Early Detection of COVID-19 Using Machine Learning

Tismeet Singh[1] *and Kartikeya Agarwal*[2]

Abstract

The COVID-19 Pandemic had a devastating impact both on social and economic fronts for a majority of the countries around the world. It spread at an exponential rate and affected millions of people across the globe. The aim of this study was to improve upon a lot of existing studies on COVID detection using Machine Learning. While Machine Learning methods have been widely used in other medical domains, there is now considerable demand for ML-guided diagnostic systems for screening, tracking, analysing, and predicting the spread of COVID-19 and finding a concrete and viable cure for it. We employed the power of Transfer Learning guided Convolutional Networks to predict the existence of the COVID-19 virus in the lung X-Ray of any subject. Deep Learning, one of the most lucrative and potent techniques of machine learning becomes the modern saviour when such crises arise. With the power of this technique, we studied a plethora of models, selected the best ones and then trained them to produce the most optimal results. We used multiple pretrained models and improved upon them by adding structured Dense and Batch Normalisation layers with appropriately selecting activation functions. Elaborate testing yielded a maximum accuracy of over 99%.

Keywords : Computer Vision, Confusion Matrix, Convolutional Neural Network, COVID-19, Deep Learning, Machine Learning, Transfer Learning, X-Ray

I. INTRODUCTION

The Novel Coronavirus originated in the Wuhan District of China in late December 2019 [1]. The virus disrupted the lives of people all across the world. The quota of medical professionals and equipment needed to combat the plethora of cases arising due to the COVID-19 virus became insufficient. Considering this scenario, the problem of fighting this virus far outweighed the resources at hand and it was therefore, crucial to use AI-driven techniques for early detection of the virus and hence, early treatment and remedy. The virus targets the lungs and the pulmonary system.

At present, RT-PCR (Real-time Polymerase Chain Reaction) and antigen tests are the major contributors in the detection of this virus. Alas, they are time consuming and can sometimes be plagued by false diagnoses. The evolution of AI driven tools and equipment has proved to be prolific in the field of biology and medicine. CT scans and X-Ray imaging can act as critical players in the early detection of COVID-19 virus as they can act as the source data for neural networks. This new world of computer vision can hence prove to be of great significance in identifying the presence of the virus in any human system.

With that in mind, we decided to tackle this medical emergency using Machine Learning, a technique that is a subset of automated data analysis using supervised or unsupervised algorithms and datasets. It is a part of Artificial Intelligence that dictates that machines can

Manuscript Received : January 2, 2022 ; Revised : January 18, 2022 ; Accepted : January 20, 2022. Date of Publication : February 5, 2022.

T. Singh[1] is a *Student* of Computer Science, Department of Computer Science and Engineering, Netaji Subhas University of Technology, Dwarka Sector - 3, Delhi - 110 078. Email : tismeet14sh@gmail.com
ORCID iD : https://orcid.org/0000-0002-0829-3261
K. Agarwal[2] is a *Student* of Computer Science, Department of Computer Science and Engineering, Netaji Subhas University of Technology, Dwarka Sector - 3, Dwarka, Delhi -110 078. Email : kartikeya72001@gmail.com
ORCID iD : https://orcid.org/0000-0002-4168-2533

DOI : https://doi.org/10.17010/ijcs/2022/v7/i1/168953

record parameters, identify patterns and understand the sensitivity in text as well as image dataset. In this study, we aimed at some specific algorithms to detect COVID-19 using chest X-Ray images as data. Publicly distributed sources and datasets were used in this study to gain insights on how machines can detect COVID-19 cases and how the disease is different from other lung related diseases and other viral Pneumonias.

We studied a dataset of over six thousand (6432, excluding testing dataset) images which included lung x-rays of Normal, COVID as well as other viral Pneumonia infected individuals. Out of these 6432 images over 4000 were from other Viral Pneumonia and about 576 formed the COVID-19 dataset and the rest formed normal images. These images were first cleaned and then an appropriate amount of image augmentation was worked up. We used different machine learning models to create a comparative analysis. From all the available techniques, we tested the dataset using multiple mathematical models, ranging from Support Vector Machines to Random Forests to Deep Convolutional Networks, but in the end, we decided that the results obtained using Deep Convolutional models were far superior and more robust than all the other mathematical systems. The highest accuracy achieved was using Deep Layers paired with Feature Selection layers provided by the VGG16 Transfer Learning model. The model output was a peak accuracy of over 99.00196% after training multiple structural and dense layers for many hundreds and thousands of steps per epoch. The data we provide is just an aid to the ongoing crisis and is in no way projected to replace any crucial decisions made by a physician.

A. Related Works

To validate and cross verify the research, we consulted a lot of papers and studied them carefully to improve upon them. The author in paper CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images [2] described his 22 layered model as a sophisticated method and reported an accuracy of over 94% for 3 class classification concerning COVID-19, normal, and other Pneumonia infected lungs.

Aras M. Ismael Abdulkadir Şengür in his paper "Deep learning approaches for COVID-19 detection based on chest X-ray images" [3] also recorded a fairly high accuracy across all the pre-trained models, ResNet50, ResNet18, VGG etc. The paper recorded a maximum accuracy of 94.7% after extracting features using the ResNet50 model and then passing them through the Linear Kernel. He then compared it to the 91% accurate end-to-end CNN trained model.

Ozturk, Talo, Yildirim, Baloglu, Yildirim, and Acharyaet [4] automated detection of COVID-19 cases using deep neural networks with X-ray images and developed a Deep Learning network termed as DarkCovidNet based on X-ray images for automated COVID-19 diagnosis. The model achieved a higher accuracy of 87.02% and 98.08% for multi-class (COVID-19, normal, and pneumonia) and two-class (COVID-19 and normal) cases.

II. MATERIALS AND METHODS

A. Dataset

The dataset was acquired from an open source Kaggle Repository. The dataset consists of three classes namely, Normal, Pneumonia, and COVID-19. The test dataset comprises 501 images taken from the Kaggle repository [5]. The dataset chosen is publicly distributed and open source to ensure better credibility of the models.

The division among different classes were as follows:

For training and validation purpose

- 1583 Chest X-Ray images of normal individuals
- 576 Chest X-Ray images of COVID-19 infected patients
- 4273 Chest X-Ray images of Pneumonia infected patients

For testing purpose

- 167 Chest X-Ray images of normal individuals
- 167 Chest X-Ray images of COVID-19 infected patients
- 167 Chest X-Ray images of Pneumonia infected patients

All the images were in different dimensions, so were correspondingly resized to 176 x 176 for better training and testing purposes.

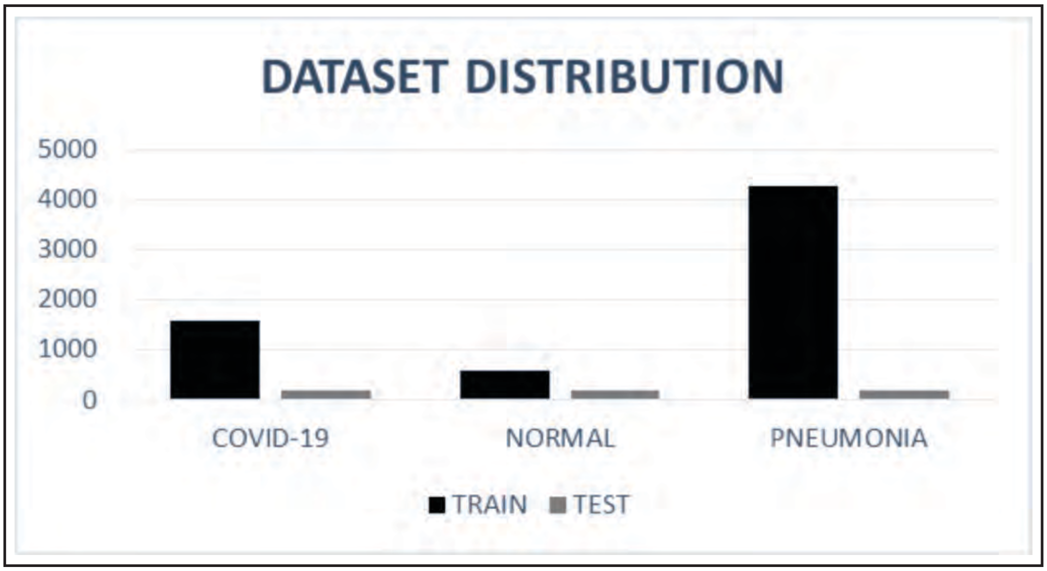

Fig. 1. Distribution of Dataset

III. UNDERSTANDING THE TECHNIQUE

A. Augmentation

The dataset obtained was augmented for better results. Data augmentation is a technique that can be used to exaggerate the dataset to introduce variability and expand the data so that it is more versatile and robust when used to train the models. The augmentation parameters selected were considered with care so as not to distort the image features that factor in deciding the result. Data augmentation features were applied to the dataset were:

- The images were re-scaled to a re-scaling ratio of 1:255.0.
- The images underwent a shear of 0.2 degrees, which is the shear angle, anticlockwise.
- The images were randomly flipped horizontally as well as vertically so as to preserve the lesion marks and create new synthetic data.
- A zoom augmentation introduced randomly zooms the image in and either adds new pixel values around the image or interpolates pixel values.

