International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 12 • Issue 1 pp. 39-47 Jan - June 2020 www.csjournals.com

Predictive Modeling of COVID-19

[1]Susheen Srivatsa C N, [1]Adithya Prasanna Murthy, [1]Bharath G C, [1]Priya Chandana R J, [2]Rakesh M B
Students, Assistant Professor
Siddaganga Institute of Technology, Tumakuru, Karnataka, India

Abstract: The Novel Corona Virus pandemic, COVID-19 has surprised the world with its rapid spread, potential virulence and with potential profound overall impact on the lives of billions of people both in terms of safety as well as an economic perspective [1]. The spread of this virus has been drastically increasing and requires a very effective tool kit to predict the presence of the disease in the human body. There are limited number of COVID-19 test kits available in hospitals due to increase in the number of cases daily [2], hence the creation of a machine learning model helps to detect the current status in predicting the virus inside a Human Being with a certain probability. The main objective of this approach is to build a CNN model which takes the chest x-ray image as an input and predict the presence of the corona virus.

Keywords: CNN Model,COVID-19,Corona Virus, Neural Network.

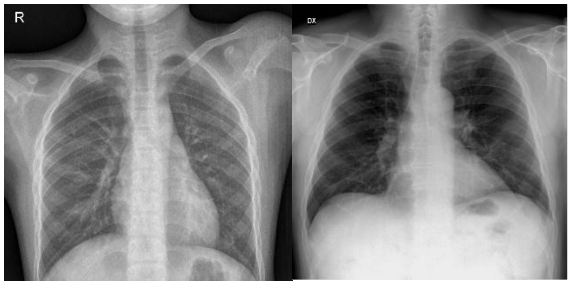
Introduction

As the entire world is facing severe threat from coronavirus (COVID-19) pandemic, it has become a challenging task for the entire medical research team around the world, to find a cure for the virus [3]. Meanwhile predicting the presence of corona virus in the human body by collecting the blood samples of the patient is also a tough task. Hence, creating a Machine learning model aids the medical research team in predicting the current status of the patient.

Deep learning models have been successfully used in many areas such as classification, segmentation and lesion detection of medical data. Analysis of image and signal data obtained with medical imaging techniques such as Magnetic Resonance Imaging (MRI), Computed Tomography (CT) and X-ray are done with the help of deep learning models [2]. Medical research observes that the virus mainly affects the respiratory tracts of birds and mammals, including humans. According to the World Health Organization, COVID-19 open holes in the lungs like SARS, giving them a "honeycomb-like appearance" [2,5]. Hence, this information leads all the Data scientists to take the image of chest x-ray for the prediction.

Since ‘Convolution Neural Network (CNN)’ models are mainly used in image recognition, image classification, object detection etc. This Approach helped us to create a model for Probabilistic Prediction and Classification of Normal Healthy chest v/s one with the virus. Since, it is a Classification based problem, the model uses supervised learning approach to solve or classify the given input image. The model will be initially trained with two types of data sets [7] one which shall include normal healthy chest images and another one containing image having the presence of corona virus.

The below two images ‘fig 1.a’ and ‘fig 2.b’ [7] show the difference between the normal healthy chest x-ray and the one affected by the virus.



a b

Fig. 1: a. Normal healthy human Chest X-ray, b. Chest with Corona Virus disease [7].

2. Model Description:

The main source of learning the knowledge about the prediction for the model is the data sets. Our model data sets consist of 234 normal chest images and 154 images containing corona virus [7], these images will be fed to train the model. The CNN model consists of different layers namely convolution, max-pooling, flattening and

fully connected network, where the training data sets have to go through all these layers after which the forward and backward propagation results in predicting the presence of coronavirus.

