COVID-19 DETECTION ANDROID APP BASED ON CHEST X-Rays CT SCANS

NITIN BASANTWANI[1], ASHISH KUMAR[2], SUMIT GANGWAR[3], ASHISH OLKHA[4], PRINCE PRIJWAL[5]
GAURI MATHUR[6]

[1 2 3 4 5]Student, Department of Computer Science and Engineering, Lovely Professional University
Phagwara, Punjab 144411, India
[6]Assistance Professor, Department of Computer Science and Engineering, Lovely Professional University
Phagwara, Punjab 144411, India
Author : [1]nitinbasantwani14@gmail.com
Co-Authors : [2]iashishkumar3551@gmail.com@gmail.com, [3]sumitkurmi9057@gmail.com, [4]iashisholkha7@gmail.com, [5]prijwal14@gmail.com5, [6]gauri.mathur@lpu.co.in

Abstract. The Covid-19 android app based on Chest X-rays & CT scans is an integration of Machine learning and android. Our group developed an android application that uses a Deep learning model and predict if the user is suffering from Covid based on the chest X-rays & CT scans. The main tools used for the development of this project were android studio & Firebase for android app development, Jupyter notebook & google Colab as a code editor along with google drive to fetch the data. The main Libraries used for training ML models were Tensorflow, Keras and the method used was Transfer Learning using the pre-trained model InceptionV3. In coping and fighting against COVID-19, the most critical step is to effectively screen and diagnose infected patients. RT-PCR is considered to be a reliable test for detection of coronavirus, but the problem with RT-PCR test and Elisa tests is that they take a lot of time to generate results, Since Covid is highly contagious and spreads through Human-to-Human Interactions it's Crucial to detect it as early as possible to stop the transmission. Corona is an Infectious disease that affects the Lungs similar to Pneumonia, Deep Learning and Machine Learning models have produced significant results in the past for pneumonia detection in Lungs for this research project we tried to implement the same approach.

Keywords: Deep Learning, Transfer Learning, Machine Learning, TensorFlow, InceptionV3, Supervised Learning, Android Studio, Java, Python

(Received May 01st, 2021 / Accepted May 31st,2021)

1 Introduction

It's been over a year now since the world has been set aback by the Global pandemic called Coronavirus or Covid19, it disrupted life at every level, devastated & destroyed thousands of lives worldwide creating an imbalance economically it led to global crisis. The very first case of the disease formerly referred to as 2019 Novel coronavirus or 2019-nCov was reported in Wuhan, China in the end of year 2019. It spread like a wildfire and engulfed more than 200 countries worldwide with a death toll of 300794, 140322903 confirmed cases till date and the scale are still rising. Covid was declared as a Global Pandemic by The World Health Organization (WHO) on 11th March 2020. It's a highly infectious disease caused by SARS-CoV-2, a virus that belongs to the large family of coronaviruses. The main symptoms associated with the virus fever, cough, sore throat, shortness of breath etc. Main mode of transmission being air or by inhaling the respiratory droplets of an infected person via human-to human interaction. The virus has spread exponentially at an alarming rate making it a threat to humanity. Since it affects the respiratory tract, oral or nasopharyngeal swabs are being used to detect the presence of covid virus via Reverse Transcription Polymerase Chain Reaction (RT-PCR) and Antigen test. RT-PCR being the main and the most effective test yet unreliable as it suffers from a high false-negative rate and another downside being that RT-PCR test is a tedious and a time-consuming process furthermore it takes more than 24 hours to get its results. Due it's highly contagious nature it's crucial to detect and control the virus from spreading as soon as possible. Some of the common practices that are being used to avoid the virus are practicing social distancing and wearing masks as it proves to be effective in preventing the acquisition of covid-19 virus. Since the RT-PCR tests consume a lot of time to generate results. Radiological imaging such as Computed Tomography (CT scans) & X-rays which are used to diagnose a lot of respiratory disorders and lung abnormalities can be used to detect & diagnose the presence of Covid-19 in the lungs too.