Data augmentation was applied with the aid of image generators and the output images were resized to 176 x 176 for uniformity. All the images were passed through all three of the channels, that is, red, green and blue (R, G, B).

B. Convolutional Neural Network

Convolutional Neural Network is a class of Neural Networks applied in deep learning mostly to analyse image or visual data [6]. A Convolutional Neuron contains learnable weights and biases which are tweaked in every cycle consisting of a Forward Pass and a Backward Pass. Thousands and millions of these decision-making nodes when arranged in a proper structure generate a Convolutional Neural Network. This graph of nodes is mainly used for classifying or ordering image data if they appear analogous or share properties [7]. The data is convolved using filters or kernels which are basically sliding matrices applying certain functions to the input, generally some form of matrix multiplication.

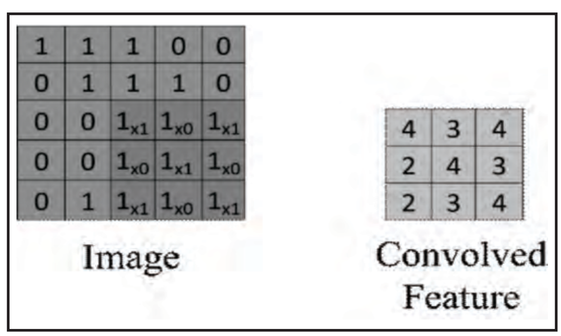

Fig. 2. Convolutional Matrix

C. Transfer Learning

Transfer Learning is the art of using pre-trained feature selection layers, pair them with powerful Dense Layers and then re-train them according to the problem at hand. It is basically the starting point for a new model with previously acquired knowledge.

1) ResNet50

Resnet 50 is a convolutional Neural Network which has a depth of 50 layers. It was initially trained for the ImageNet dataset for over a thousand classes and for a million images. The fundamental breakthrough with ResNet was that it allowed us to train extremely deep neural networks with more than 150 layers successfully. The model has over 23 million trainable parameters with a few thousand non-trainable parameters.

2) DenseNet121

DenseNet was developed specifically to improve upon the declined accuracy which was caused by the vanishing gradient in high-level deep neural networks. In simpler terms, due to the longer path between the input layer and the output layer, the information vanishes before reaching its destination or even more simply the information is changed extremely radically from its way into its way out causing a loss in accuracy.

3) InceptionNet_V3

The InceptionNet model was introduced as a 48-layer deep model scoring over 78% in the 1 million imageNet Dataset and is the third installment of Google's InceptionNet Module. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch-normalisation is used extensively throughout the model and applied to activation inputs while the loss is computed via SoftMax.

4) MobileNet

MobileNetV2 uses depth wise separable convolution as its foundation for efficient computational capability. It is well known for its reduced size and complexity without encroaching upon its capability. It proves to be much better than its former version in terms of reduced parameters and less computations. Apart from the depth wise separable convolution, MobileNet consists of bottlenecks between the layers and shortcut connections between the said bottlenecks enabling faster training and better accuracy.

5) XceptionNet

XceptionNet was designed as an improvement over the InceptionNet model. It is said to be computationally more efficient and robust as compared to the Inception-v3 model because of the presence of modified depth wise separable convolution. The modified depth wise convolution involves performing 1x1 convolution first followed by channel-wise spatial convolution. Also, it does not involve any intermediate non-linearity.

6) VGG16

VGG16 is a Convolutional Neural Network model with large kernel-sized flyers in its initial layers. Due to the dense structure and architecture of the model, it is immensely slow in training and the network weights are quite large. The network input consists of images of dimensions 224 x 224. A padding of 1 pixel is added after every convolutional layer. The hidden layers use ReLU as the activation function to allow increased efficiency in the computational task and avoid the problem of vanishing gradients.

D. Dense Layers Architecture

We employed to the best of our knowledge a new structure of Dense and activation Layers and linked them to multiple pretrained feature selection models [8]. This

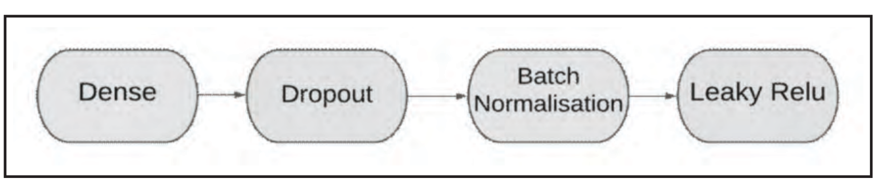

Fig. 3. Dense Layer Structure

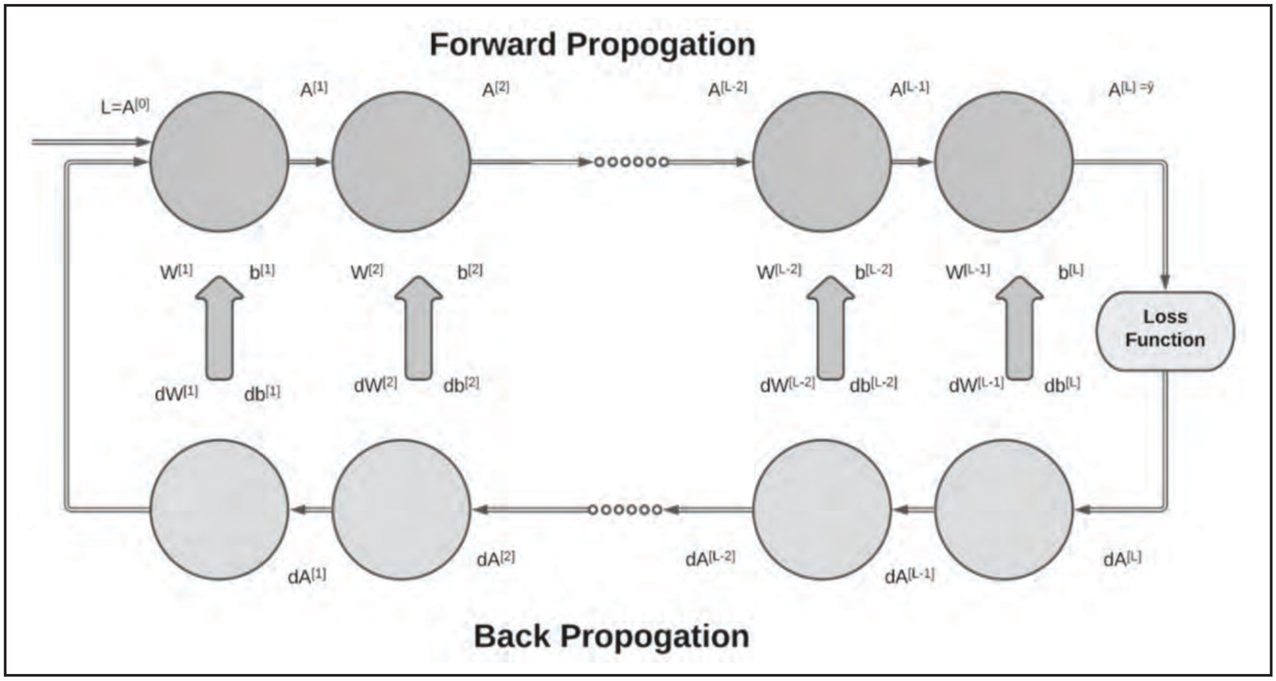

Fig. 4. Conceptualisation of Forward and Backward Propagation

helped us augment the power of those models to a great extent and helped us achieve an accuracy far beyond if we would have used any other Machine Learning Technique, be it Support Vector Machine or Random Forests etc. We will now try to explain the underlying repeating structure for the Dense layer model we have employed.

Each neuron in every layer is basically used to create a weighted average of the input it receives from the previous layer. This input is then propagated forward to the next layer, where the next neuron uses its weights to create a new double which then gets passed on. This happens in every neuron of every layer until the output layer, where output of each neuron gets classified into one of the many classes possible according to the rule set. The accuracy or the loss function then calculated is used as a metric to measure the effectiveness of the weights of each neuron in the previous pass. The result obtained is back-propagated in the neural network [9]. Pertaining to the results produced each neuron adjusts its weights and biases to produce the desired result which is more accurate.

1) The Dense Neuron

The dense layer is a Neural Node Layer which is deeply connected, which means that every node in the i^{th} layer receives input from all the layers in the $i\text{-}1^{th}$ layer and transfers its output to every node in the $i+1^{th}$ layer. What a deep neuron basically does is that it performs matrix-vector multiplication. The parameters used in the multiplication are the tuned values of a node (weights and biases) which are updated every step per epoch. The output generated is an m-dimensional vector which is passed on to the layers further down the network.

$$y = \sum (w * input) + b \qquad (1)$$

Where w is the weight, b is the bias, and y is the output we are looking for.