Fig 2 depicts the whole description or a procedure of the model step by step. There are 3 levels, namely:

1. Data pre-processing
2. CNN model
3. ANN model

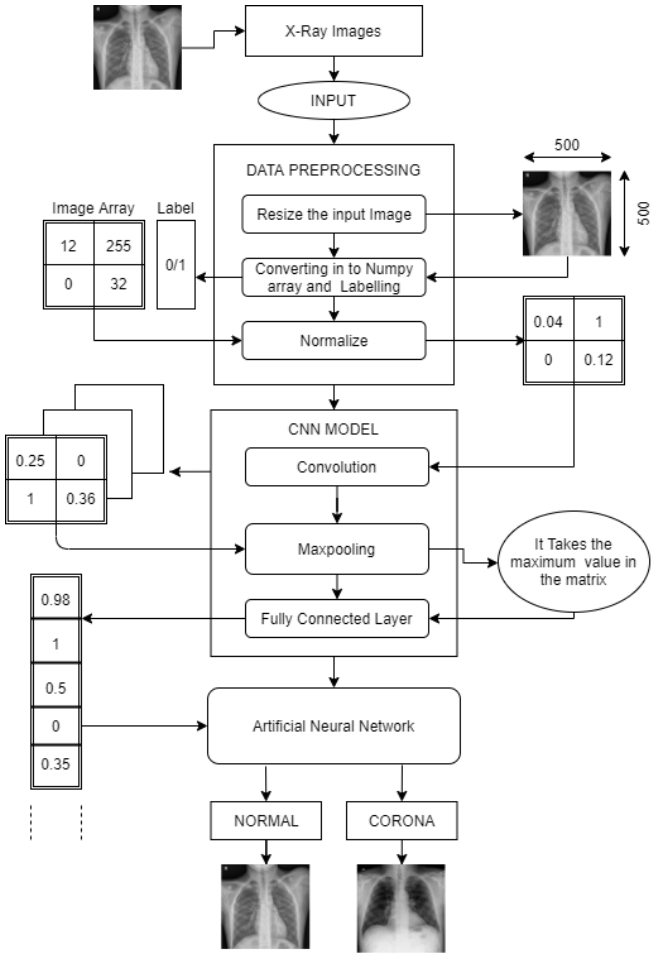


Fig. 2Neural Network Model

2.1 Data Pre-Processing:

Pre-processing refers to all the transformations on the raw data before it is fed to the machine learning or deep learning algorithm. For instance, training a convolutional neural network on raw images will probably lead to bad classification performances [6], hence the manipulation of the data, before training is a must.

In maximum of the cases all the input will not be of the same size hence it has to be resized, in our case we have resized the image to 500 x 500 pixels. Later on, the image will be converted to NumPy array and correspondingly labelled. Normalization is used to standardise the values from 0 to 1 in a converted NumPy array.

2.2 CNN model:

The 'convolutional neural network (CNN)' is a class of **deep learning neural network**. CNNs represent a huge breakthrough in image recognition. They're most commonly used to analyse visual imagery and are frequently working behind the scenes in image classification [8].

2.2.1 Convolution:

The main purpose of the convolution step is to extract features from the input image. In our particular case x-ray images are given as input, filters of suitable size are applied pixel by pixel block to the input image [8].The filter moves to the right with a certain Stride Value till it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same Stride Value and repeats the process until the entire image is traversed. [8]

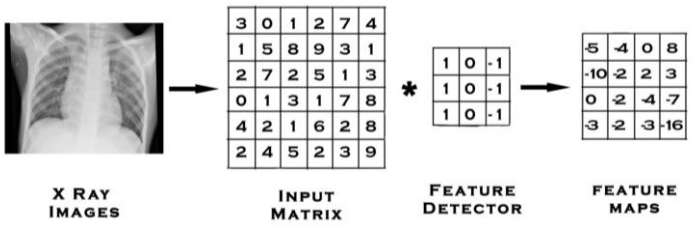


Fig. 3 Convolution

2.2.2 Relu (Activation function):

It is applied on the feature maps to increase non-linearity in the network. This is due to the high non-linearity of the images themselves. It removes negative values from an activation map by setting them to zero [8].

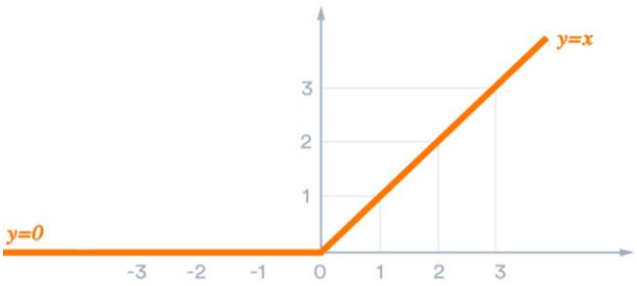


Fig. 4 Relu Function

2.2.3 Adam:

The choice of optimization algorithm is very essential to get an effective result. Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions is based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large interms of data and/or parameters [9]. Adam can be looked at as a combination of RMSprop and Stochastic Gradient Descent with momentum. It realizes the benefits of both AdaGrad and RMSProp. [13]

2.2.4 Max-pooling:

It is applied to grab the maximum values in the applied filter matrix. This reduces the size and parameters hence the model won't over fit on that information, if that was a case, we can go for dropout techniques to reduce the overfitting in the layers [8].