Since it's vital to identify the presence of the Covid virus in the body to help curb down the further spread of the virus, Though RT-PCR is the most reliable method itâs still a sensitive, Time consuming and complex method. So the main objective of our project is to build up an android app that can helps a user to predict the presence of covid virus in the Chest X-rays and CT scans. Since X-rays arenât costly and their apparatus is available at almost all the health centre and does not require any complex method such as sample collection unlike in other Covid-19 related test and is quite cheap as compared to other testing alternatives. Radiologists have been using chest X-rays to diagnose respiratory disorders and abnormalities, so after the emergence of the corona, Chest X-rays and CT scans Datasets which are carefully labelled by the team of experts are available on the internet, we have used one such dataset which was available on Kaggle. our main objective was to develop an android application that can use a deep learning model and can return a good accuracy [1, 7, 4, 12, 14, 6, 10, 9, 8].

2 LITERATURE REVIEW

Deep learning and image processing in biomedical image analysis and processing has produced distinct results especially in the field of chest radiology. These methods are often used to diagnose pulmonary tuberculosis[5] and to diagnose lung nodule[3]. There are a variety of features such as CNN's, feature extraction and ensemble learning. A brief review of some of the important contributions from the existing publications is highlighted below.

- Stephen et al provided Deep learning to detect and classify pneumonia in chest X-ray images[11] . The presented ConvNet model extracted features from images and utilized those features to detect pneumonia. The model achieved an accuracy of 93.73

- Varshney et al. used pre-trained models like Xception, ResNet50, VGG-16, DenseNet121, DenseNet-121, and DenseNet169 for feature extraction and used algorithms like SVM, K-nearest neighbours, NaÃ¯ve Bayes and Random Forest for classifying pneumonia using X-ray images[13]. Wang et al. applied hyperparameter optimization to further improve the accuracy.

- Wang et al. suggested a DL model draw out visual characteristics from CT Scan images categorizing Covid19[15]. The Dataset compromised of 1065 CT Scans, out of which 740 images were of viral pneumonia and 325 images were of patients suffering from COVID-19. The model attained an accuracy of 79.3

Most of the past approaches uses transfer learning concept along with Data augmentation technique, fine-tuning and ensemble learning for the classification of pneumonia. These techniques have produced outstanding results with the pneumonia datasets and the same method can be used for identification of coronavirus using chest radiography and X-ray images.

In our approach, we used balanced Datasets with $13,685$ CT Scans and 9544 Chest X-rays Images. The Deep Learning models are always hungry for Data and the more Data we provide to a system, the better is the accuracy we can expect a system to return. There arenât any mobile applications for the system so we tried developing an user-friendly system can be implemented. We have used TensorFlow lite, firebase and android studio for the integration of the machine learning model into the android application.

3 Methodology

In our study, transfer learning is employed using the incepetionV3 pretrained architecture. The CNN model accomplishes outstanding execution on Chest X-rays and CT scans images dataset related to chest-related infections. The Deep Learning model is trained on 13000 CT scans and 9000 X-ray images and the accuracy attained is 93%, with a minimal false positive

3.1 Image Pre-Processing

The balanced data set of every Chest X-rays and CT scan were transferred into the pre-processing, augmentation and training. The pre-processing consist of several stages. The real image of X-rays and CT scan covid were compressed. Then the images were regularized during pre-processing.

3.2 Data pre-processing

The data was loaded into the model using the ImageDataGenerator Library from TF.preprocesing.

- The images were resized to 256 * 256 pixels and were divided into batches of 16. The entire dataset of 13000 images was divided into a ratio of 70:30 for training and Validation.

- The training dataset consisted of 9787 images which was used for training the Validation dataset consisted of 4193 images which was used for validation.

3.3 Data augmentation

Data Augmentation is a technique used for generating additional data using the methods like, horizontal flips, vertical flips, zooming, tilting, etc. The main purpose of Data Augmentation is to increase the quality of the data and gain more data for better model training. In our model the main Augmentation techniques we used were, Rescaling the images, Rotation range was used to rotated the images by 20Â° on both sides right and left, width shift range was used to shift the images by 20% and the same for height. We used Horizontal flip vertical flip to deal with the case where a user might upload an image that is upside down and flipped horizontally

3.4 Model Building

After the data was loaded and augmented, the model was ready to be defined and built. We used a pre trained model inceptionV3. Since we only need the features extracted by the model from lower layers, we didnât include the top layers by setting the include_top = False. Since the images are resized to 256 * 256 we set the input for inceptionV3 to be the same so that the pre-trained model is in sync with our input images. On top of the pre-trained model, we have set 2 more hidden layers the final layer was used for output and only had 2 hidden units.