2) The Dropout Layer

The dropout layer solves a very big problem encountered by any deep neural network, "Overfitting". The dropout layer sparsely selects neurons and deactivates them for a certain pass causing the model to lose a few nodes randomly, hence preventing the overfitting of data due to too many parameters. Dropout has the effect of making the training process noisy, forcing nodes within a layer to probabilistically take on more or less responsibility for the inputs.

3) Batch Normalisation

Batch Normalisation is a technique for training very deep

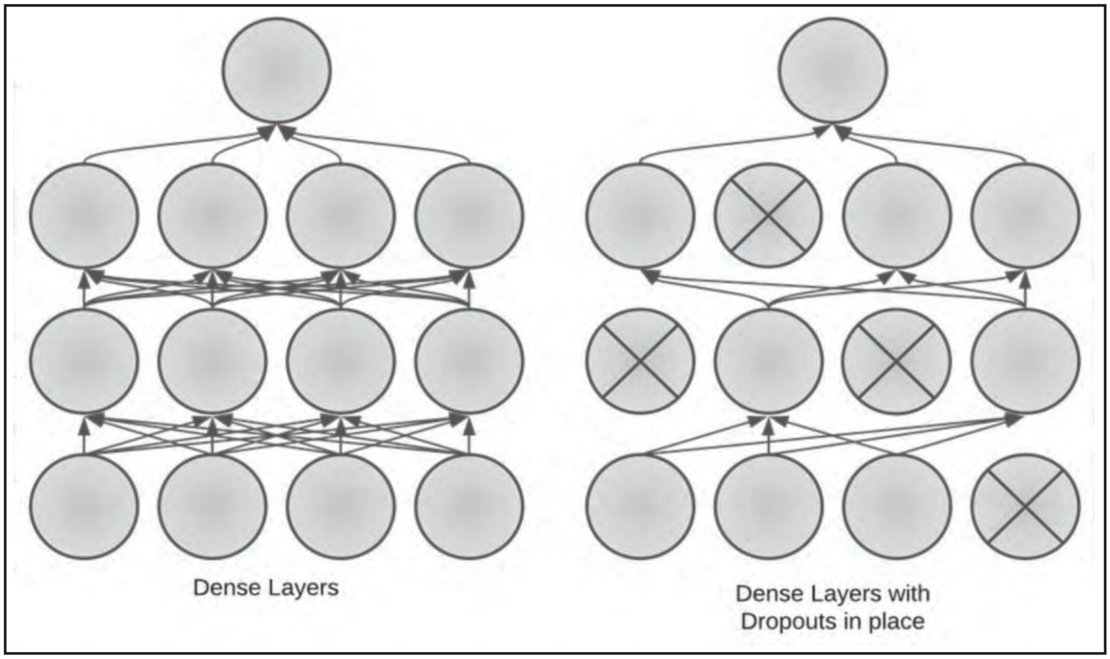

Fig. 5. Dense Nodes and Dense Nodes With Dropout in Place

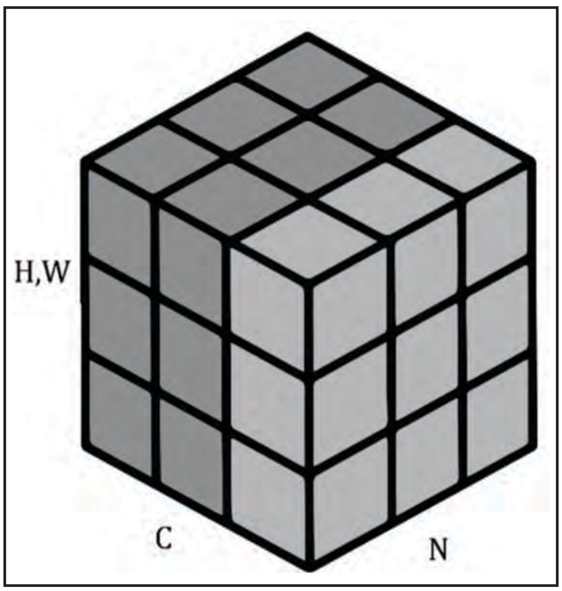

Fig. 6. Batch Normalisation Visualisation

neural networks that standardises the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and introducing a factor of robustness and dramatically reducing the number of training epochs required. Batch Normalisation reparametrizes the model to make some units always be standard by definition.

$$x^{*} = (x - E[x]) / \sqrt{(var(x)} \qquad (2)$$

Where x^* is the new value of the single component, *E[x]* is its Expectation or the mean within a batch and *var(x)* is the variance within a batch

$$x^{**} = \gamma * x^{*} + \beta \qquad (3)$$

Where x^{**} is the final normalised value, while γ and β are learned per layer

4) Leaky ReLU

Leaky ReLU is an advancement over the standard Rectifier function. The new function by-passes the dead neuron problem as instead of resulting in a 0 for a negative valued input, the Leaky ReLU function assigns a very small positive quantity to the output.

$$f(x) = \max(a, x), \text{ where } a \text{ is a very small value} \qquad (4)$$

5) Loss Function

The optimal selection of a loss function is incredibly important as it functions as a feedback mechanism to the backpropagation cycle. The function we selected for our testing and training purposes was the categorical cross entropy to minimise the culminating loss.

$$Loss = -\sum_{i=1}^{\substack{output\\size}} (y_i \cdot \log \hat{y}_i) \quad (5)$$

E. Optimizers and Metrics

Optimizers or the hyper-parameters of the function help in micro-tuning of the model to get that last bit of possible accuracy. The main objective, as has been, is to get the maximum accuracy over the previously un-exposed to testing data. The 'adam' optimizer was used for training the model with a learning rate of 0.0001. We used validation loss as a point of reference for the learning rate and whenever it crossed a certain threshold, we updated the learning rate. The metric used for the model was 'accuracy' and Early stopping was employed to prevent overfitting or overtraining of the model. Here too, the monitored hyper-parameter was the validation-loss with a high yield patience level.

The activation function defines the output of a node with reference to a particular input or a set of inputs. In their simplest forms they can be understood as "ON" or "OFF" switches which are triggered based on the initial conditions. We here enlisted softmax to provide the desired optimal results which works on a probabilistic model.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \quad (6)$$

σ = softmax

$\vec{z}$ = input vector

K = standard exponential function for input vector

e^{z_i} = number of classes in the multi-class classifier

e^{z_j} = standard exponential function for output vector

IV. RESULT

The test data consisted of 501 images, with 167 images from each class. The testing metrics were calculated according to each class for every model including accuracy, Precision, Recall, and $F1$-score. The confusion matrix and the graphical representation of the testing metrics for each model was prepared for better insights related to the efficiency and robustness of the model.

$$Precision = \frac{True\ Positive}{True\ Postive + False\ Positive} \quad (7)$$

$$Recall = \frac{True\ Positive}{True\ Postive + False\ Negative} \quad (8)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Postive + False\ Negative + False\ Positive + True\ Negative} \quad (9)$$

$$F1\ Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (10)$$

A. MobileNet

MobileNet is a light and easy to train model with less parameters and is usually trained for handheld devices and low compute power devices. This model was constructed for better efficiency and low power and resource consumption. Despite its lighter weight, the model performed quite well in classifying COVID-19 from chest X-Ray images. The model accuracy on the

