PREDICTION OF COVID 19 USING CHEST X-RAY IMAGES

UNDER THE SUPERVISION OF

MS. KHUSHBOO CHHIKARA

UTKARSH BHARDWAJ 2K18/CO/379
VISHESH DARGAN 2K18/CO/389

DEPARTMENT OF COMPUTER ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(FORMERLY DELHI COLLEGE OF ENGINEERING)

OBJECTIVE

The tests currently being used to detect COVID 19 are very accurate and can detect the presence of the novel corona virus with precise accuracy . However, it is important to take note that these tests are very time consuming and complex. Our motivation behind making this model was that not all places have sufficient medical facilities available and with the increasing number of COVID 19 cases, healthcare facilities are diminishing and are inaccessible to people in the rural areas. A great doctor might outperform an AI algorithm but an AI algorithm performs more consistently and better than an average doctor. Hence, we wanted to come up with a testing methodology that is both inexpensive and easily accessible.

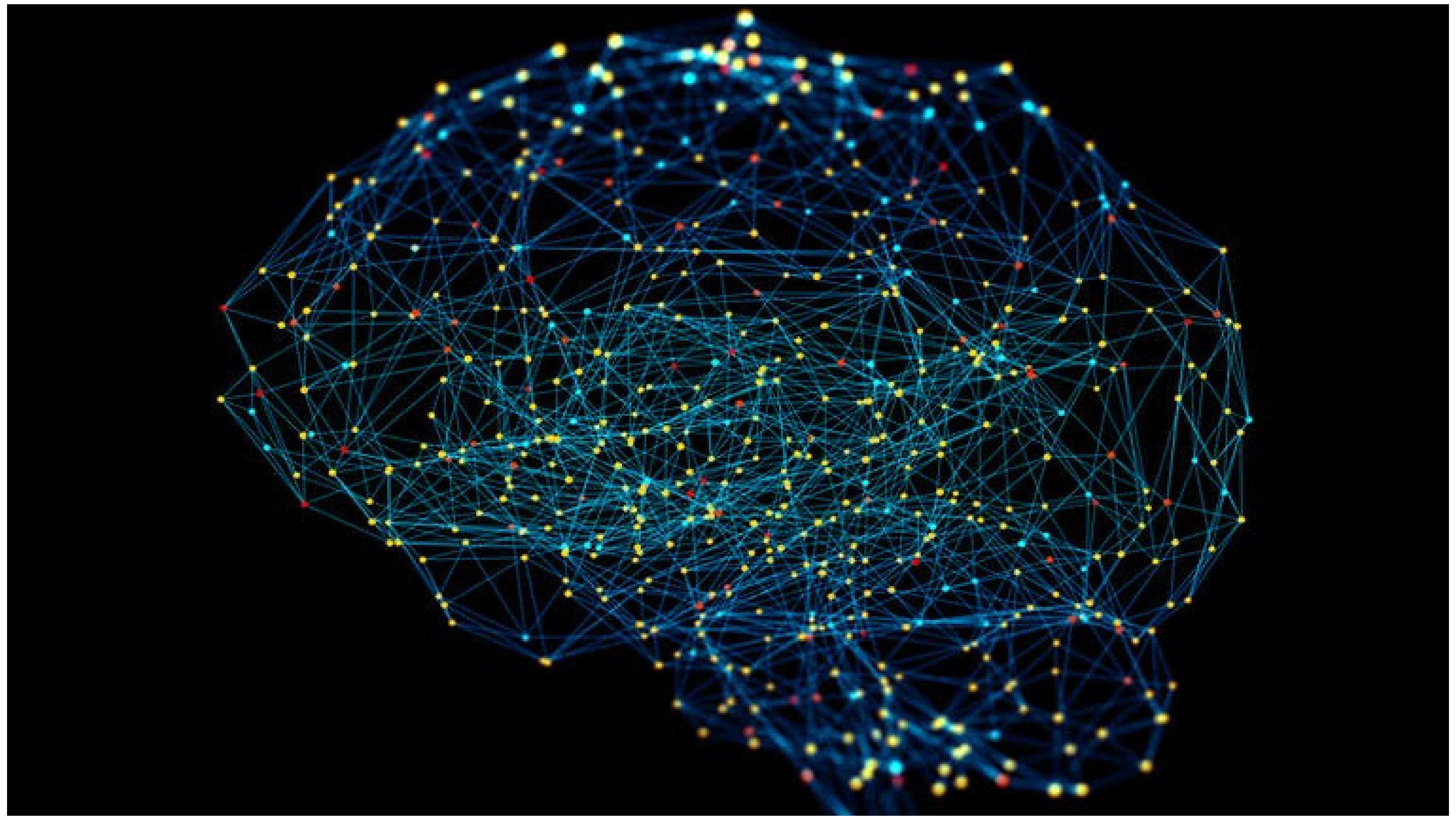
OBJECTIVE

We identified 2 problems with the existing models –
a) Some of them didn't take into account that COVID and pneumonia have very similar radiology results.

b) Some of them leveraged transfer learning which made the model quite big in size and not easy to deploy without using paid services.
Hence we have designed a Machine Learning model based on Convolutional Neural Networks to predict COVID - 19, viral pneumonia and healthy patients separately using Chest X-Ray images.

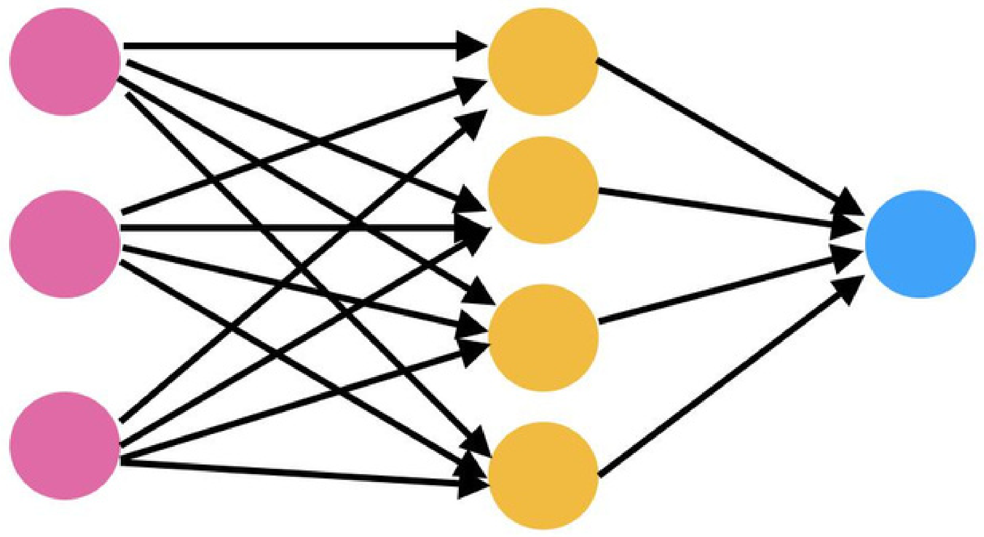
What is a Neural Network

A neural network is a network or chain of neurons, or in today's terms, an artificial neural network made up of artificial neurons or nodes. Thus, a neural network can be either a biological neural network (made up of actual biological neurons) or an artificial neural network (made up of artificial biological neurons) for solving artificial intelligence (AI) problems.