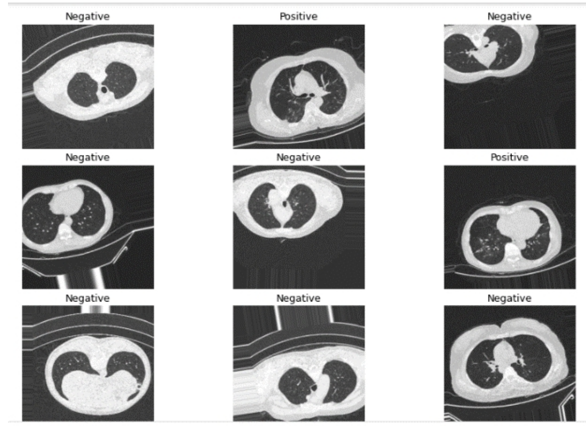


Figure 1: Shows the visualization of loaded images after data augmentation, Data Augmentation has generated the results as expected and now the model can be trained on almost all the possible types of images a user might upload.

3.4.1 Model compiling

- We chose Categorical-Crossentroy to be our loss since we have categorical data with 2 main classes i.e. Covid Non-Covid.

- We used Adam optimizer which is a combination of Classic Gradient Descent along with the momentum for better and fast convergence.

- The metrics used to evaluation was accuracy to return the accuracy of our model on each epoch.

3.4.2 Transfer Learning and training

Transfer learning is a process of using the knowledge learnt from one model or the features extracted from one model and implement it on another model. This process helps a model to quickly converge and reduce the training time and sometimes can even improve the performance. The weights of the hidden units in the model are updated after every Iteration. There are 314 layers and 25 million attainable parameters in InceptionV3 model and was trained on the ImageNet dataset. The precision is shown by validation data on a randomly selected group of images. If the training accuracy is higher than the validation accuracy that means the network is overfitting the training data and this arises a problem of bias. We trained the model for 50 Epochs, with a batch size of 16. Our model was doing quite good on the training data and hence we didnât have a bias problem, as well as the validation accuracy was almost 94% for the validation data.

3.4.3 Inception Model

InceptionV3 is a convolutional model that comprises 48 layers. The model is trained on more than a million images from the ImageNet repository[2]. The pre-trained model can classify images into 1000 categories. Figure 2 shows InceptionV3 is an acknowledged image recognition model that has been conveyed to achieved greater than 78.1% accuracy on the ImageNet dataset.

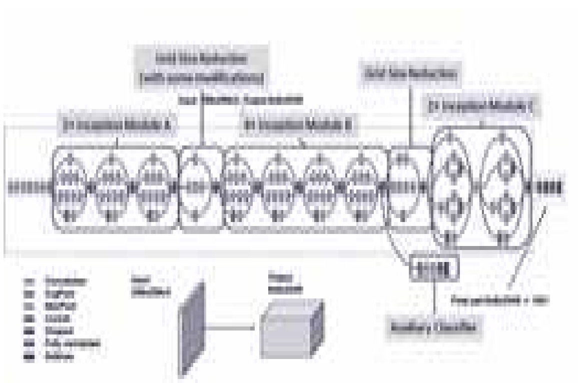


Figure 2: InceptionV3 model

Many researchers have developed this model over the years which is the pinnacle of many ideas. The categorization architecture of the inceptionV3 is shown in figure 3. As mentioned earlier, over a million images from 1000 categories have been trained in this model on some very robust machines which can preserve the expertise that the model has grasp during its original training and apply it to the small-scale dataset in the final layer. As a result of this work, computational power and an exceptionally precise classification will be acquired without the need for extensive training.