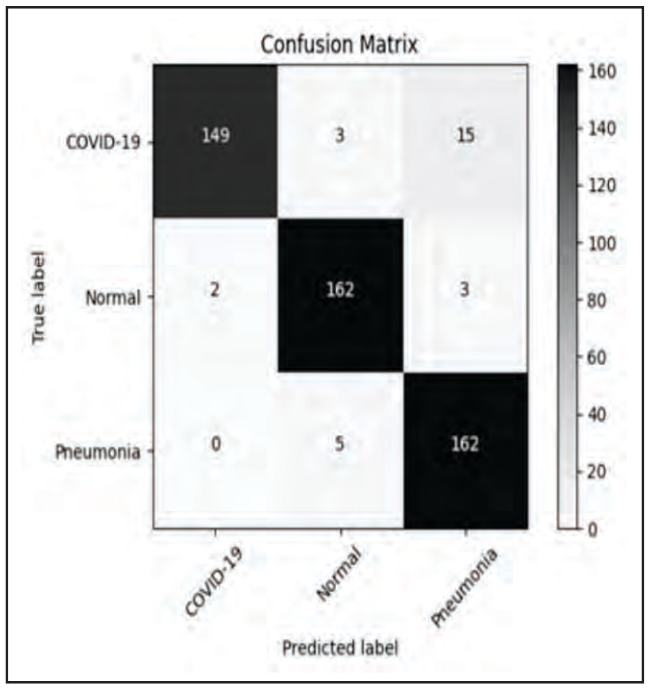

Fig.7. Confusion Matrix – MobileNet

TABLE I.
TABULATION OF TESTING METRICS - MOBILENET

MobileNet	PRECISION	RECALL	*F*1-SCORE
COVID-19	0.9867	0.8922	0.9371
NORMAL	0.9529	0.97	0.9614
PNEUMONIA	0.9	0.97	0.9337

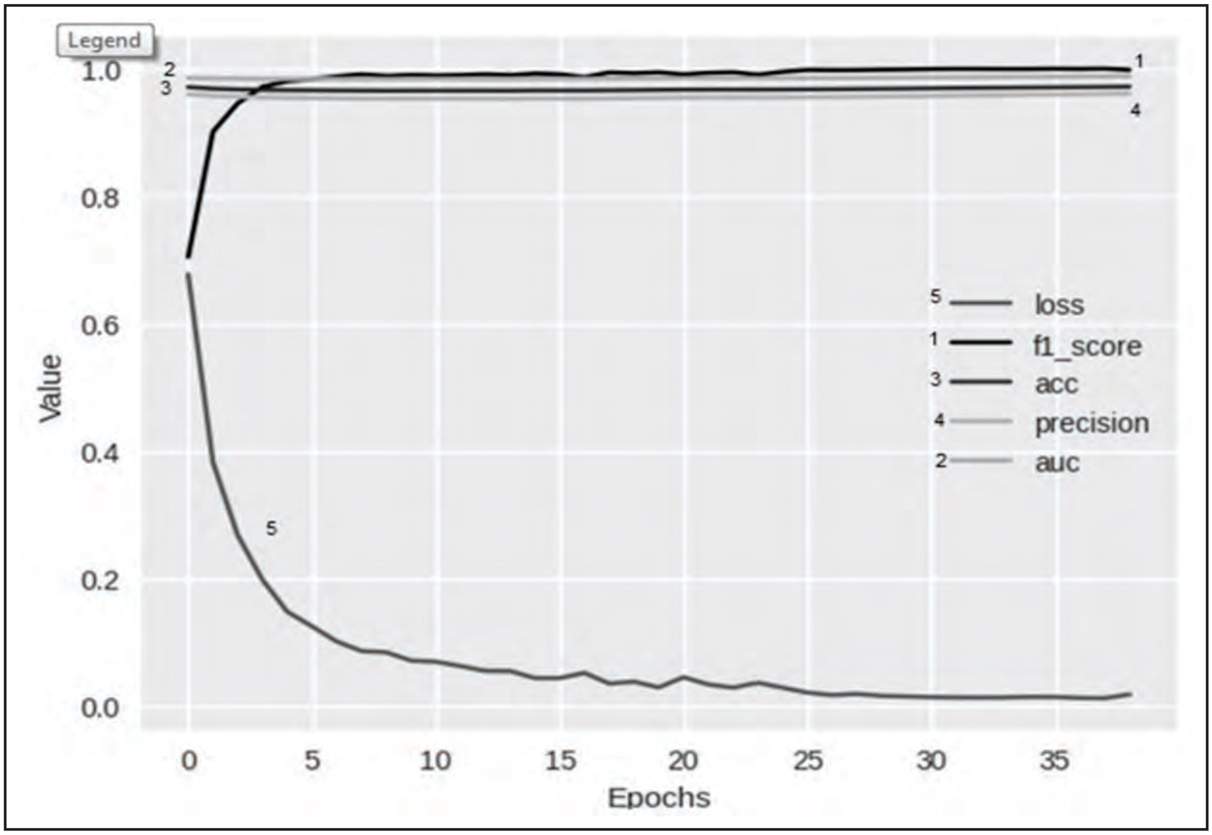

Fig. 8. Graphical Representation of Training and Validation Metrics – MobileNet

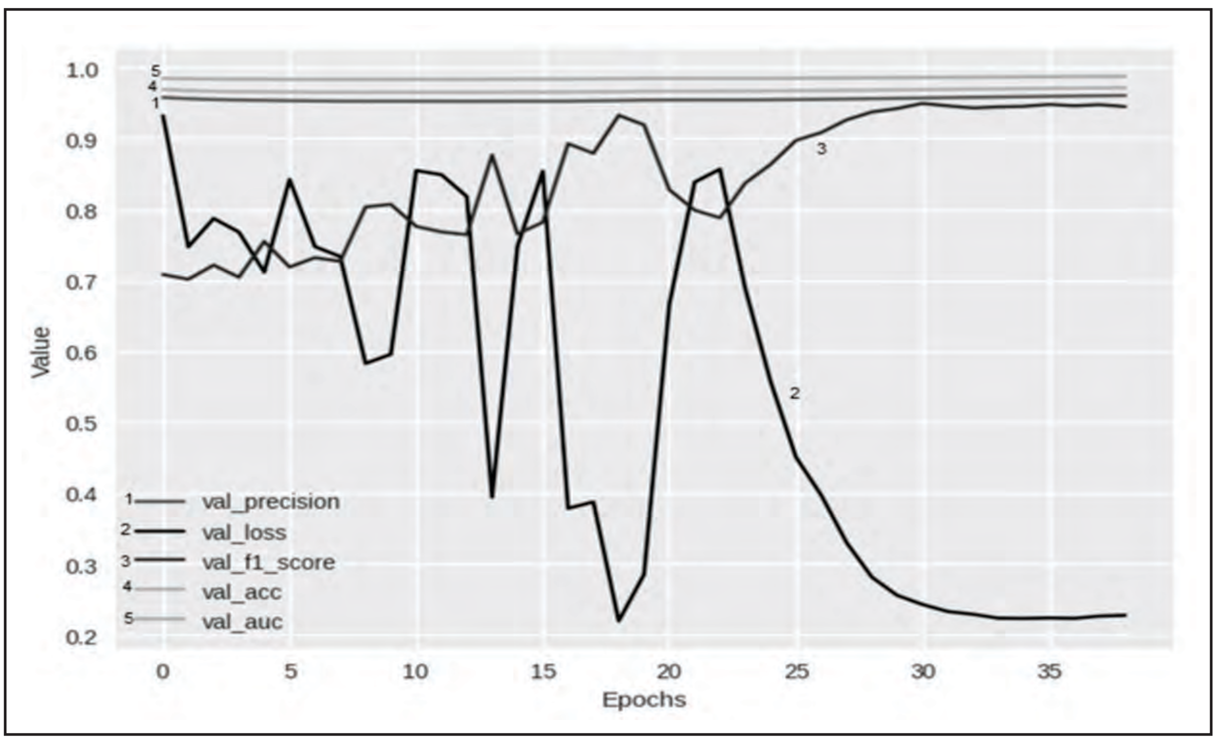

Fig. 9. Graphical Representation of Validation Metrics – MobileNet

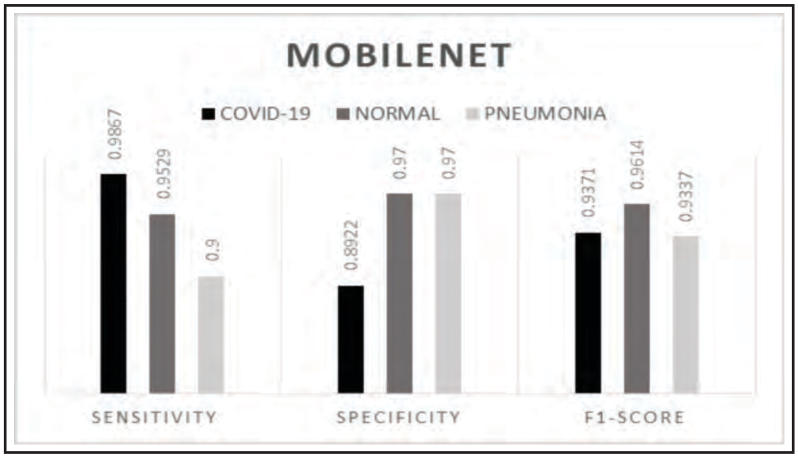

Fig. 10. Visualisation of Testing Metrics - MobileNet

testing dataset was about 94%. The confusion matrix was also plotted on the testing data for better insights of the model's efficiency and accuracy.

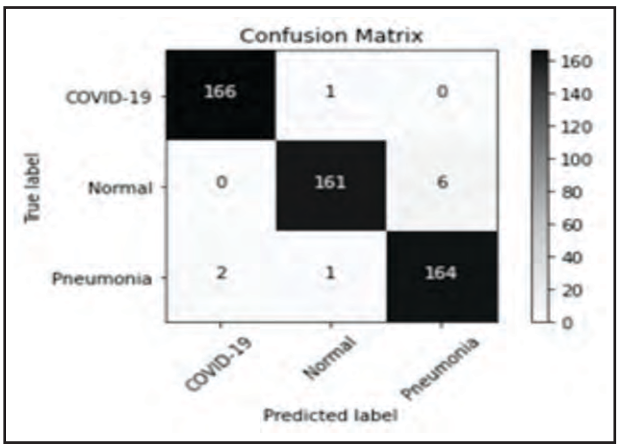

Fig. 11. Confusion Matrix – ResNet50

Metrics such as accuracy, $F1$-Score, sensitivity, and specificity were also plotted during the training and validation phase of the model for each epoch.

