect positive COVID-19 cases from the given X- ray dataset.

- Positive Cases : https://github.com/ieee8023/covid-chestxray-dataset
- Normal Cases : https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

INTRODUCTION:

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus.

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

The best way to prevent and slow down transmission is be well informed about the COVID- 19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face.

The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it’s important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow).

At this time, there are no specific vaccines or treatments for COVID-19. However, there are many ongoing clinical trials evaluating potential treatments.

Artificial intelligence applied to the medical domain can have very real consequences.

COVID-19 tests are currently hard to come by — there are simply not enough of them and they cannot be manufactured fast enough, which is causing panic.

Given that there are limited COVID-19 testing kits, we need to rely on other diagnosis measures.

Since COVID-19 attacks the epithelial cells that line our respiratory tract, we can use X-rays to analyze the health of a patient's lungs.

And given that nearly all hospitals have X-ray imaging machines, it could be possible to use X-rays to test for COVID-19 without the dedicated test kits.

A drawback is that X-ray analysis requires a radiology expert and takes significant time — which is precious when people are sick around the world. Therefore developing an automated analysis system is required to save medical professionals valuable time.

DATASET PREPARATION:

Dataset was taken from following links:

- Positive Cases : https://github.com/ieee8023/covid-chestxray-dataset
- Normal Cases : https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

We have gathered all the images from the positive COVID images with view point of "PA" to our dataset because we have negative COVID images of same view only. This would make our model to perform better we suppose.

So, we ended up with 140 images of positive COVID X rays with view-'PA' out of 350 total images and same 140 number of images were taken from the Kaggle chest X rays for negative COVID images.

Finally, train-test split of 80-20 was taken to split the data into training and validation parts respectively. Total number of images in each folder before training are as shown below:

Train-Positive Images	115
Train-Negative Images	115
Validation-Positive Images	25
Validation-Negative Images	25

DETAILED DESCRIPTION AND ANALYSIS

- Firstly, we pre-process the given input X ray images and convert them to RGB channel ordering to make them ready for our CNN.
- We initialize the data augmentation generator object where each image is first normalized by dividing each image by 255 so that its pixels are now between 0 and 1.After that, rotation of image is done with shear-range provided of 0.2 and zoom-range of 0.2 and horizontal-flip also set to true.
- We instantiate the CNN model with nodes of size 32,64,64,128,128 in each CNN layer respectivel. Filters of size (3*3) are used to detect the patterns in images. After this, we used **Relu** activation function to map inputs to other layers.
- After that, we added the Max pooling layer comprising of pool size (2*2)after each convolutional layer and a drop-out of 0.25 is added to the model after each convolutional and max-pooling layer and append it to the model.
- To compile the network, we use the Binary cross-entropy loss and Adam optimizer. We have used **binary cross entropy** loss rather than categorical cross entropy because it is a class 2 problem.
- To start our CNN training process, we make a call to Keras fit generator method, while passing in our chest X-ray data via our data augmentation object.
- For evaluation, we first make predictions on the testing set and grab the prediction indices. We then generate and print out a confusion matrix report using scikit-learn helper utility and use it to derive the accuracy, sensitivity, and specificity.
- We then plot our training accuracy/loss history for inspection, outputting the plot to an image file.
- Finally we save our keras COVID-19 classifier model to disk.

RESULTS AND METRICS:

```
Model: "sequential_1"
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 222, 222, 32)      896
conv2d_2 (Conv2D)            (None, 220, 220, 64)      18496
max_pooling2d_1 (MaxPooling2 (None, 110, 110, 64)      0
dropout_1 (Dropout)          (None, 110, 110, 64)      0
conv2d_3 (Conv2D)            (None, 108, 108, 64)      36928
max_pooling2d_2 (MaxPooling2 (None, 54, 54, 64)        0
dropout_2 (Dropout)          (None, 54, 54, 64)        0
conv2d_4 (Conv2D)            (None, 52, 52, 128)       73856
```

```
max_pooling2d_3 (MaxPooling2 (None, 26, 26, 128)       0
dropout_3 (Dropout)          (None, 26, 26, 128)       0
conv2d_5 (Conv2D)            (None, 24, 24, 128)       147584
max_pooling2d_4 (MaxPooling2 (None, 12, 12, 128)       0
dropout_4 (Dropout)          (None, 12, 12, 128)       0
flatten_1 (Flatten)          (None, 18432)             0
dense_1 (Dense)              (None, 64)                1179712
dropout_5 (Dropout)          (None, 64)                0
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 1,457,537
Trainable params: 1,457,537
Non-trainable params: 0
```