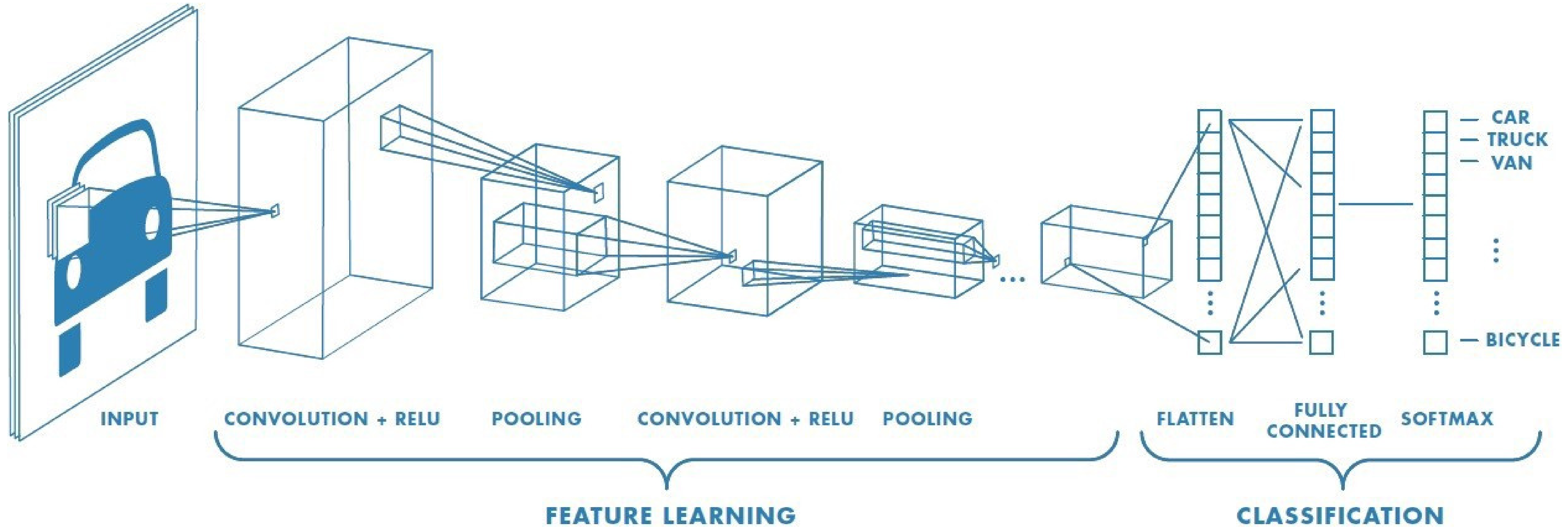
What is a Neural Network

The biological neuron's relations are modeled as weights. An excitatory relation has a positive weight, whereas inhibitory connections have a negative weight. A weight is applied to all inputs before they are summed. A linear combination is the name for this operation. Finally, the output's amplitude is regulated by an activation mechanism. For example, an acceptable range of output is usually between 0 and 1, or it could be −1 and 1.



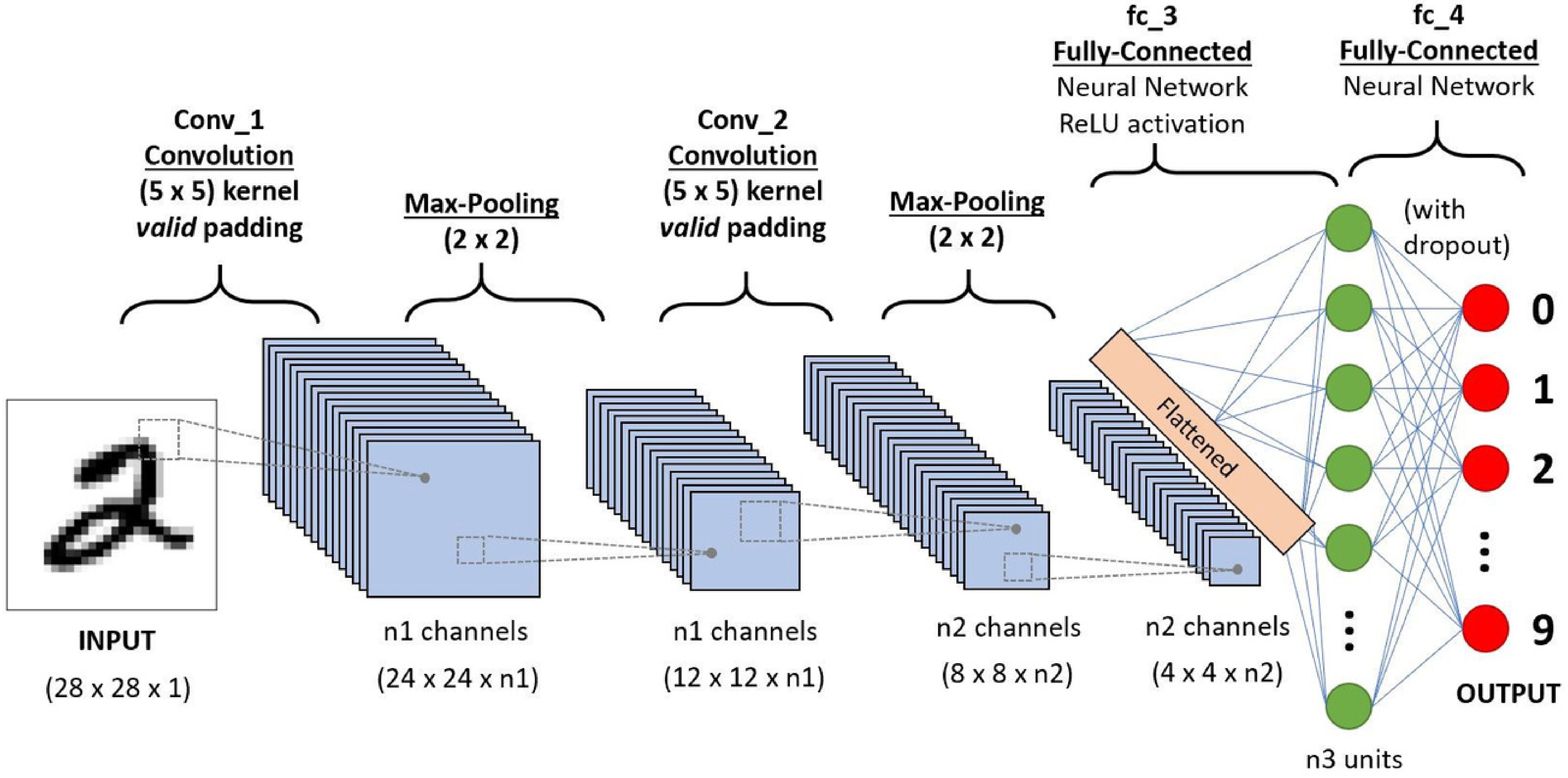
What is a CNN

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take an input image, attach significance (learnable weights and biases) to various aspects/objects in the image, and distinguish between them. As compared to other classification algorithms, the amount of pre-processing needed by a ConvNet is significantly less. Although primitive methods require hand-engineering of filters, ConvNets can learn these filters/characteristics with enough training.