B. ResNet50

The ResNet model encompasses the concept of skipping connections which means that input to a layer can be passed to some other layer without having to go through

TABLE II.
TABULATION OF TESTING METRICS – RESNET50

ResNet50	PRECISION	RECALL	*F*1-SCORE
COVID-19	0.9881	0.994	0.991
NORMAL	0.9877	0.964	0.9757
PNEUMONIA	0.9647	0.982	0.9732

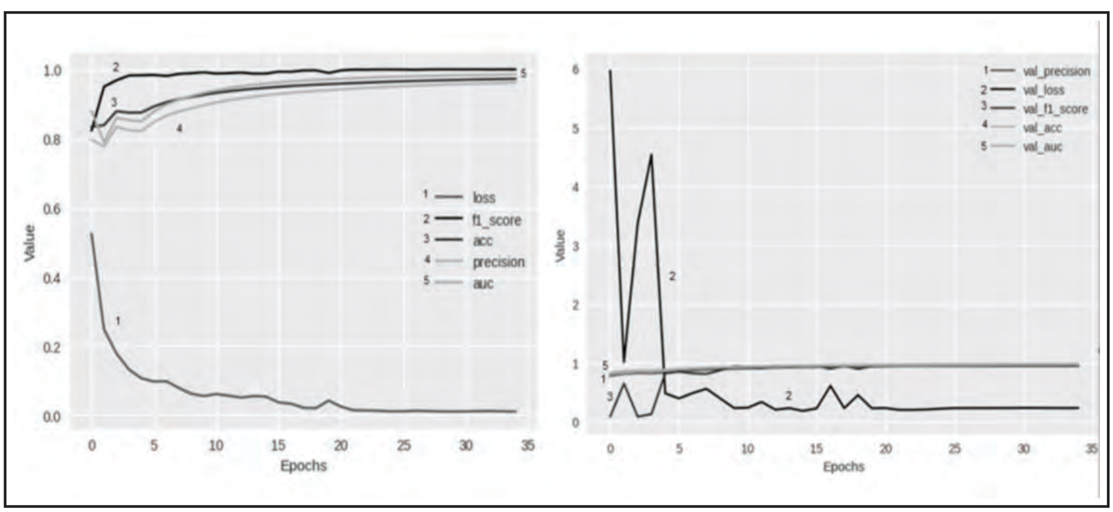

Fig. 12. Graphical Representation of Training and Validation Metrics - ResNet50

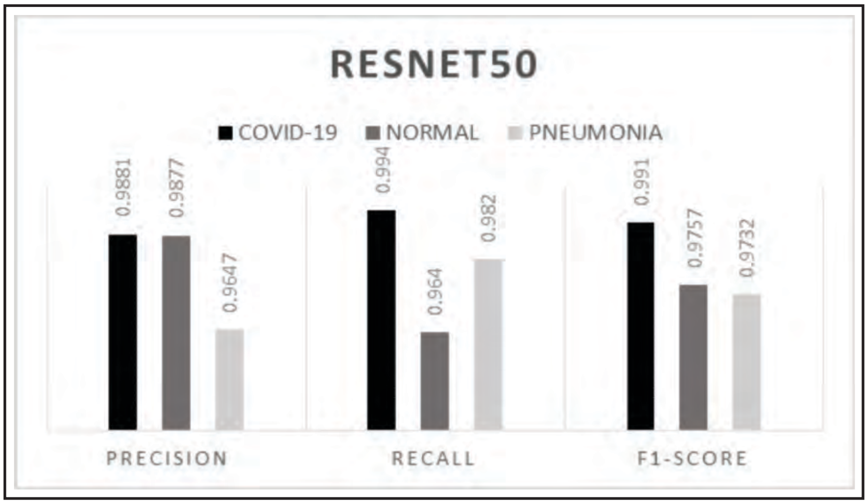

Fig. 13. Visualisation of Testing Metrics – ResNet50

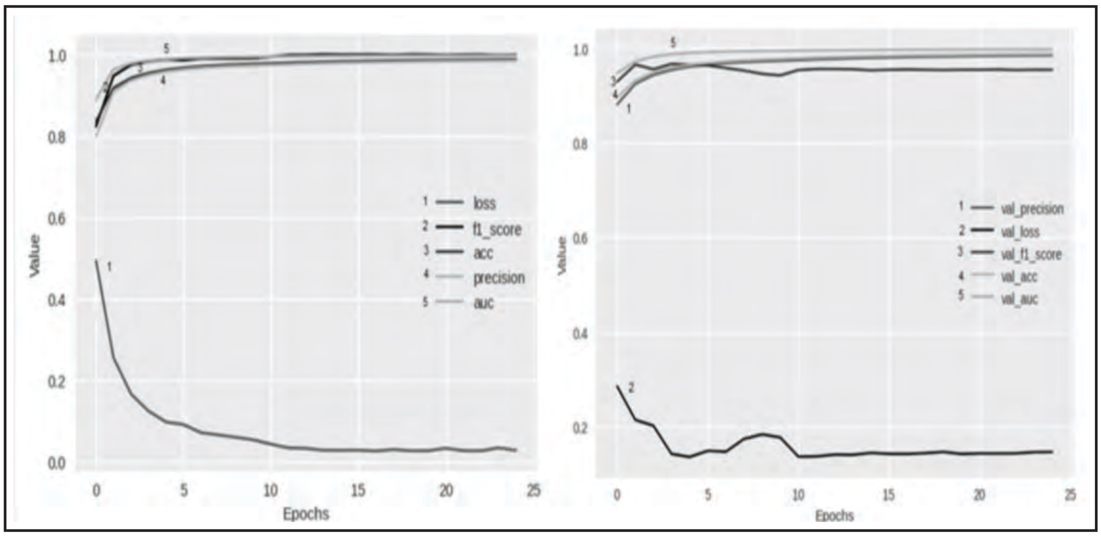

Fig. 14. Graphical Representation of Training and Validation Metrics – XceptionNet

the layers in between or by skipping layers. The model also employs identity mapping which helps to avoid the vanishing gradients problem. The accuracy obtained over the test dataset was about 98%. For visual insights, the confusion matrix, processed from the test data, and the graphical representation of the training and validation metrics are shown in Fig. 11 to 13.

C. XceptionNet

The XceptionNet architecture makes use of modified depth wise convolution and is about 36 layers deep. The order of convolution and arrangement of its layers help in achieving computational efficiency and makes it better and faster than Inception-V3. The model was trained on the same training dataset and the performance analysis

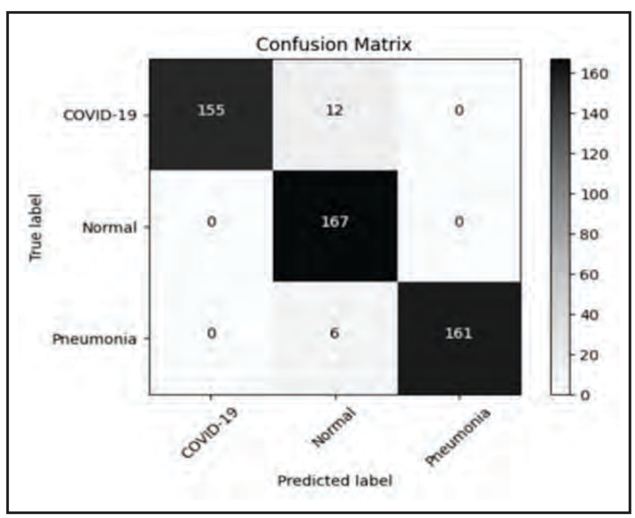

Fig. 15. Confusion Matrix – XceptionNet

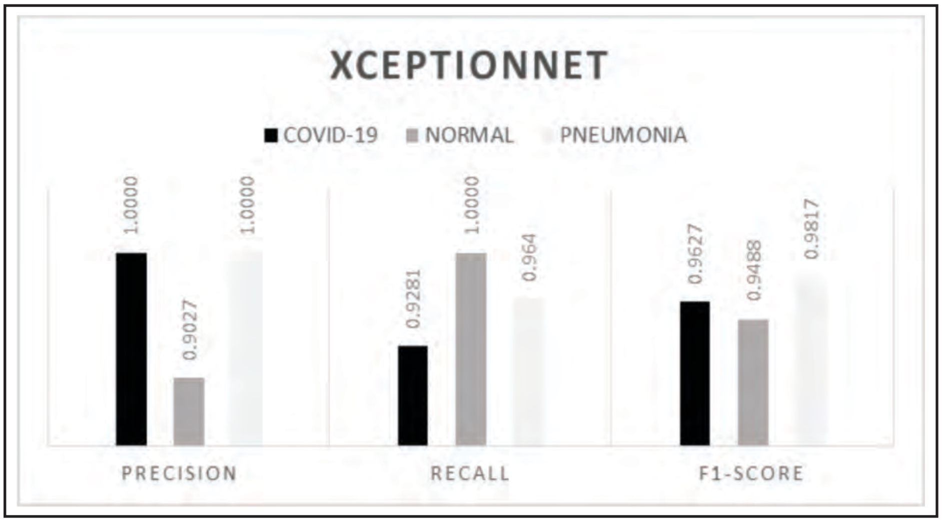

Fig. 16. Visualisation of Testing Metrics - XceptionNet

TABLE III.
TABULATION OF TESTING METRICS - XCEPTIONNET

XceptionNet	PRECISION	RECALL	*F*1-SCORE
COVID-19	1.0000	0.9281	0.9627
NORMAL	0.9027	1.0000	0.9488
PNEUMONIA	1.0000	0.964	0.9817

was done through the validation set. Some metrics were considered and plotted during the training and validation phase of the model for better understanding of the computed results.