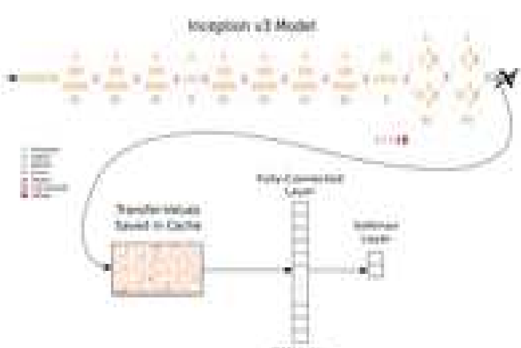


Figure 3: InceptionV3 model

3.5 Performance Measures and confusion matrix

To check our modelâs performance, we used the F-score which is calculated as the mean score of both precision and recall. On evaluating the final model, the accuracy returned is 94.4%. The Confusion matrix in the Figure 4shows us that we have a very low False-negative Rate. In medical diagnosis, we always aim to get a low False Negative rate. The Final report for the validation data-set returns an accuracy of 94%.

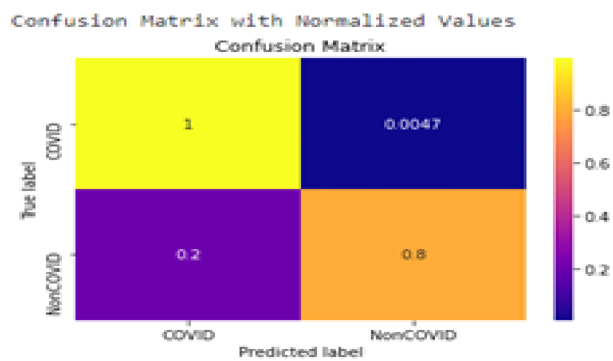


Figure 4: Confusion matrix with normalised values

4 Android application integration

4.1 Android studio

The Android app was developed on Android Studio which is an Integrated Development Environment used for developing android apps. Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. Android supports Kotlin and Java Programming language.

4.2 Login/sign Up

Firebase: It is a NoSQL database based on the cloud. It is generic where user can modify and arrange things as per their need. Firebase serves real-time database, authentication and many more tools which help developers to build high-quality applications. Firebase tools we have used in our app:

- Authentication
- Realtime Database

Here, in our project, we have used its authentication feature for logging in into the app and real-time database to store the data of the user. Firebase Auth is a service that can authenticate users using only client-side code. It includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