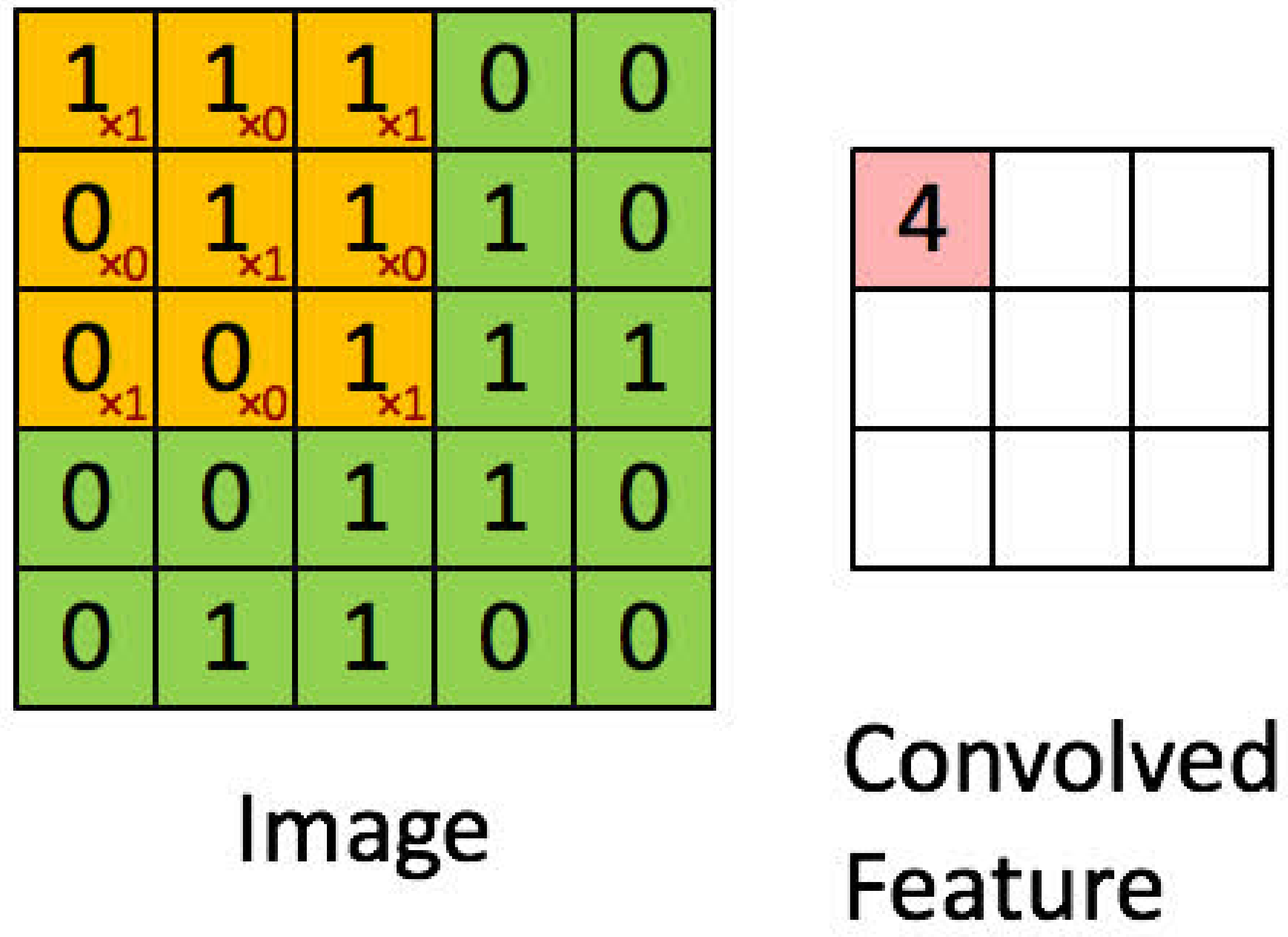
What is a CNN

The architecture of a CNN is inspired by the organization of the Visual Cortex and is similar to the connectivity pattern of Neurons in the Human Brain. Individual neurons can only respond to stimulus in a small area of the visual field called the Receptive Field. A set of such fields may be stacked on top of each other to occupy the whole visual field.



What is a CNN

Convolution extracts the predominant features in an image. What's interesting about CNNs is that during the back propagation, we update not only the weights and biases but also the kernel/ filter itself. This way after enough iterations through a number of data points, each kernel represents some sort of predominant feature of an object that could be undetectable to even the naked eye.



THE DATSET

The dataset we have chosen for our study is the COVID - 19 Radiography Dataset from Kaggle. The dataset contains Chest X-Ray images for COVID-19 positive cases along with Normal and Viral Pneumonia images. It constitutes 219 COVID-19 positive images, 1341 normal images and 1345 viral pneumonia images.