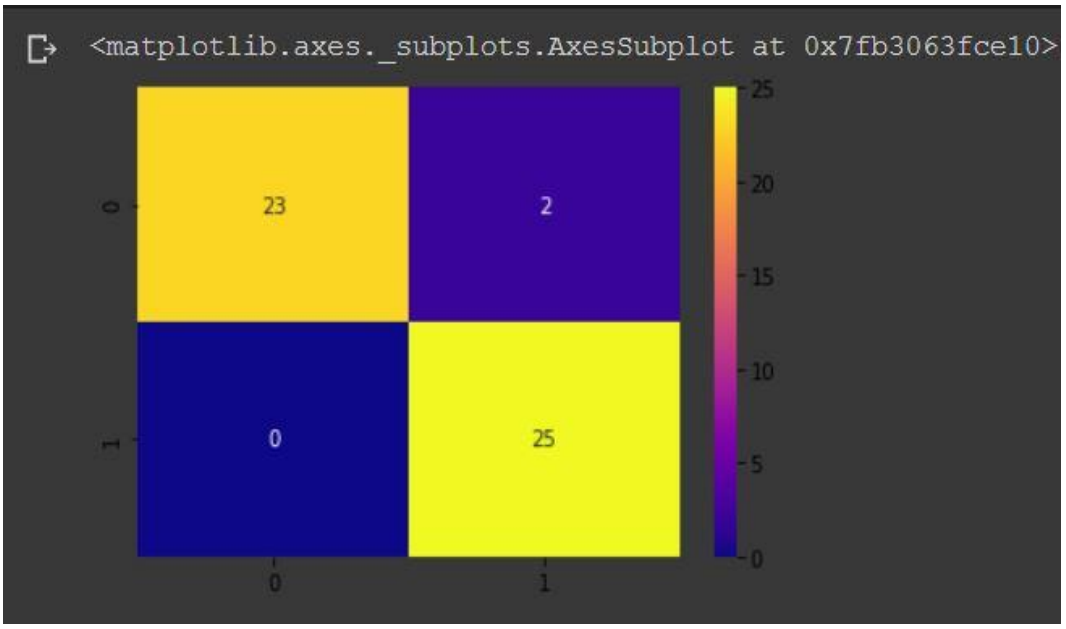
After 10 epochs of validation steps of 2 and batch size of 32, the following result was obtained:

```
Epoch 1/10
8/8 [==============================] - 179s 22s/step - loss: 0.8587 - accuracy: 0.5261 - val_loss: 0.6933 - val_accuracy: 0.5000
Epoch 2/10
8/8 [==============================] - 8s 1s/step - loss: 0.6903 - accuracy: 0.5348 - val_loss: 0.6920 - val_accuracy: 0.5000
Epoch 3/10
8/8 [==============================] - 9s 1s/step - loss: 0.6526 - accuracy: 0.6391 - val_loss: 0.6227 - val_accuracy: 0.9600
Epoch 4/10
8/8 [==============================] - 9s 1s/step - loss: 0.4976 - accuracy: 0.7435 - val_loss: 0.4181 - val_accuracy: 0.9600
Epoch 5/10
8/8 [==============================] - 9s 1s/step - loss: 0.3209 - accuracy: 0.8957 - val_loss: 0.1838 - val_accuracy: 0.9200
Epoch 6/10
8/8 [==============================] - 9s 1s/step - loss: 0.3156 - accuracy: 0.9087 - val_loss: 0.0912 - val_accuracy: 0.9600
Epoch 7/10
8/8 [==============================] - 9s 1s/step - loss: 0.2250 - accuracy: 0.9217 - val_loss: 0.2611 - val_accuracy: 0.9400
Epoch 8/10
8/8 [==============================] - 9s 1s/step - loss: 0.1297 - accuracy: 0.9391 - val_loss: 0.2291 - val_accuracy: 0.9600
Epoch 9/10
8/8 [==============================] - 9s 1s/step - loss: 0.1411 - accuracy: 0.9478 - val_loss: 0.0359 - val_accuracy: 0.9600
Epoch 10/10
8/8 [==============================] - 9s 1s/step - loss: 0.1225 - accuracy: 0.9522 - val_loss: 0.1918 - val_accuracy: 0.9600
```