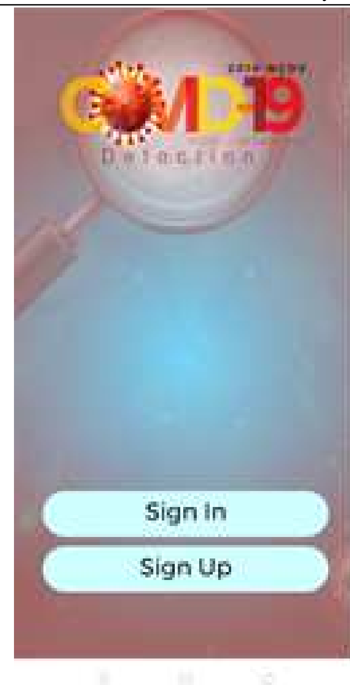


Figure 5: Android app Main Sign In/Sign up Page

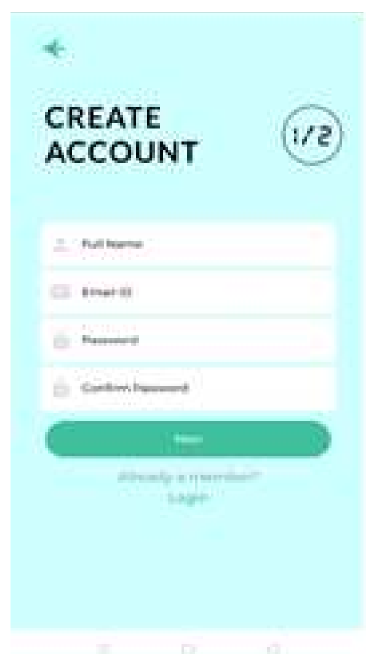


Figure 6: Sign Up Page 1

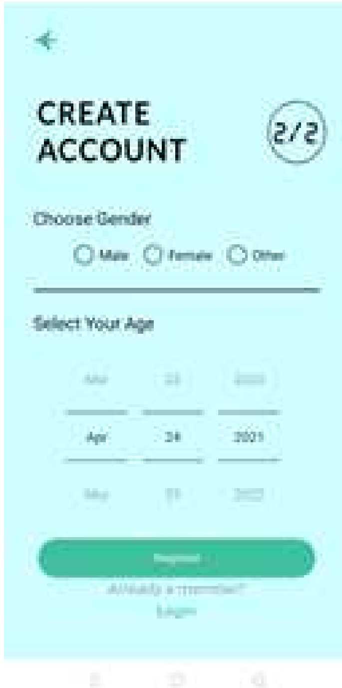


Figure 7: Sign Up page 2

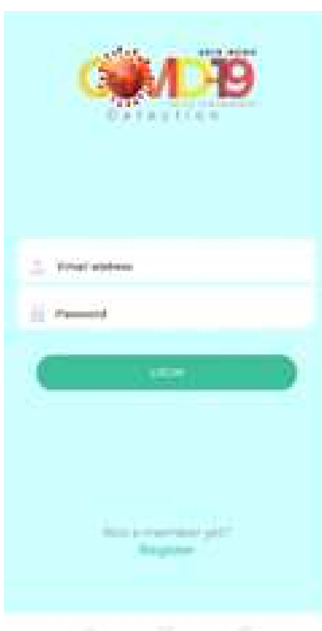


Figure 8: Apps Main Sign In page

4.3 Covid Case Live Tracker

In our App, we have also added the feature of tracking coronavirus which help people to track infections more accurately and effectively in a country. Users can also keep live track of the number of people are infected, recovered and also the number of deaths in a country.

Figure 9: Live Tracking interface

4.4 Connecting to Machine Learning Model

The model that we built could only be loaded by a PC or a virtual machine with TensorFlow installed, but since our main aim was to build an android app we had to convert the final model into a TFlite model which could be used in making the android model. Using TensorFlow mobile lite library which is supported by Android Studio. Figure 11 shows the relationship between machine learning and android application. It shows how the CT Scan and X-rays images are uploaded and tested by machine model and output the result in the application.

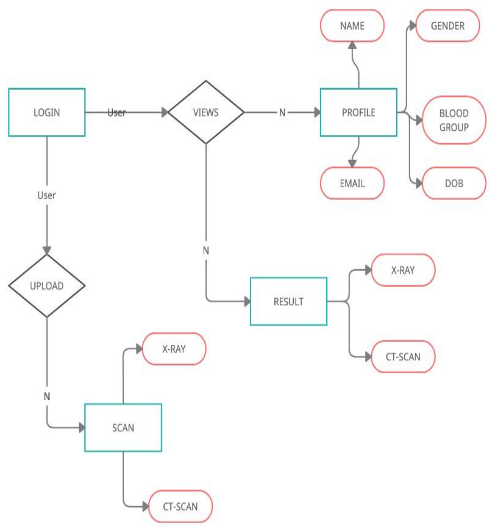
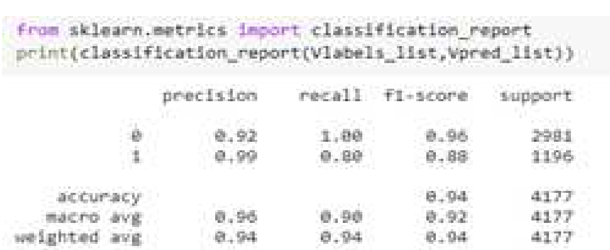