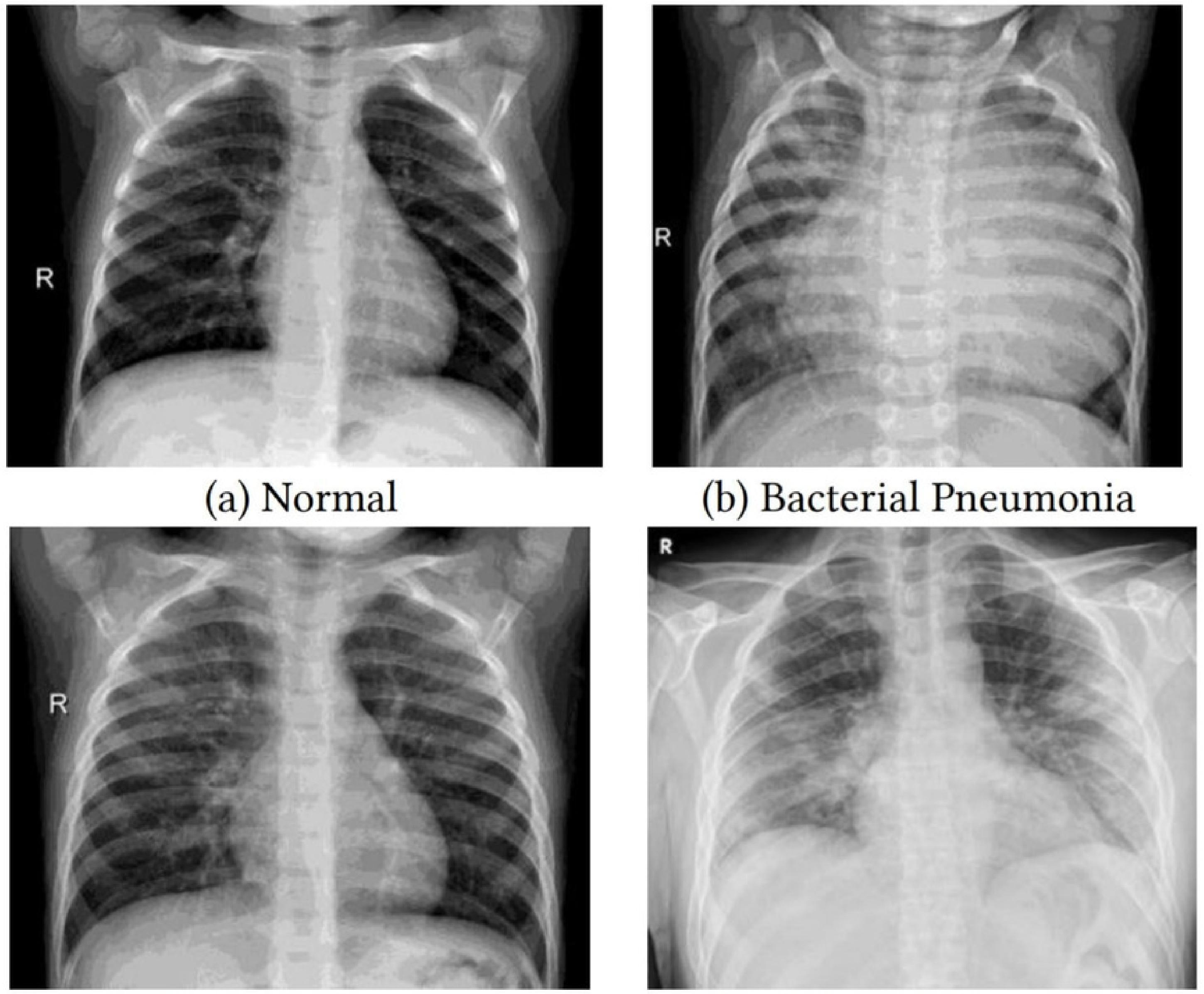
The test set consists of 30 images each of COVID, normal and viral pneumonia, while the rest of the images are used for training and validation purposes.

The validation set consists of 20% of the training set.

We Chose this dataset as it had viral pneumonia images as well. It has been observed that COVID and pneumonia have very similar symptoms and radiology results. It is an extremely difficult task even for a trained professional to distinguish between a COVID and viral pneumonia radiology result.

THE DATSET

Thus, a neural network or for that matter any algorithm trained just on COVID and normal radiology images will almost always give poor results (false positives) on viral pneumonia radiology scans and would thus become inadequate for use.



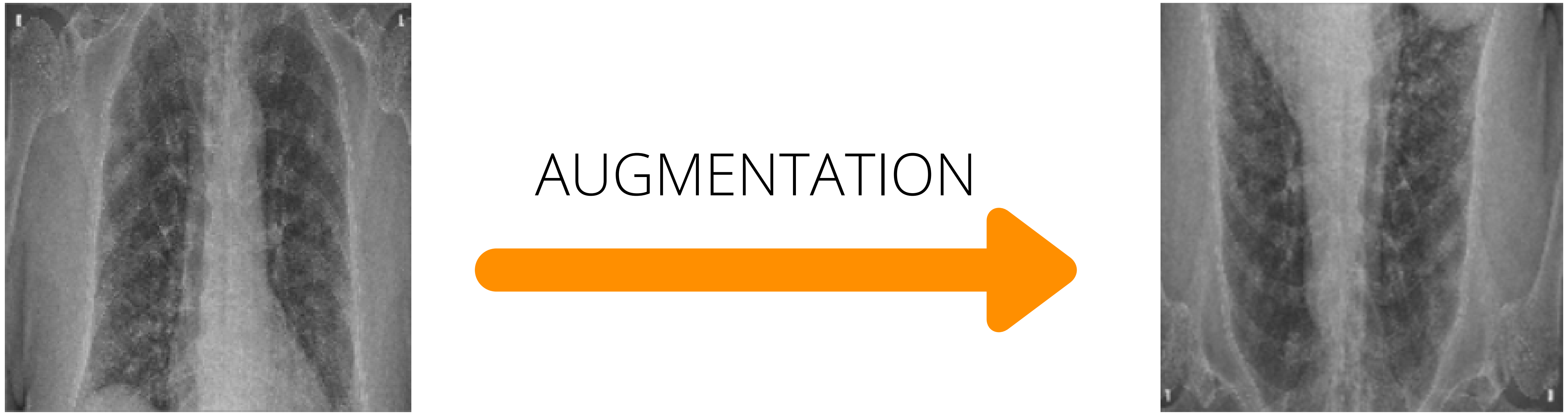
(a) Normal (b) Bacterial Pneumonia

(c) Viral Pneumonia (d) COVID-19 Pneumonia

DATA PREPROCESSING

- The dataset from Kaggle was in a form such that there were 3 directories each for COVID, viral pneumonia and normal. We wrote a script using the OS library to divide the data into training and test set which is in an acceptable format for Image Data Generators.
- The number of channels in the images were reduced from 3 to 1 using the OpenCV library. This was done based on the hypothesis that it would reduce the number of trainable parameters significantly without affecting the model results as the training images are radiology scans which are mostly (if not completely) in grayscale.

- After this, using the Image Data Generator, a lot of Image Augmentation was done like shifting, shearing, zooming, rotating and mirror flips. This enabled the data set to be more diverse and handle all sorts of images.
- We used 'Callbacks' for 'Early Stopping' while training the models. Callbacks enable us to stop the training when a particular threshold is met or is not met. The callback we used was 'EarlyStopping' which monitored the 'Validation Accuracy' and whenever it didn't exceed by at least 0.1% in every 10 epochs, the training would terminate automatically and the best weights would be restored.
- This not only saves us the trouble of chosing an appropriate number of epochs (in our case, we simply set it to 1000) but also prevents Over - fitting.