The confusion matrix was analysed and testing metrics were calculated for fruitful insights. The model accuracy for the given model on the test dataset was about 96%.

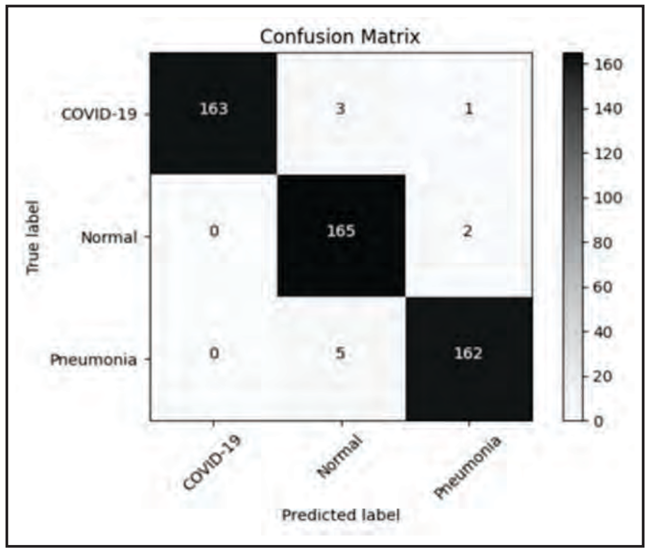

Fig. 17. Confusion Matrix – Inception_V3

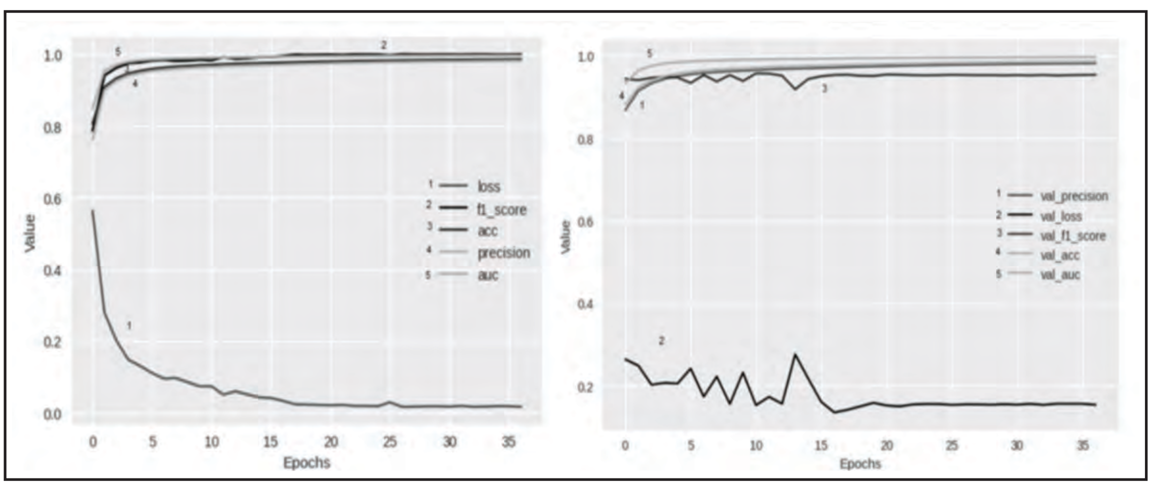

Fig. 18. Graphical Representation of Training and Validation Metrics – Inception_V3

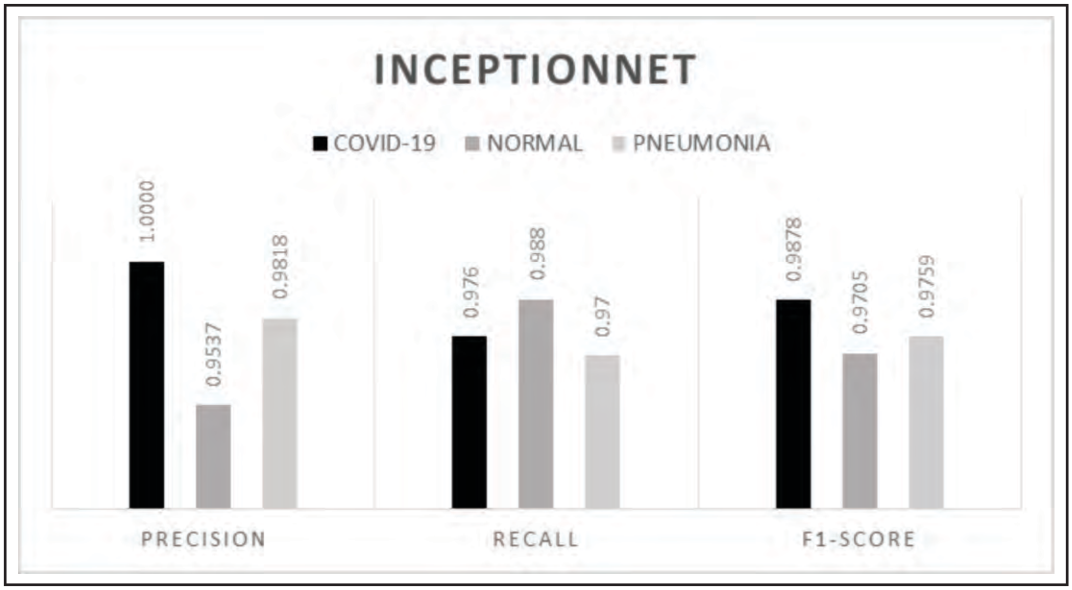

Fig. 19. Visualisation of Testing Metrics - InceptionNet

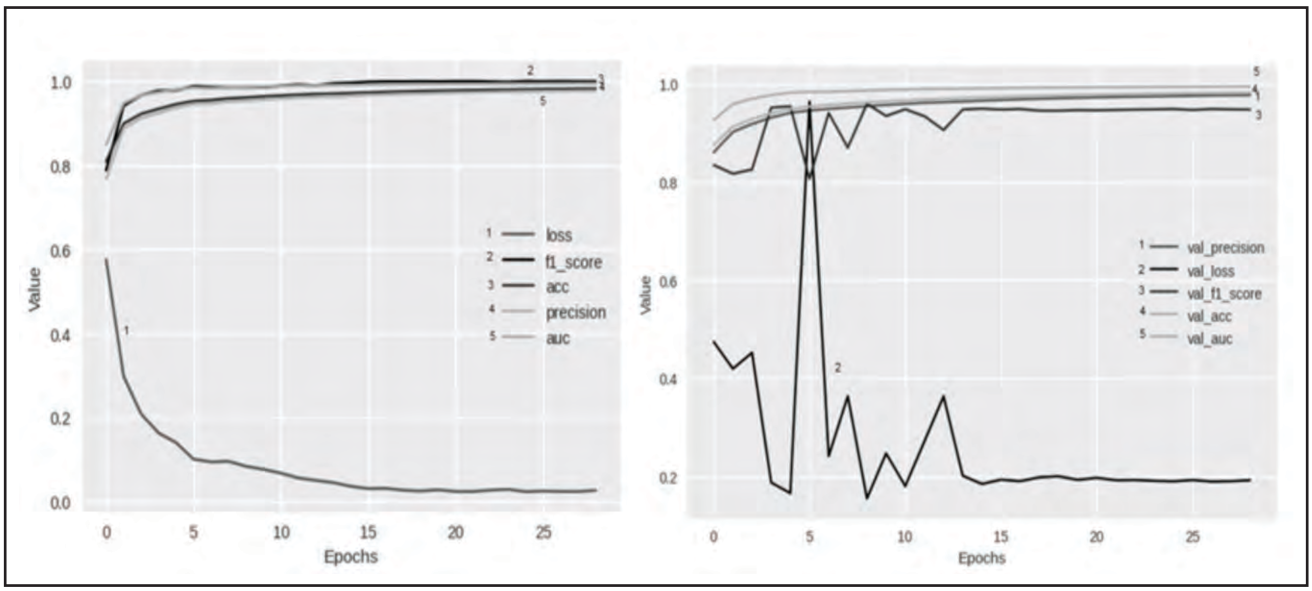

Fig. 20. Graphical Representation of Training and Validation Metrics – DenseNet121

TABLE IV.
TABULATION OF TESTING METRICS – INCEPTION_V3

InceptionNet	PRECISION	RECALL	F1-SCORE
COVID-19	1.0000	0.976	0.9878
NORMAL	0.9537	0.988	0.9705
PNEUMONIA	0.9818	0.97	0.9759

D. InceptionNet

Deep models and more parameters can tend towards overfitting a solution for this is to move on to sparsely connected network architectures which will replace fully connected network architectures. This is the basic principle of the InceptionNet model and key benefits of it are better speed and accuracy. The model gave an accuracy of around 97.8% when tested on the test dataset.