2.2.5 Fully Connected Network:

At this step, we add an artificial neural network to our convolutional neural network. The main purpose of the artificial neural network is to combine our features into more attributes. This combines features and attributes that can predict classes better. At this step, the error is calculated and then back propagated. The weights and feature detectors are adjusted to help optimize the performance of the model. Then the process

happens again and again. This is how our network trains on the data.Once the network has been trained, we can pass in an image and the neural network will be able to determine the image class probability for that image with a great deal of certainty [8].

3. Results and Discussion:

3.1 Summary of the model:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 166, 166, 32)      320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 83, 83, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 27, 27, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 5408)              0
_________________________________________________________________
dense (Dense)                (None, 128)               692352
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 710,306
Trainable params: 710,306
Non-trainable params: 0
```

Fig.5 Summary of the model

Fig. 5 shows the Summary of the CNN model with number of inputs, output shapes and number of parameters. Here Number of batches is None.

We have considered 2 convolution layers with each of 32 number of filters with a 3*3 feature detector and the input image shape of (500*500) pixels. Hence by considering the stride values equal to 3 gives the output resultant shape as (166*166). (i.e. Feature map shape).

3.1.1 Calculation:

If we consider the input matrix as n*n, feature matrix as f*f and stride as s then the output matrix size will be [(n-f)/s+1] * [(n-f)/s+1]

Note: Here 1 represents the number of bias

Output shape = ((500-3)/3) +1 = approx. (166).Number of Parameters = output channels * (input channels * window size + 1)

Therefore, Number of Parameters =32*((1*3*3) +1)
=320 Parameters

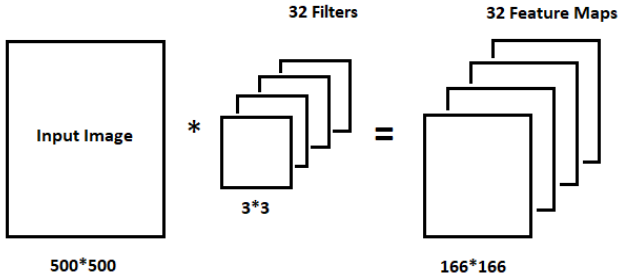


Fig.6: Output of first Convolution layer

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 12 • Issue 1 pp. 39-47 Jan - June 2020 www.csjournals.com

The second line in Fig.6 describes the features of max-pooling layer and the resultant output shape, when the pervious feature maps are passed through it.