NETWORK ARCHITECTURE

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_8 (Conv2D)            (None, 178, 178, 64)      640
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 176, 176, 64)      36928
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 88, 88, 64)        0
_________________________________________________________________
dropout_6 (Dropout)          (None, 88, 88, 64)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 86, 86, 64)        36928
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 84, 84, 64)        36928
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 42, 42, 64)        0
_________________________________________________________________
dropout_7 (Dropout)          (None, 42, 42, 64)        0
_________________________________________________________________
flatten_2 (Flatten)          (None, 112896)            0
_________________________________________________________________
dense_6 (Dense)              (None, 64)                7225408
_________________________________________________________________
dense_7 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_8 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_8 (Dense)              (None, 3)                 99
=================================================================
Total params: 7,339,011
Trainable params: 7,339,011
Non-trainable params: 0
```

- The input to the model is of dimension - (none, 180, 180, 1).
- All the convolutional layers have 64 filters each of size (3, 3) with the activation function as 'relu'
- The size of pool for the pooling layers is 2*2.
- Dropout is introduced to reduce the chance of overfitting.
- There are 2 dense layers at the end with 64 and 32 nodes each followed by the output layer consisting of 3 nodes and a softmax activation function.

TRAINING THE MODEL

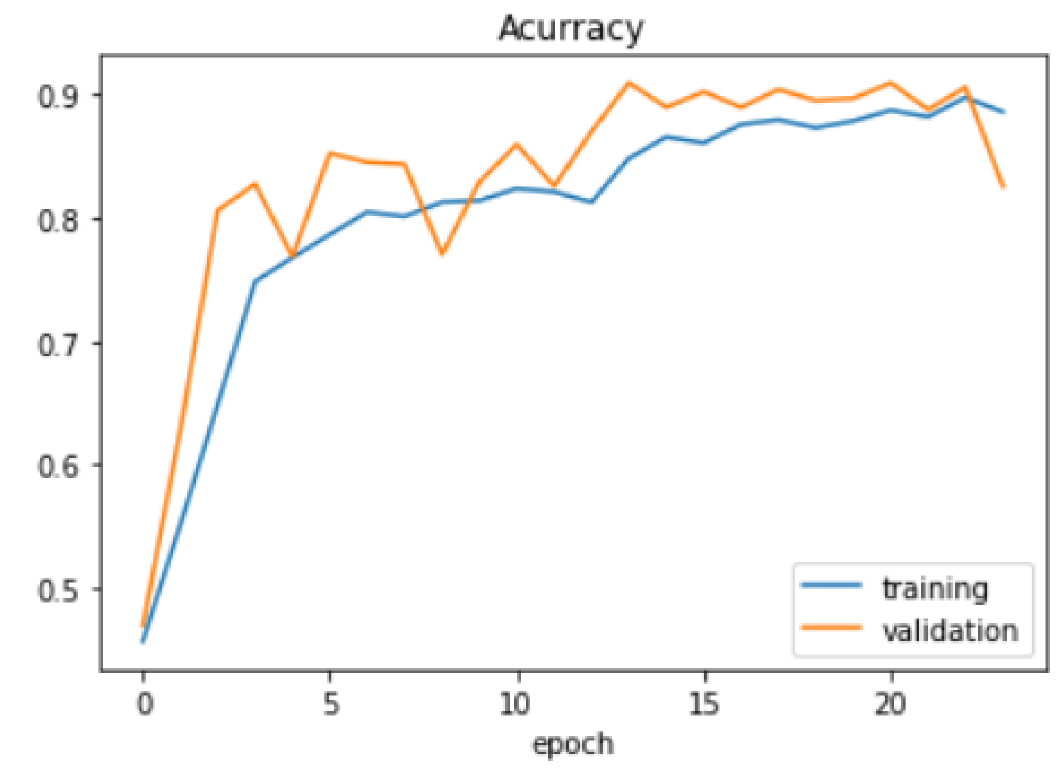
- The training was done with the help of data generators. The training set consisted of 80% of the training data and the validation set consisted of 20% of the data
- We used callbacks to save us the trouble of hard coding the number of epochs or use. A callback is an object that can perform actions at various stages of training (e.g. at the start or end of an epoch, before or after a single batch, etc.). Callbacks is a way to tune the "no. of epochs" hyper parameter.
- We used a custom Early Stopping Callback and set the 'val_accuracy' up to be monitored with a patience of 10 epochs and a min_delta value of 1%.

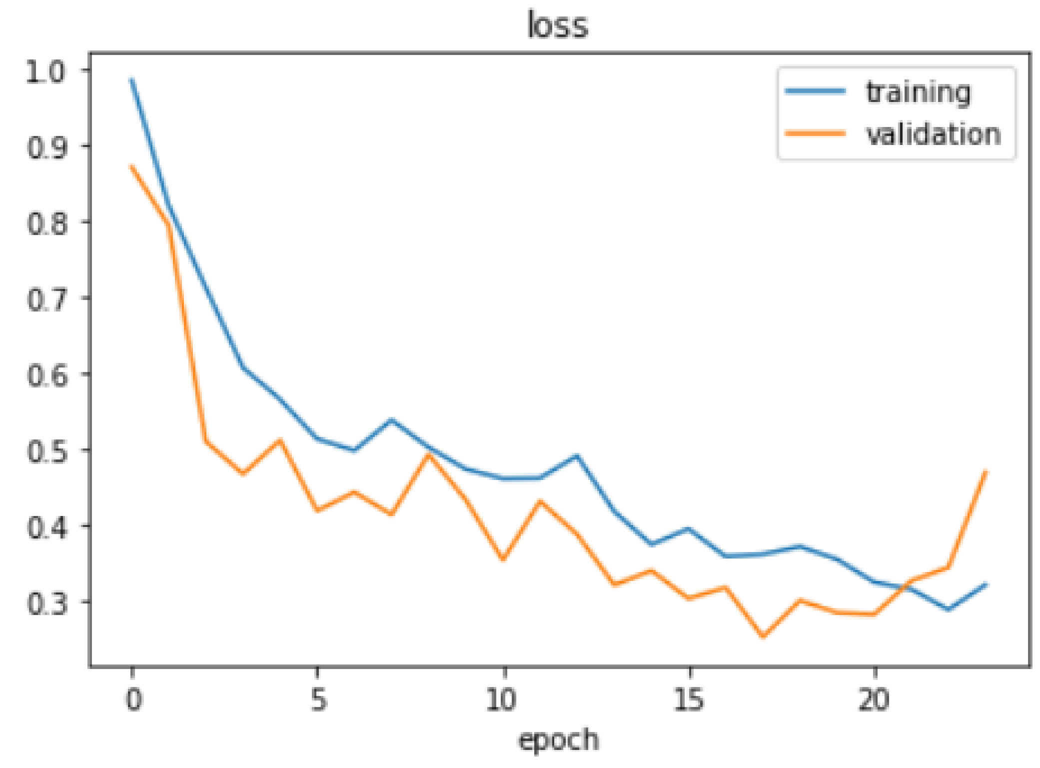
TRAINING THE MODEL

- We also set the restore_best_weights parameter to true. So we were able to set the number of epochs to 1000 and not worry about it.
- In simple words this means that the model would keep on training until every 10th epoch has a val_accuracy increase of at least 0.001. The moment this condition becomes false, the training would stop and the weights of the epoch with the best val_accuracy will be restored.
- The model trained for 25 epochs before the Early Stopping Callback stopped the training.
- The training accuracy achieved was - 88.73% and the validation accuracy achieved was 90.93%.

RESULTS AND ANALYSIS

- Holdout validation was used for testing as k-fold couldn't be used due to long training times.
- The model was tested on 90 images, 30 each of covid, viral and normal radiography scans.
- An interesting statistic was that there were 96% true positives for COVID and only 3% false positives.
- We also tested the model on the entire training set.
- The following were the accuracy and loss curves achieved while training.

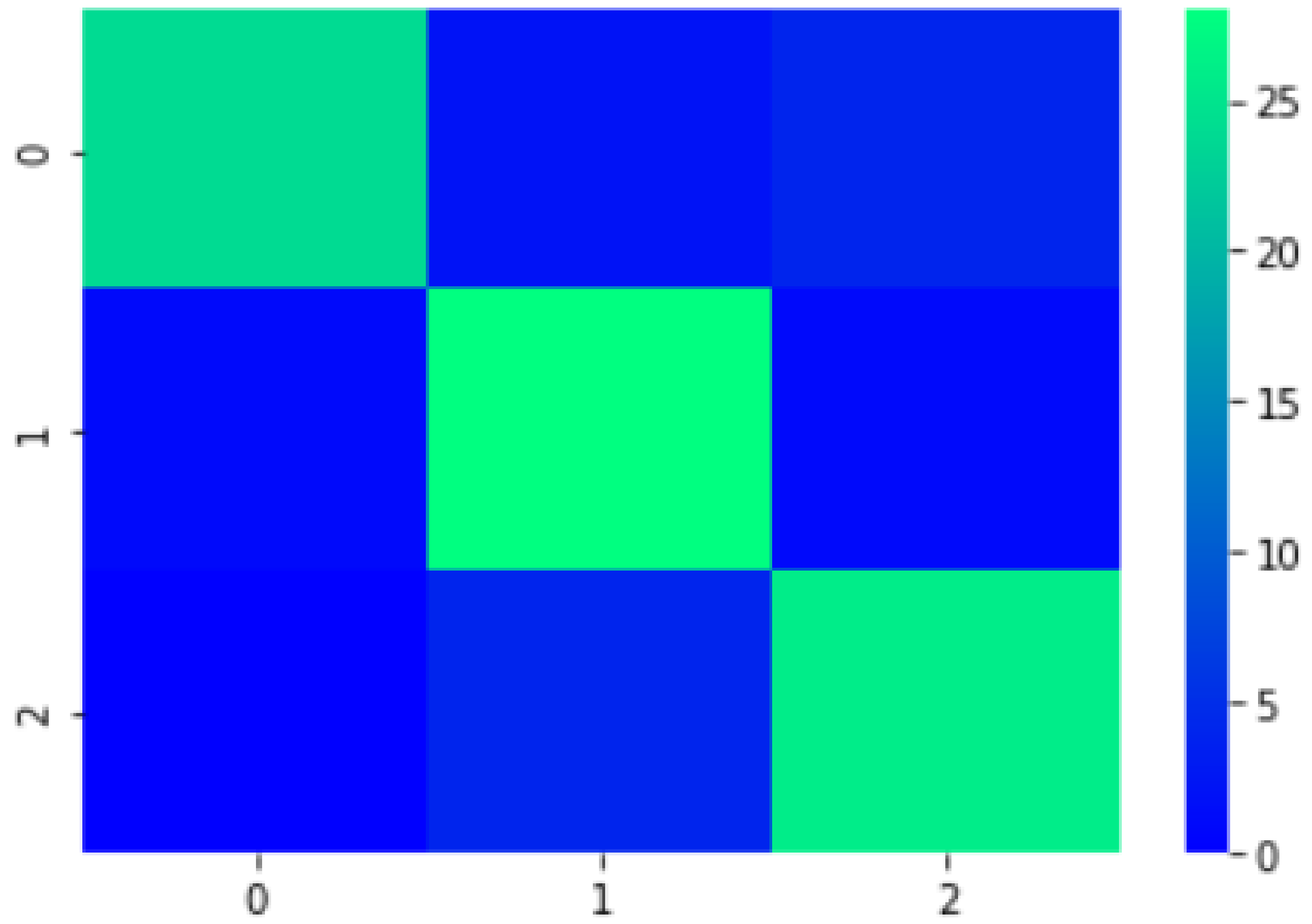




Test Set results Analysis

Confusion Matrix for the test set