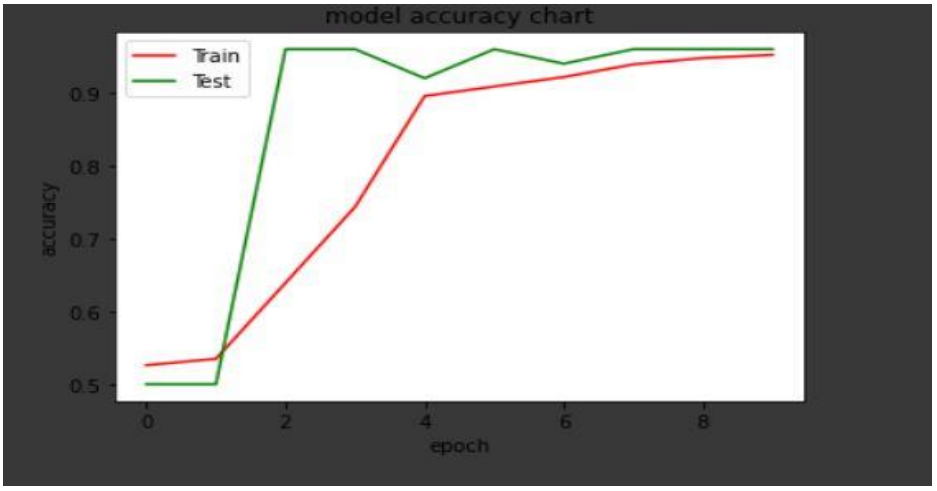
As we can see from the above results, COVID-19 detector is obtaining approximately **96% accuracy** on our testing dataset based solely on X-ray images We also obtain **100% sensitivity and 98% specificity** implying that:

- Of patients that do have COVID-19 (i.e., true positives), we could accurately identify them as "COVID-19 positive" **100%** of the time using our model.
- Of patients that do not have COVID-19 (i.e., true negatives), we could accurately identify them as "COVID-19 negative" only **98%** of the time using our model.

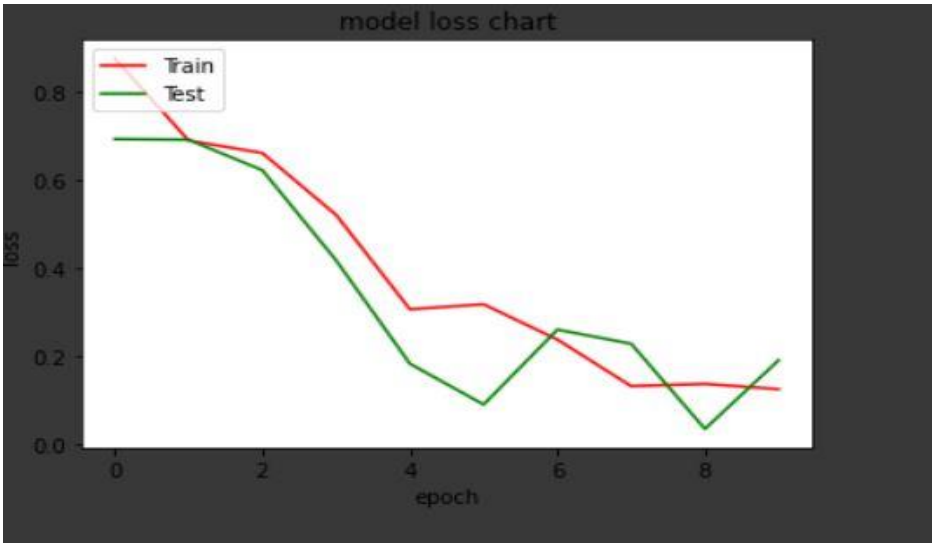
Training Accuracy(CNN)	95.22%
Testing Accuracy(CNN)	96.00%