The results from all the models were compiled and the metrics plotted. Some metrics were tabulated based on the results obtained from the prediction of the test set. The metrics used were Precision, Recall, and $F1$-Score which were used to compare the model performance with the other models.

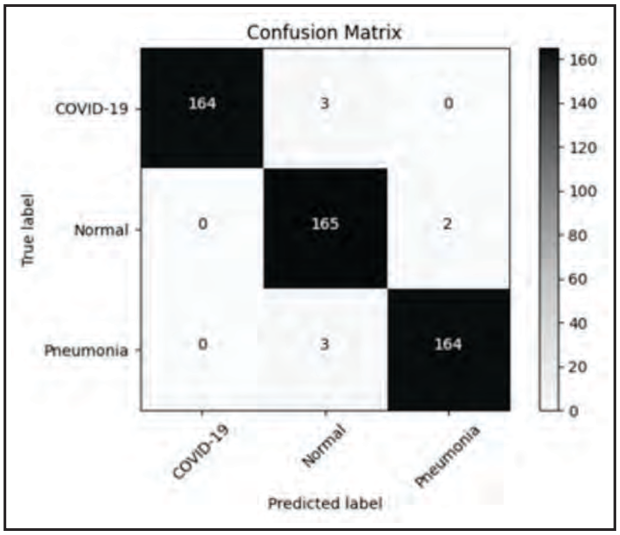

Fig. 21. Confusion Matrix – DenseNet121

E. DenseNet121

DenseNet121 performed extremely well on the Validation split as well as on the final test split. The Neural Network paired with Dense Layers which in turn complemented with Leaky ReLU gave high accuracies as well as $F1$ Score. DenseNet helped solve the problem of vanishing-gradients and enabled feature reuse and propagation leading to substantial results and accuracy. For visual insights, the graphical representation of the metrics used during training and validation steps and the Confusion Matrix derived from the test set are shown in Fig. 20 and 21 respectively.

The testing metrics and results for the model are shown in Table V and for visual representation a graph (Fig. 22) has been plotted to shown how better each class was predicted by the model.

F. VGG16

The VGG16 model performed exceptionally well on the dataset. The VGG16 architecture, like all other models was augmented with a combination of feature extraction layer which gave high accuracies on the training, validation, and the test dataset. The confusion matrix along with the metrics used during the training and validations steps were plotted to draw comparisons with the other models used in this study. The model gave an accuracy of over 99% on the test data.

TABLE V.
TABULATION OF TESTING METRICS – DENSENET121

DenseNet121	PRECISION	RECALL	*F*1-SCORE
COVID-19	1.0000	0.982	0.9909
NORMAL	0.9649	0.988	0.9763
PNEUMONIA	0.9879	0.982	0.9849

TABLE VI.
TABULATION OF TESTING METRICS – VGG16

VGG16	PRECISION	RECALL	*F*1-SCORE
COVID-19	0.994	0.994	0.994
NORMAL	0.988	0.988	0.988
PNEUMONIA	0.988	0.988	0.988

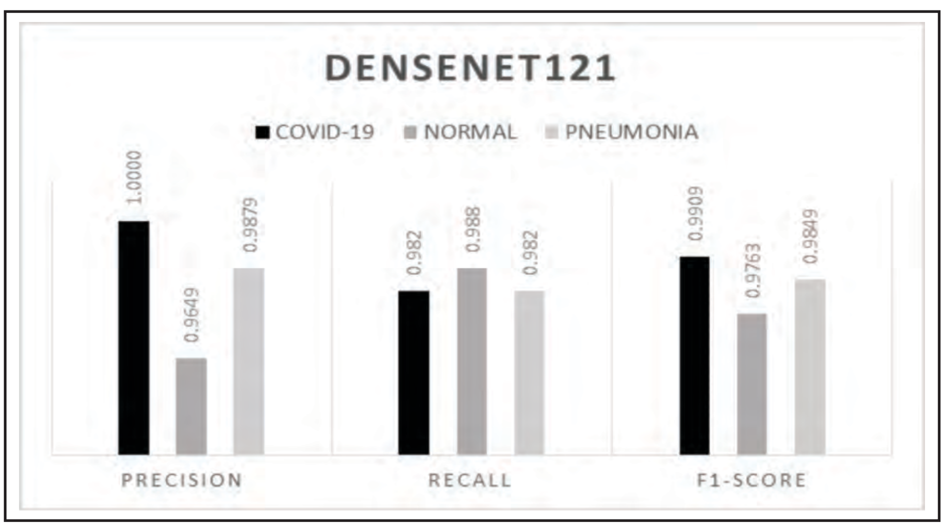

Fig. 22. Visualisation of Testing Metrics – DenseNet121

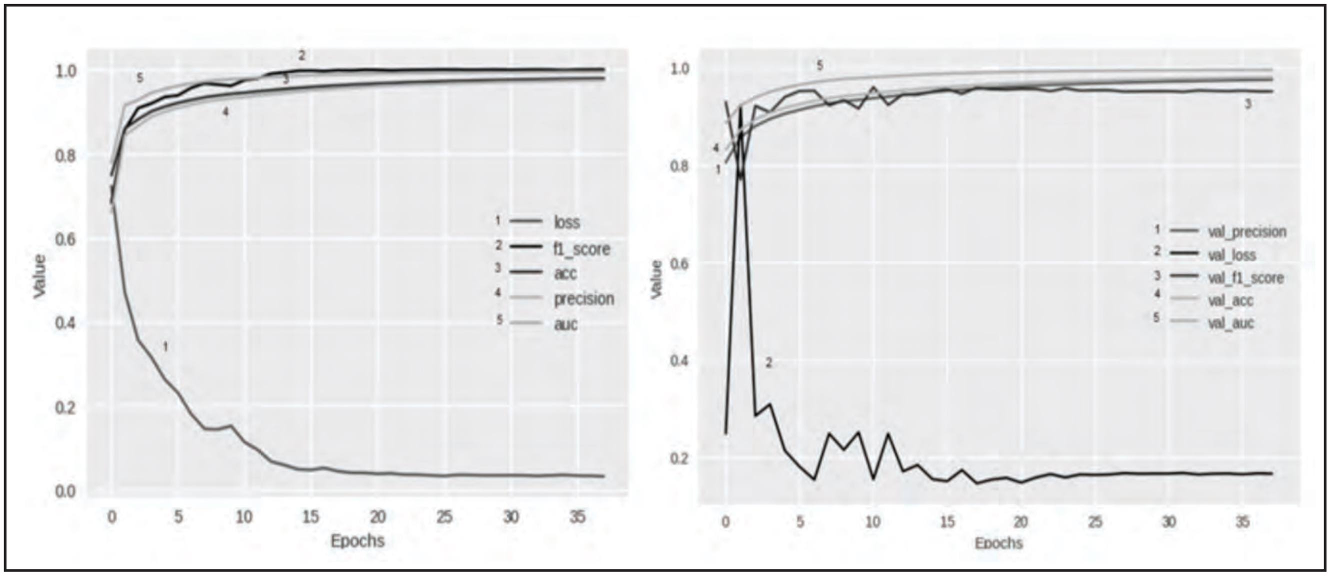

Fig. 23. Graphical Representation of Training and Validation Metrics – VGG16

The testing metrics and results for the model are shown in Table VI and for visual representation a graph (Fig. 25) has been plotted to show how better each class was predicted by the model.

G. CROSS ANALYSIS

The results from all the models were compiled and the metrics plotted. Some metrics were tabulated based on the results obtained from the prediction of the test set. The metrics Accuracy, Precision, Recall, and $F1$-Score were used to compare the model performance among all the chosen architectures.