```
[[24  2  4]
 [ 1 28  1]
 [ 0  4 26]]
```



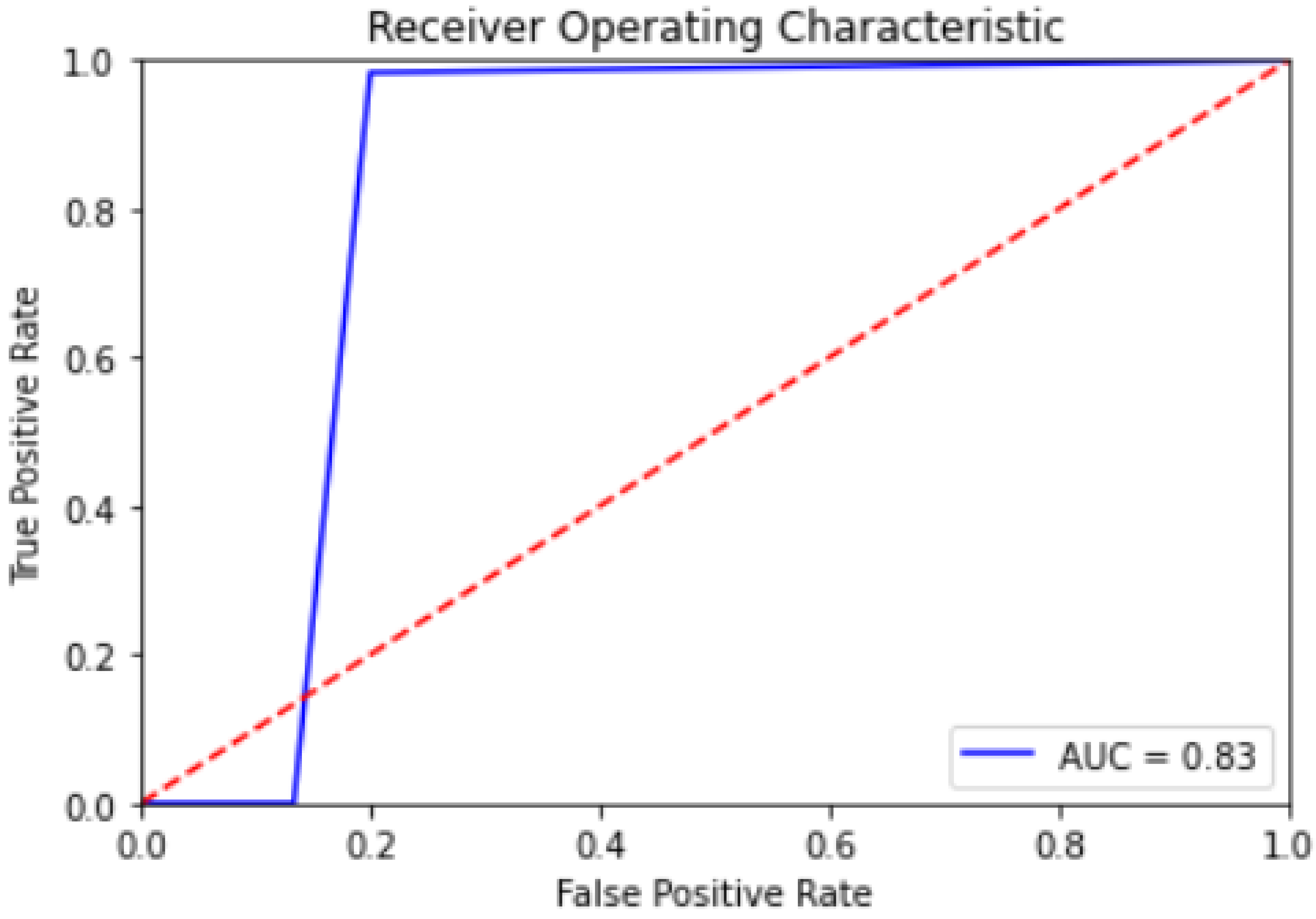
Test Set results Analysis

Classification report for the test set

	precision	recall	f1-score	support
0	0.96	0.80	0.87	30
1	0.82	0.93	0.87	30
2	0.84	0.87	0.85	30
accuracy			0.87	90
macro avg	0.87	0.87	0.87	90
weighted avg	0.87	0.87	0.87	90

Test Set results Analysis

AUC - SCORE for test set



Understanding the results on the test set

- The precision for the 3 classes are 96%, 82%and 84% respectively.
- The confusion matric has most of the elements around the diagonal showing that it is a good model.
- The recall for the three classes are 80%, 93% and 87% respectively.
- The accuracy of the model is 87%

Test Set results Analysis

Understanding the results on the test set

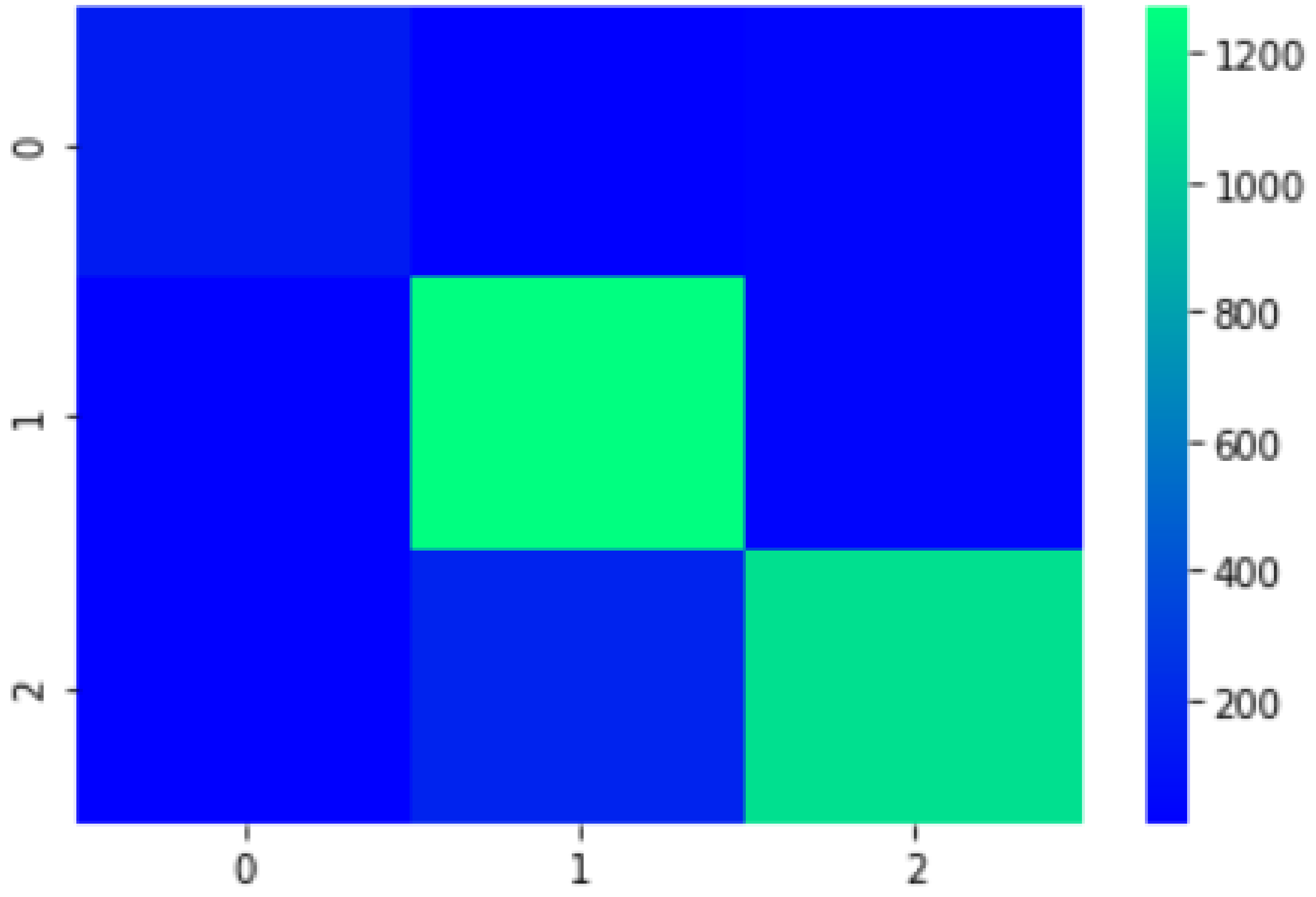
- The accuracy, precision and recall are often misleading so we calculate the f1 score and the Area Under the Curve (AUC score) to confirm our results
- The f1 score for the three classes are 87%, 87% and 85%.
- The AUC is 0.83 as well.
- All this demonstrates that the model is not over fitted and does not recognise analogous or bogus patterns as it given good results on the test set based on a number of metrics as stated above.

Test Set results Analysis

Training Set results Analysis

Confusion Matrix of the training set