Here in our model we have considered the max-pooling detector with (2*2) shape. Hence by considering the stride values equal to 2 gives the output resultant shape as (83*83).

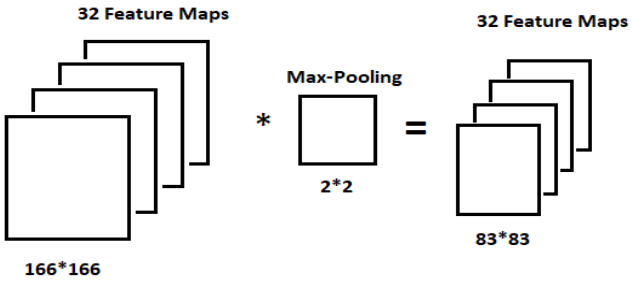


Fig.7 Output of Max-pooling layer.

Similarly, we have 2 convolution layers and correspondingly 2 max-pooling layers. Hence, the 3rd and 4th line in the Fig.5 describes the 2nd layer of convolution output shapes and number of parameters.
The number of Parameters is calculated and is equal to 9248
i.e. 32*((32*3*3) +1) =9248.

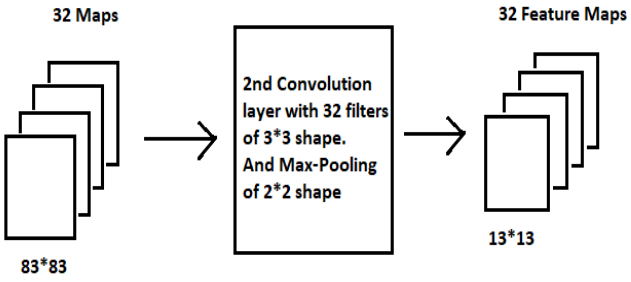


Fig.8 Output of 2nd Convolution and Max-pooling Layers.

In between the convolutional layer and the fully connected layer, there is a 'Flatten' layer. Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier. Hence, in this, a 2D matrix is converted in to 1D Row matrix of shape (5408,1) i.e. each pixel value is transposed into row matrix. Hence the shape is calculated by multiplying size of the feature maps and number of filters (13*13*32=5408).

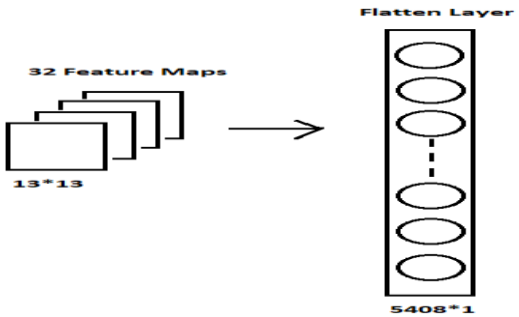


Fig.9 Flatten Layer

Finally, this layer will be given as the input to the ANN model with 2 dense layer of size 128 and 64 with one output layer with 2 nodes as the classified outputs.

International Journal of Electronics Engineering (ISSN: 0973-7383)
Volume 12 • Issue 1 pp. 39-47 Jan - June 2020 www.csjournals.com

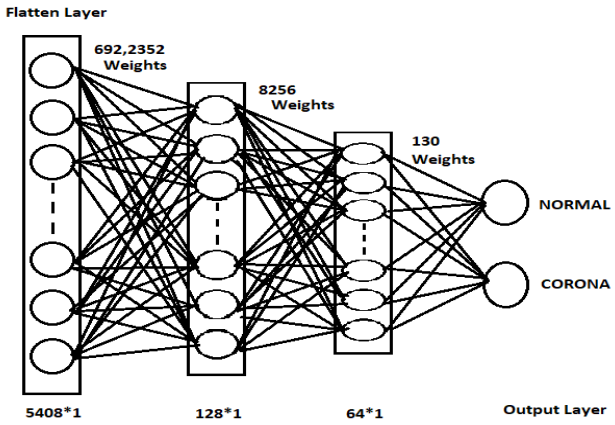


Fig 10. Fully Connected Layer

Hence, totally we have 710,306 number of parameters being trained with high accuracy.

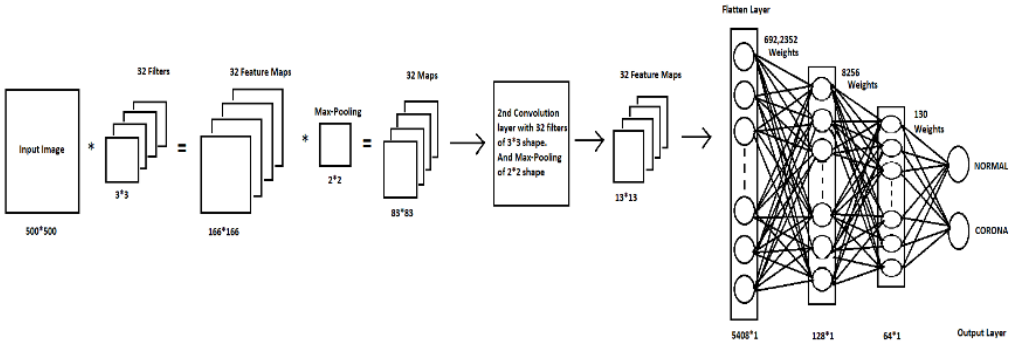


Fig 11. Complete CNN Model

3.2 Classification Metrics and Confusion Matrix:

There are two great methods to see how well our machine can predict or classify. One of them is the classification metrics and the other is the confusion matrix [10].

	precision	recall	f1-score	support
Corona	0.96	0.90	0.93	29
NORMAL	0.94	0.98	0.96	49
micro avg	0.95	0.95	0.95	78
macro avg	0.95	0.94	0.94	78
weighted avg	0.95	0.95	0.95	78
samples avg	0.95	0.95	0.95	78

Fig 12. Obtained Classification Metrics

Fig 12. Describes the parameters like precision, recall, f1-score and support with a different average value.These parameters can be easily calculated with the help of confusion matrix.

In every classification phase, there will be two different categories i.e.

1. True condition
2. Predicted Condition