Figure 10: ER Diagram of the Entire Application

5 RESULT

The performance of the models was examined after Data Augmentation and it increased the accuracy of our model by 3 â 4%. We have also calculated the F1 score, precision and accuracy. Figure 10 shows the calculated value of f1-score, precision and accuracy. On evaluating the final model, the accuracy is returned is 94.4%. The False-negative Rate means the model predicted the person doesn't have the disease but is suffering from the disease. The Confusion matrix shows that the model has a very low False-negative rate. In medical diagnosis, we always aim to get the False Negative rate to be as low as possible. The Final report for the validation dataset returns an accuracy of 94%. We have also integrated the android application with the machine learning model, a user can use this application to see if they have covid or not using Either CT-Scans or X-rays it can also be used to track live covid case across the country.



```
from sklearn.metrics import classification_report
print(classification_report(Vlabels_list,Vpred_list))

              precision    recall  f1-score   support

           0       0.92      1.00      0.96      2981
           1       0.99      0.80      0.88      1196

    accuracy                           0.94      4177
   macro avg       0.96      0.90      0.92      4177
weighted avg       0.94      0.94      0.94      4177
```

Figure 11: Classification Report showing accuracy, Precision Recall

Table 1: Confusion Matrix

		Predicted	
		Covid	Non-Covid
Actual	Covid	2967 (True Positive)	14 (False Negative)
	Non-Covid	242 (False Positive)	952 (True Negative)

Table 2: Classification Table

	Precision	Recall	F1 Score	Samples
Non-Covid (0)	92%	100%	96%	2981
Covid (1)	99%	80%	88%	1196

The data we had was pretty imbalanced, the Negative samples were far greater than the positive samples. We could have deleted the negative samples to balance the dataset and it might have also increased the accuracy but we decided not to do so, as we wanted to mimic the real-world scenario where in a random sample, on an average the number of positive samples would be less than the negative ones, the ratio we followed were 3:1 which means out of every 4-person tested; 1 person will be positive while the rest of the patients will be negative.

5.1 Precision

The precision which depicts the number of actual positive samples out of the total samples that the model predicted as positive and is usually calculated as

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{1}$$

Therefore, the Precision for our model is

$$Precision = \frac{2967}{2967 + 242} = 0.92 \tag{2}$$

Which means 92%

5.2 Recall

The Recall which means out of all the actual positive cases how many samples did the model labelled correctly. It is given as

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

Therefore, the Recall for our model is

$$Recall = \frac{2967}{2967 + 14} = 0.995 \quad (4)$$

Which means 99.5%

5.3 Accuracy

The accuracy of the model is simply the ratio between the correctly predicted samples divided by the total number of samples. It is calculated as

$$Accuracy = \frac{True\ Positive + True\ Negative}{TP + TN + FP + FN} \quad (5)$$

Where TP = True positive, TN = True Negative, FP = False Positive, FN = False Negative
Therefore, the Accuracy for our model is

$$Accuracy = \frac{2967 + 954}{2967 + 954 + 14 + 242} = 0.938 \quad (6)$$

Which means 93.8%

5.4 F1 Score

F1 Scores of a model is the weighted average of both precision Recall, Unlike the accuracy it not only takes the True Positive True Negative but it also takes False Positive False negative into the consideration to return the final accuracy and thatâs is usually a better estimate of accuracy, especially when the data is imbalanced. It is given as

$$F1Score = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (7)$$

Therefore, the F1 Score for our model is

$$F1Score = 2 * \frac{0.92x0.99}{0.92 + 0.99} = 0.95 \quad (8)$$

Which means 95%

5.5 Macro Average

Macro average is the average of f1 scores of different classes or sometimes is also calculated as the average of unweighted mean per label.

$$MacroAverage = \frac{F11 + F12... + F1n}{NumberofClasses(N)} \quad (9)$$

$$MacroAverage = \frac{0.96 + 0.88}{2} = 0.92 \quad (10)$$

Which means 92%

5.6 Weighted Average

Weighted Average is the weighted mean of F1 score of different classes, it is calculated as,

$$WeightedAvg = \frac{\text{F11 * W1+F12*W2...+ F1n*Wn}}{Total\ Weight} \quad (11)$$

where W is Weight

So the weighted average for our model is

$$WeightedAverage = \frac{0.96 * 2981 + 0.88 * 1196}{2981 + 1196} = 0.937 \quad (12)$$

i.e 93.7%

6 CONCLUSION

So even though we had an unbalanced dataset, the accuracy and F1 score that our model returned is pretty good. The final accuracy F1 score we achieved on the test dataset were 94% 95%. The Android application was tested on the real dataset, i.e. CT scan of a actual Covid patient dated 26th April 2021 and the android application returned Covid which concludes that the final application is functioning as Intended.