```
[[ 146   13   30]
 [   9 1270   32]
 [  12  186 1117]]
```



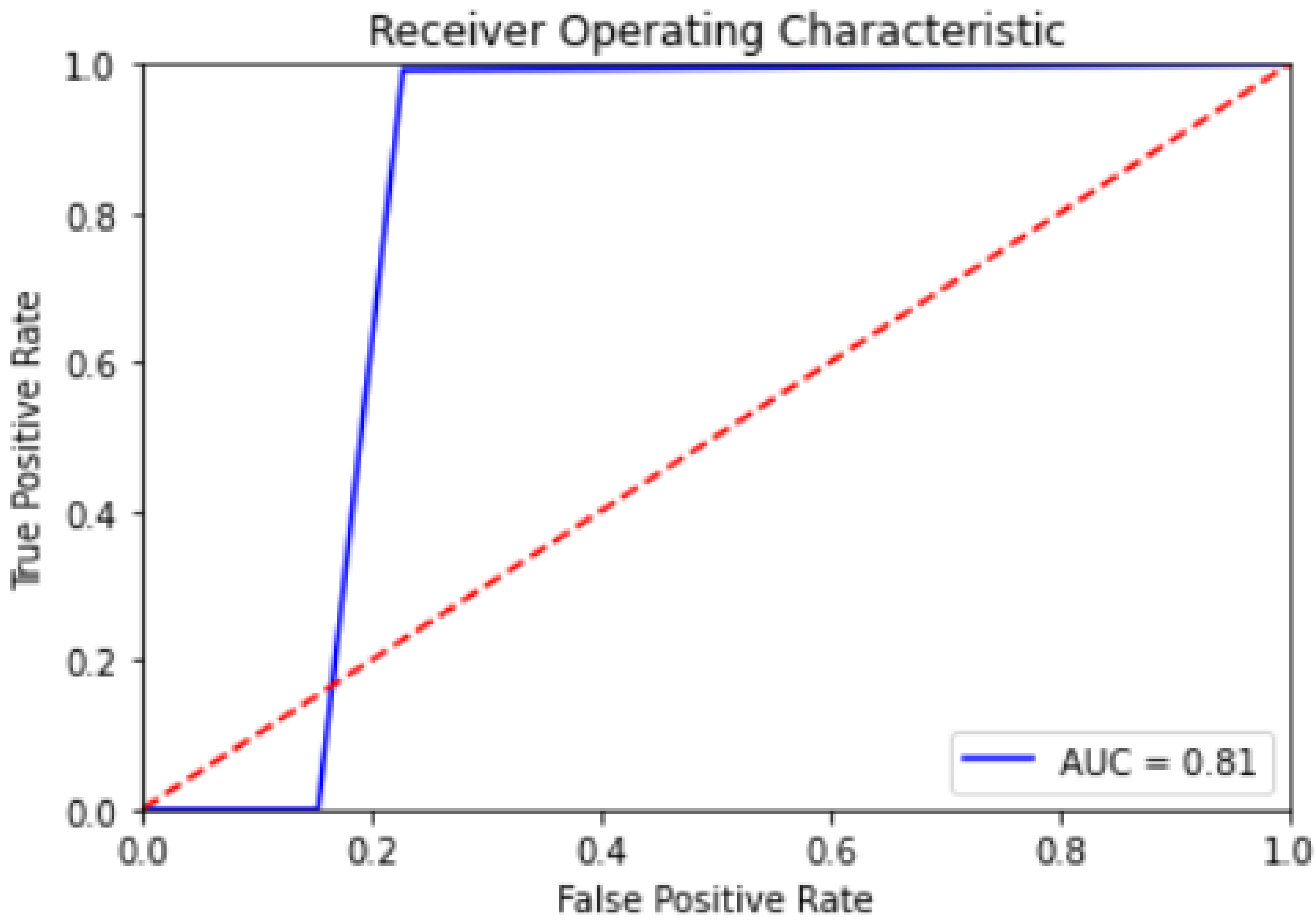
Training Set results Analysis

Classification report of the training set

	precision	recall	f1-score	support
0	0.87	0.77	0.82	189
1	0.86	0.97	0.91	1311
2	0.95	0.85	0.90	1315
accuracy			0.90	2815
macro avg	0.90	0.86	0.88	2815
weighted avg	0.90	0.90	0.90	2815

Training Set results Analysis

AUC - SCORE on the training set



Training Set results Analysis

Understanding the results on the training set

- The precision for the 3 classes are 87%, 86%, 95% respectively.
- The confusion matric has most of the elements around the diagonal showing that it is a good model.
- The recall for the three classes are 77%, 97%, 85% respectively.
- The accuracy of the model is 90% on the training set.

Understanding the results on the training set

- The accuracy, precision and recall are often misleading so we again calculate the f1 score and the Area Under the Curve (AUC score) to confirm our results
- The f1 score for the three classes are 82%, 91% and 90%.
- The AUC score for the training setis 0.81 as well.
- All this demonstrates that the model is not over fitted and does not recognise analogous or bogus patterns as it given good results based on a number of metrics as stated above.

__Training Set results Analysis__

Future Work

- Newer versions of the data set are now available on kaggle which have much more images. So this could be used to train a better model.
- We were able to keep the size of the model file relatively low as per our objective as we plan to deploy it on the web. However the actual deployment part still remains as a future work yet to be completed.
- Also we would like to develop a state of the art model as well leveraging transfer learning techniques for comparison with our current "lite-ready-to-be-deployed-easily" model.

REFERENCES

[1] Punia, R., Kumar, L., Mujahid, M., & Rohilla, R. (2020, June). Computer Vision and Radiology for COVID-19 Detection. In 2020 International Conference for Emerging Technology (INCET) (pp. 1-5). IEEE.

[2] Sakib, S., Tazrin, T., Fouda, M. M., Fadlullah, Z. M., & Guizani, M. (2020). DL-CRC: Deep Learning-Based Chest Radiograph Classification for COVID-19 Detection: A Novel Approach. IEEE Access, 8, 171575-171589.

[3] Sethi, R., Mehrotra, M., & Sethi, D. (2020, July). Deep Learning based Diagnosis Recommendation for COVID-19 using Chest X-Rays Images. In 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 1-4). IEEE.

[4] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R. K. Chakrabortty and M. Ryan, "A Hybrid COVID-19 Detection Model Using an Improved Marine Predators Algorithm and a Ranking-Based Diversity Reduction Strategy," in IEEE Access, vol. 8, pp. 79521-79540, 2020, doi: 10.1109/ACCESS.2020.2990893.

[5] S. Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio and S. K. Antani, "Iteratively Pruned Deep Learning Ensembles for COVID-19 Detection in Chest X-Rays," in IEEE Access, vol. 8, pp. 115041-115050, 2020, doi: 10.1109/ACCESS.2020.3003810.

[6] R. G. Babukarthik, V. A. K. Adiga, G. Sambasivam, D. Chandramohan and J. Amudhavel, "Prediction of COVID-19 Using Genetic Deep Learning Convolutional Neural Network (GDCNN)," in IEEE Access, vol. 8, pp. 177647-177666, 2020, doi: 10.1109/ACCESS.2020.3025164.

[7] M. J. Horry et al., "COVID-19 Detection Through Transfer Learning Using Multimodal Imaging Data," in IEEE Access, vol. 8, pp. 149808-149824, 2020, doi: 10.1109/ACCESS.2020.3016780