That means we have 4 different possible ways to test the model ability,

a. Correctly classified to the category Normal
b. Correctly classified to the category Corona
c. Incorrectly classified to the category Normal

d. Incorrectly classified to the category Corona

The main point of confusion matrix and the classification metrics is basically a fundamental way of comparing the predicted values with true values [11].

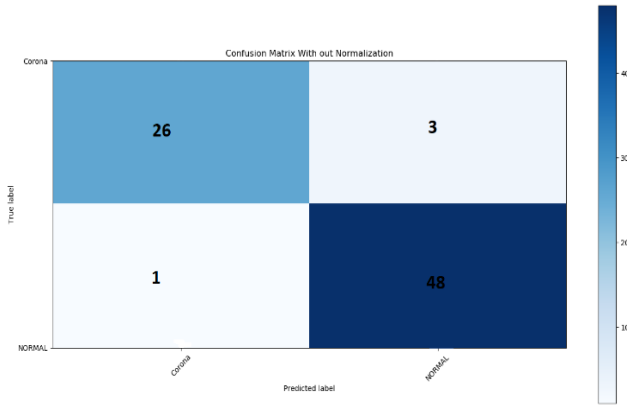


Fig 13. Confusion Matrix Without Normalization

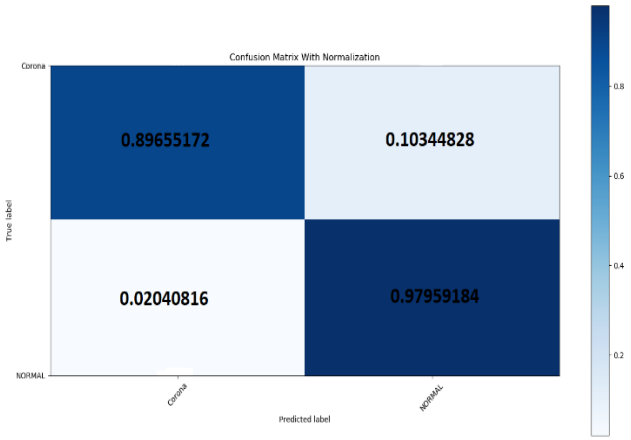


Fig 14. Confusion Matrix with Normalization

Fig. 13 and Fig. 14 show the confusion matrix without and with normalized value respectively.

From Fig. 13 ,we can derive 2 different conclusions i.e.

a. Out of 29 Predicted Positive samples 26 samples are correctly classified as corona and reaming 3 samples are wrongly classified as normal.
   Therefore, True Positive =26 and False Positive =3.
b. Similarly, out of 49 predicted negative samples 48 samples are being correctly classified as Normal and remaining 1 sample is not.
   Therefore, False negative=1 and True negative=48.

3.2.1 Calculations:

Precision: Precision talks about how precise/accurate the model is out of those predicted positive.
Therefore,

Precision= (26)/ (26+1)
= 0.96

Recall: Recall can be defined as the ratio of the total number of correctly classified positive examples divide to the total number of positive examples. High Recall indicates the class is correctly recognized (a small number of FN).[12]

Recall = (26) / (26+3)
= 0.896

F-measure: Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.[12]

F-measure= (2*0.9*0.96) / (0.9+0.96)

= 0.93

4. Results:

4.1Accuracy:

For the model to be successful it should get very good accuracy for the proper prediction of the images. In our practical scenario we got a training accuracy of around 99 percent (Train in fig.15) and testing accuracy of around 95 percent (Val in fig. 15) out of 15 epochs.

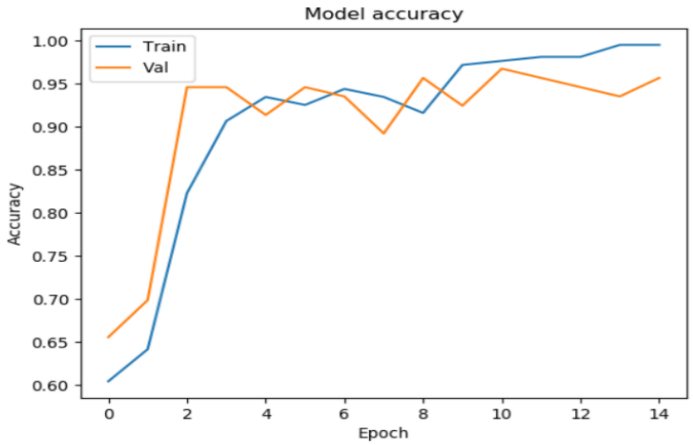


Fig.15 Accuracy

Fig.15 describes the increase in the accuracy of the model at every epoch out of 15 (from 0 to 14).Theoretically, Accuracy = ((True positive) + (True Negative))/ Total sum of samples in the confusion matrix.
Therefore, Accuracy= (26+48)/ (78) = approx. (0.95)

4.2 Loss:

Fig. 16 shows the decrease in the error or loss percentage at every epoch. We can see that the Training and Testing set loss percentage decreases approximately from 25 percent to 0.5 percent and hence we can conclude that as the accuracy increases, the loss percentage decreases resulting in a better model for the good prediction of the input. In our model we have used the "mean square error" function to predict the loss and we have used "Adam" function for optimization.[9]

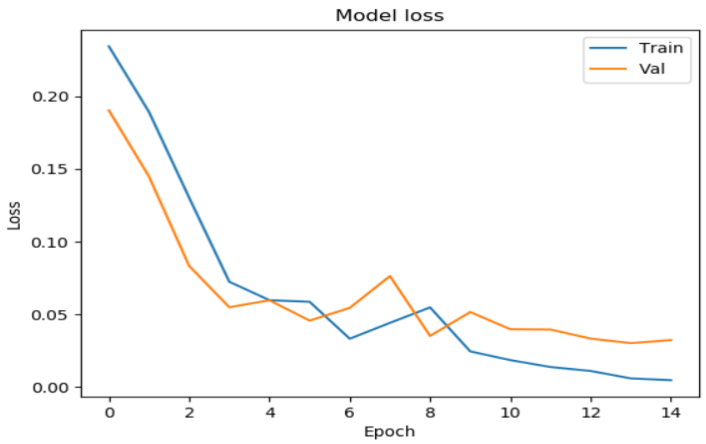


Fig.16 Loss

5. Conclusion:

The Corona virus pandemic has affected the lives of every being immensely, and developing a cure for this virus is a difficult task. The main task before curing a patient is the correct analysis of whether the person is affected by the virus or not. Thus, our model satisfies that primary objective or task by taking the x-ray scans of a patient's chest and analysing it to further classify it as affected or not. With a training accuracy of 99% and a testing accuracy of 95% our model has the efficiency in aiding the medical research facility in providing a faster way in confirming the prediction of a person being affected by the virus or not.

References:

1. Gozes, O., Frid-Adar, M., Greenspan, H., Browning, P. D., Zhang, H., Ji, W., Bernheim, A., and Siegel, E. Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis. ArXiv preprint arXiv: 2003.05037, 1-19, 2020.
2. Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks-2020
3. CNN News (https://edition.cnn.com/)-March 2020
4. "What to know about coronaviruses" an article written in
Medical news today. https://www.medicalnewstoday.com/articles/256521#covid-19).
5. https://www.nationalgeographic.com/science/2020/02/here-is-what-coronavirus-does-tothe-body/ 20.03.2020
6. https://www.freecodecamp.org/news/https-medium-com-hadrienj-preprocessing-for-deep-learning-9e2b9c75165c/
7. www.kaggle.com
8. "The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks and Image Classification". An article written in the medium platform- (https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb)
9. Diederik P. Kingma , Jimmy Lei Ba , "Adam : A Method for stochastic Optimization",ICLR 2015.
10. https://towardsdatascience.com/a-simple-cnn-multi-image-classifier-31c463324fa.
11. https://www.pieriandata.com/
12. Confusion matrix in machine learning by Greeks for Greeks
13. Adam — latest trends in deep learning optimization.Vitaly Bushaev on Oct 22, 2018."Gentle Introduction to the Adam Optimization Algorithm for Deep Learning" by Jason Brownlee on July 3, 2017 in Deep Learning Performance.