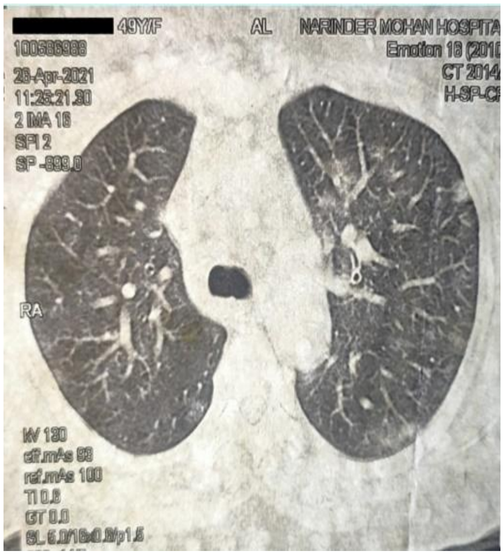


Figure 12: Chest CT Scan of a Covid Positive patient Dated 26th April 2021

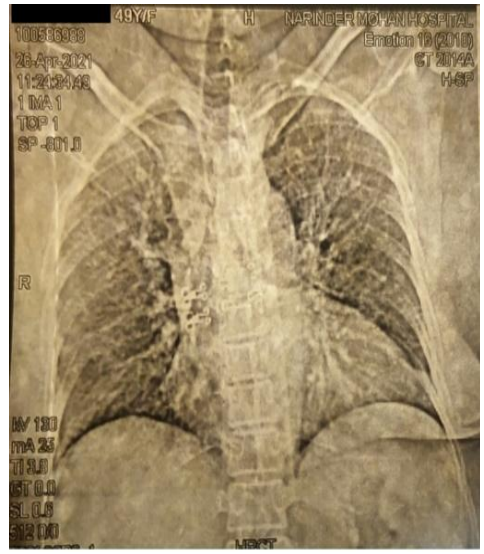


Figure 14: Chest X-ray of a Covid Positive patient Dated 26th April 2021

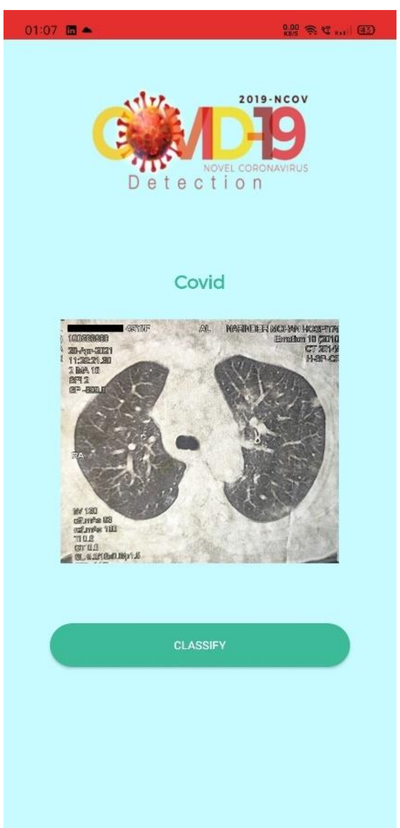


Figure 13: Prediction returned by the App For the CT Scan

7 FUTURE SCOPE

Since Deep Learning model craves for Data, the more data we provide to a model, the better is the accuracy we can hope a model to return unless and until we overfit the data. we hope to increase the accuracy of our model in future by increasing the data. Currently the App that we built is only supported for an Android Device but in future we hope to develop an IOS app too. The Live Tracking feature is currently using the live data from a single source and there might be a bit of lag as the Data is updated periodically, we hope to integrate the data from different sources so that the app can return the most recent updates. We also hope to include a section where users can help us out by uploading their own labelled CT scans X-Ray images to improve our dataset and add value to the quality of dataset.

References

[1] de Almeida, F. L., Rosa, R. L., and Rodríguez, D. Z. Voice quality assessment in communication services using deep learning. In *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, pages 1–6. IEEE, 2018.