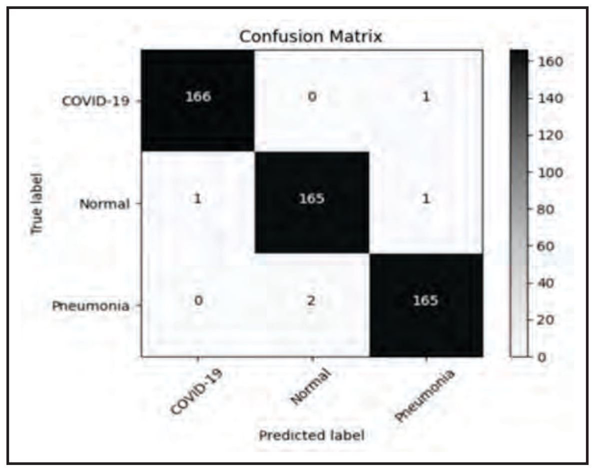

Fig. 24. Confusion m Matrix – VGG16

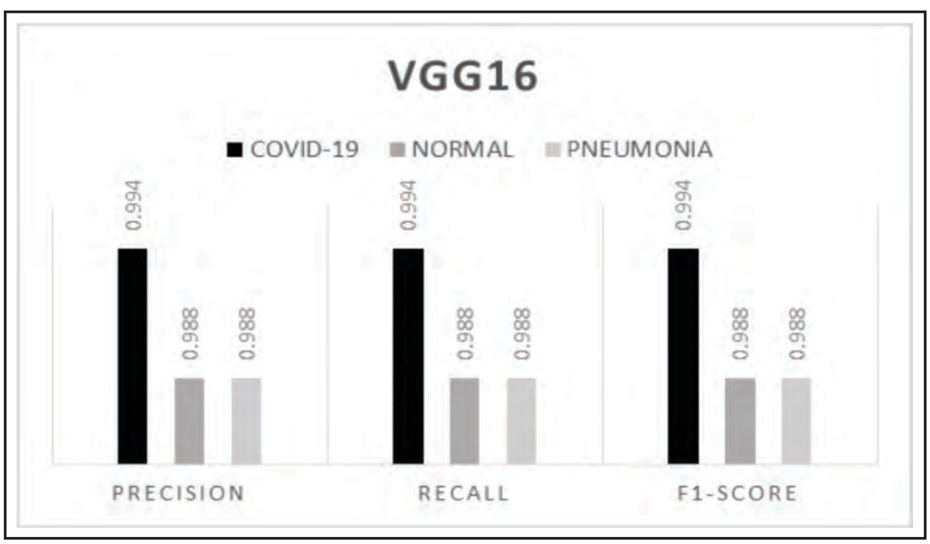

Fig. 25. Visualisation of Testing Metrics – VGG16

TABLE VII.
COMPARATIVE ANALYSIS OF TESTING METRICS

METRICS	MobileNet	ResNet50	InceptionNet	XceptionNet	DenseNet121	VGG16
Accuracy	0.9441	0.98	0.978	0.964	0.984	0.99
Precision	0.9465	0.9801	0.9785	0.9676	0.9843	0.99
Recall	0.944	0.98	0.978	0.964	0.984	0.99
*F*1-Score	0.944	0.9799	0.978	0.9644	0.984	0.99

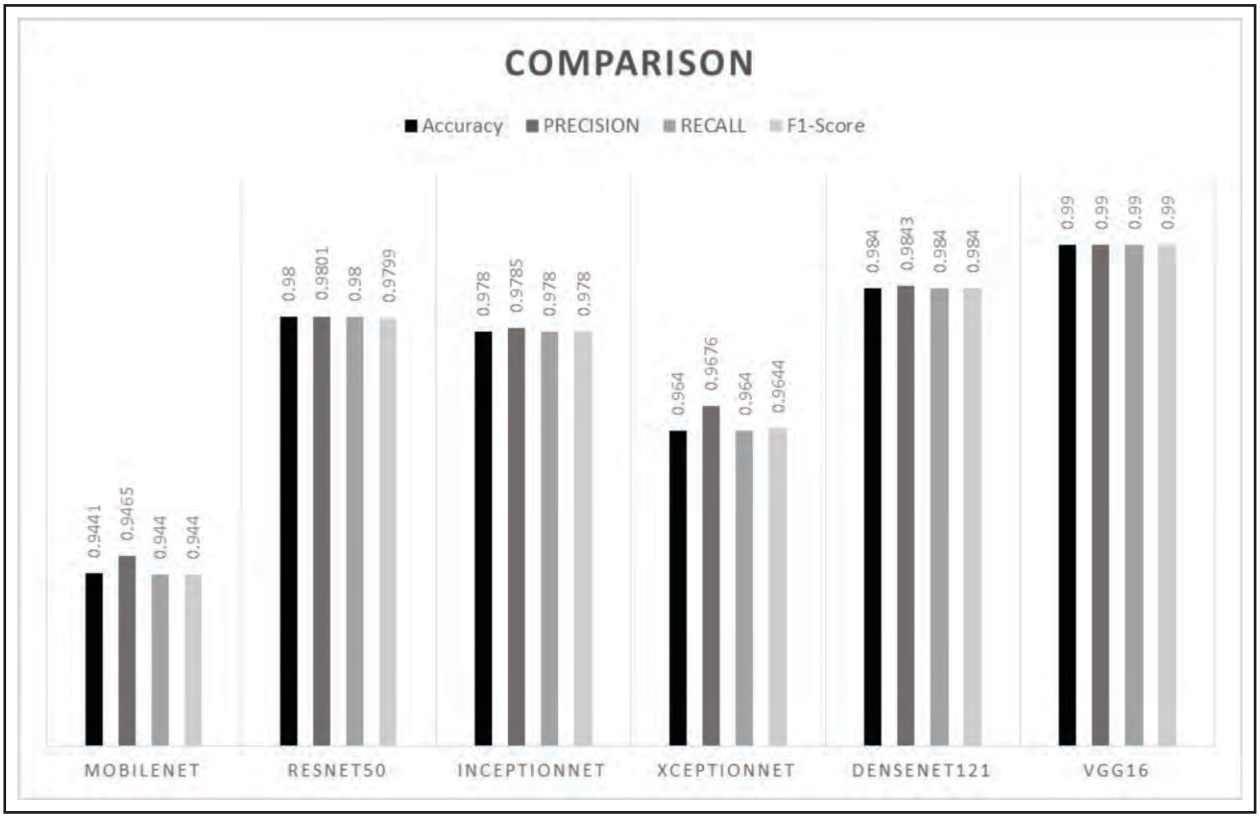

Fig. 26. Graphical Comparative Analysis of Testing Metrics

V. CONCLUSION

Machine Learning and Artificial Intelligence based classification can help in instantaneous diagnosis and classification of skin cancer and like diseases. The models developed by us are proof of the concept that cost-effective, user-friendly, and non-invasive Machine Learning and Artificial Intelligence based methods can be developed to win the battle against COVID-19 and other diseases. During the research process, each model trained was a successor to the last one, every new model we created took notes from the shortcomings of the previous one. All the insights gained from the precursing models were employed to improve the accuracy as well as the robustness for the future to be trained models.

In this paper we have provided a comparative analysis between multiple Transfer Learning based models and have logged all the data to enable ease in identifying COVID-19 from chest X-Rays. The results obtained from this study have been presented both in written and visual (with the help of graphs and confusion matrices) form for the perusal of the scientific community.

Further, we hope that our findings will help with the ongoing crisis and be a useful contribution in fighting against this virus. However, human judgement is indispensable. Hence with our model, we are not trying to replace any physician but merely aid with the predictions and decision-making process.

VI. FUTURE SCOPE

This study can act as a base for further research and development in this field. The world is experiencing a surge in COVID-19 cases and this study can go a long way in the development of assistive technology that would aid the frontline workers in tackling the gravity of this situation. The elementary results seem promising and such studies can prove to be exorbitant towards the development of finer technology to fight this pandemic.

AUTHORS' CONTRIBUTION

Tismeet Singh and Kartikeya Agarwal have performed the entirety of the research work as mentioned in the paper. Both the authors conceived the idea, developed the theory, and trained the models. All authors discussed the results and contributed to the final manuscript.

CONFLICT OF INTEREST

The authors proclaim that there is no irreconcilable situation with respect to the distribution of this paper. The authors declare that they have no financial interests or personal affiliations that could have influenced the study and its findings.

FUNDING ACKNOWLEDGEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *"Deep Learn. (Adaptive Computation Mach. Learn. Serie.),"* 2016.

[2] E. Hussain, M. Hasan, M. A. Rahman, I. Lee, T. Tamanna, and M. Z. Parvez, "CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images," *Chaos, Solitons Fractals,* vol. 142, 2021, doi: 10.1016/j.chaos.2020.110495

[3] "WHO Coronavirus Disease (COVID-19) Dashboard," *World Health Org.* https://covid19.who.int

[4] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, "Automated detection of COVID-19 cases using deep neural networks with X-ray images," *Comput. Biol. Med.*, vol. 121, 103792, 2020. doi: 10.1016/j.compbiomed.2020.103792

[5] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 *Int. Conf. Eng. Tech. (ICET),* 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186

[6] P. Patel, "Chest X-Ray (Covid-19 & Pnuemonia)." https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia¬

[7] B. Ramsundar and R. B. Zadeh, *"Fully Connected Deep Networks,"* in Tensor Flow for Deep Learn. [Online] Available: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html

[8] Wilame, "The math behind neural networks - Analysing a dense layer," Vallant.in

[9] A. M. Ismael and A. Şengür, "Deep learning approaches for COVID-19 detection based on chest X-ray images," *Expert Syst. Appl.,* vol. 164, 2021, doi: 10.1016/j.eswa.2020.114054

About the Authors

Tismeet Singh is an Undergraduate student of Computer Science at Netaji Subhas University of Technology. He is a Machine Learning and Artificial Intelligence enthusiast and is currently working in the field of Computer Vision and Natural Language Processing.

Kartikeya Agarwal is a Computer Science student currently in the 3rd year of his Bachelors Degree. He has great fervour for Machine Learning and Data Science and is working on real-life problems with the assistance of programming.