As our training history plot shows, our network is not overfitting, despite having very limited training data.



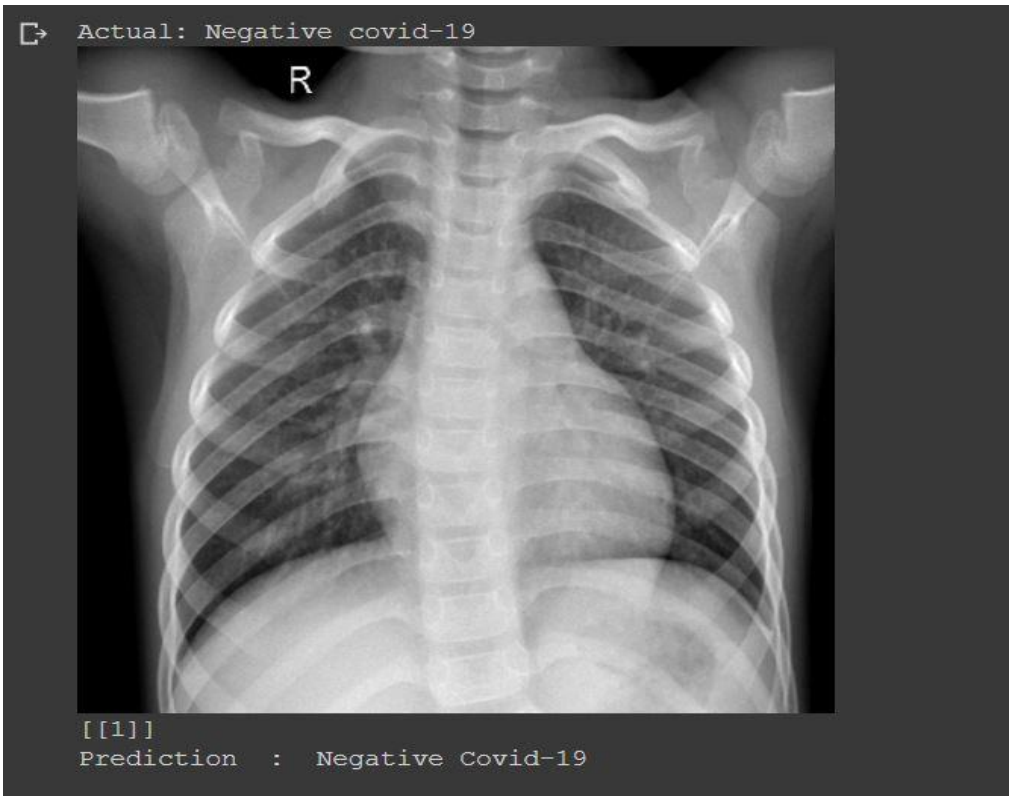
Accuracy chart

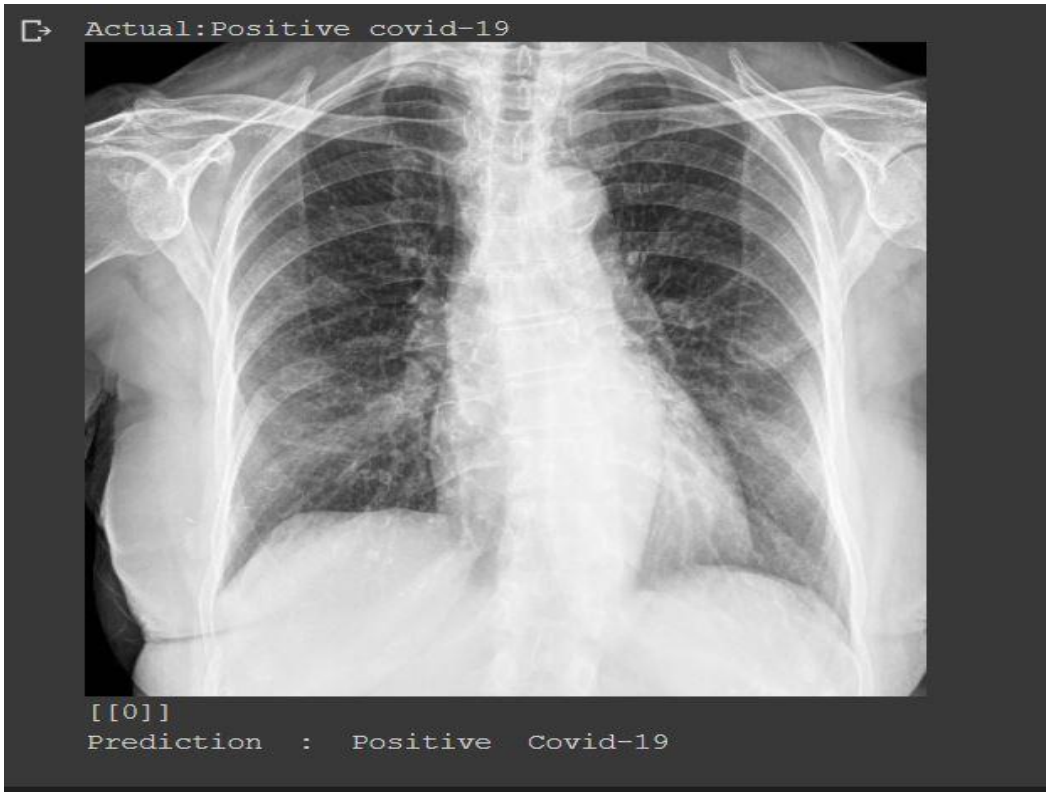


Loss chart

Model Prediction:

As we can see that model has predicted the same label which was shown in the actual label.





LIMITATIONS AND FUTURE IMPROVEMENTS:

- Lack of enough reliable data to train our CNN.
- Hospitals are already overwhelmed with the number of COVID-19 cases, and given patients rights and confidentiality, it becomes even harder to assemble quality medical image datasets in a timely fashion.
- In the next 12-18 months we'll have more high quality COVID-19 image datasets; but for the time being, we can only make do with what we have.
- The possibility remains that our model is learning patterns that are not relevant to COVID-19, and instead are just variations between the two data splits (i.e., positive versus negative COVID-19 diagnosis).
- Our future (and better) COVID-19 detectors will be multi-modal.
- Currently we are using only image data (i.e., X-rays) — better automatic COVID-19 detectors should leverage multiple data sources not limited to just images, including patient vitals, population density, geographical location, etc. Image data by itself is typically not sufficient for these types of applications.

REFERENCES:

Data Modelling & Analysing Coronavirus (COVID19) Spread - https://in.springboard.com/blog/data-modelling-covid/

WHO – Health topics/ Coronavirus

Kaggle – Coronavirus datasets - https://www.kaggle.com/tags/covid19

Github - https://github.com/ieee8023/covid-chestxray-dataset/tree/master/images

Cornell University - Prediction of COVID-19 Disease Progression in India : Under the Effect of National Lockdown - https://arxiv.org/abs/2004.03147?fbclid=IwAR0C415fGKdmbDlJxsOLnOibhtNJ20sdg2nfQvPX_HnBV9c-brZrYnwe2co0

Ministry of Health and Family Welfare - https://www.mohfw.gov.in/