[2] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

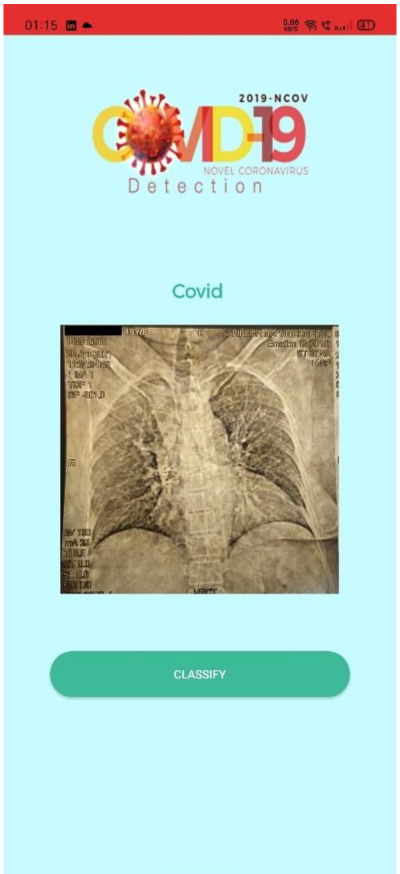


Figure 15: Prediction returned by the App For the X-ray

[3] Huang, P., Park, S., Yan, R., Lee, J., Chu, L. C., Lin, C. T., Hussien, A., Rathmell, J., Thomas, B., Chen, C., et al. Added value of computer-aided ct image features for early lung cancer diagnosis with small pulmonary nodules: a matched case-control study. *Radiology*, 286(1):286–295, 2018.

[4] Jordane da Silva, M., Carrillo Melgarejo, D., Lopes Rosa, R., and Zegarra Rodríguez, D. Speech quality classifier model based on dbn that considers atmospheric phenomena. *Journal of Communications Software and Systems*, 16(1):75–84, 2020.

[5] Lakhani, P. and Sundaram, B. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017.

[6] Nunes, R. D., Rosa, R. L., and Rodríguez, D. Z. Performance improvement of a non-intrusive voice quality metric in lossy networks. *IET Communications*, 13(20):3401–3408, 2019.

[7] Rodríguez, D. Z., Carrillo, D., Ramírez, M. A., Nardelli, P. H. J., and Möller, S. Incorporating wireless communication parameters into the e-model algorithm. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:956–968, 2021.

[8] Rodríguez, D. Z., da Silva, M. J., Silva, F. J. M., and Junior, L. C. B. Assessment of transmitted speech signal degradations in rician and rayleigh channel models. *INFOCOMP Journal of Computer Science*, 17(2):23–31, 2018.

[9] Rodríguez, D. Z. and Junior, L. C. B. Determining a non-intrusive voice quality model using machine learning and signal analysis in time. *INFOCOMP Journal of Computer Science*, 18(2), 2019.

[10] Rodríguez, D. Z., Rosa, R. L., Almeida, F. L., Mittag, G., and Möller, S. Speech quality assessment in wireless communications with mimo systems using a parametric model. *IEEE Access*, 7:35719–35730, 2019.

[11] Stephen, O., Sain, M., Maduh, U. J., and Jeong, D.-U. An efficient deep learning approach to pneumonia classification in healthcare. *Journal of healthcare engineering*, 2019, 2019.

[12] Terra Vieira, S., Lopes Rosa, R., Zegarra Rodríguez, D., Arjona Ramírez, M., Saadi, M., and Wuttisittikulkij, L. Q-meter: Quality monitoring system for telecommunication services based on sentiment analysis using deep learning. *Sensors*, 21(5):1880, 2021.

[13] Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R., and Mittal, A. Pneumonia detection using cnn based feature extraction. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–7. IEEE, 2019.

[14] Vieira, S. T., Rosa, R. L., and Rodríguez, D. Z. A speech quality classifier based on tree-cnn algorithm that considers network degradations. *Journal of Communications Software and Systems*, 16(2):180–187, 2020.

[15] Xu, B. and Meng, X. A deep learning algorithm using ct images to screen for corona virus disease (covid-19). *Preprint*.