# COVID-19 DETECTION ON CT SCANS USING CNN

**By,**
**Amrutha Manikonda**
**Teja Chintha**
**Gnaneswari Lolugu**

## ABSTRACT

A virus that is having a disastrous impact on human life is COVID-19. This pandemic has had an impact on people's existence, well-being, and national economy. According to clinical research on COVID-19-infected patients, the majority of these patients get an infection of the lungs as a result of having contact with the disease. A Chest X-ray or Chest Computerized Tomography (CT) is a more reliable imaging method for identifying lung-related issues. In this study, we use CT scans of both COVID-19-affected patients and healthy patients. This study focuses on the classification of patients with CT Scan pictures who have been exposed to the coronavirus using four different Convolutional Neural Network architectures - Xception, InceptionV3, ResNet50, and VGG19.
**Keywords:** COVID, Convolutional Neural Network, Xception, InceptionV3, ResNet50, VGG19

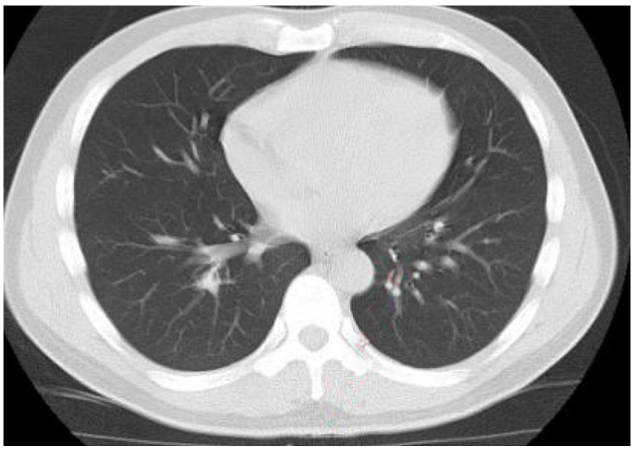
## INTRODUCTION

Coronavirus disease 2019 (COVID-19), which is brought on by the severe acute respiratory illness coronavirus, continues to have a devastating impact on the health and well-being of the global population (SARS-CoV-2). The condition was initially identified in December 2019 in Wuhan, China, and has since spread throughout the world. This condition has the potential to be extremely harmful. Nearly 2.49 million lives have been lost to COVID-19 as of February 2021, and these are only the confirmed cases. The price of diagnosing the illness is quite significant. Currently, genetic testing such as reverse transcription polymerase chain reactions makes up the majority of examinations. Although they are very expensive and take a long time to get findings, they are extremely accurate and can find even the slightest sign of an infection. This means that not all hospitals can afford to do it. Recent research demonstrates that viruses from this family can be easily identified on radiographic imaging. Therefore, CT scans are significantly quicker, more accurate, and less expensive than PCR tests.

Healthcare professionals could benefit from the application of AI in several areas of patient care and administrative tasks. Deep Learning (DL) is utilized to identify various issues, such as respiratory disorders from CT Scans. Due to its success in solving image-based classification and regression issues, DL has earned a special place in the history of AI. Image-based programming has been able to grow in popularity in recent years because of CNN. CNN is a neural network that is particularly proficient at extracting information from photos and requires very little pre-processing of the images before supplying them to the network. The best technique to detect COVID-19 is therefore using CT Scan images, which is what we did using four different CNN architectures.

## METHODS

Our dataset was downloaded from GitHub. The utility of our dataset has been verified, according to the description on Github, by a senior radiologist at Tongji Hospital in Wuhan, China, who has diagnosed and treated a significant number of COVID-19 patients during the pandemic of this disease between January and April. This dataset includes 397 CT images of patients who are in perfect health and 349 CT scans of patients whose problems have been detected using COVID. So, 746 photos will be used to train our models. 80% of the images will be used for training the models and the remaining 20% for testing the accuracy of the models.



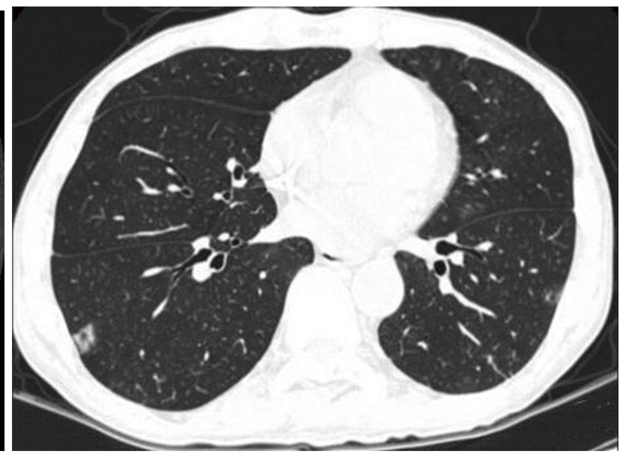


**Fig 1(a)** CT Scan of a patient with COVID

**Fig1(b)** CT Scan of a healthy patient

### VGG19 Architecture

The ImageNet database's VGG-19 convolutional neural network was trained using more than a million photos. The 19-layer network can categorize photos into 1000 different object categories, including a keyboard, mouse, pencil, and numerous animals. The network has therefore acquired rich feature representations for a variety of images.

The mean RGB value of each pixel, calculated throughout the whole training set, was the only preprocessing that was carried out. This network received a fixed-size (224 * 224) RGB picture as input, indicating that the matrix was shaped (224,224,3). The mean RGB value of each pixel, calculated throughout the whole training set, was the only preprocessing that was carried out. They were able to cover the entirety of the image by using kernels that were (3 * 3) in size with a stride size of 1 pixel. To maintain the image's spatial resolution, spatial padding was applied. Max pooling was carried out with stride 2 over 2 * 2-pixel windows. Following this, a Rectified linear unit (ReLu) was used to add non-linearity to the model to increase classification and computation time. This was done because earlier models had relied on tanh or sigmoid functions, which had proven to be considerably more effective. Three completely connected layers were implemented, the first two of which had a size of 4096. The final layer is a Softmax function which has 1000 channels.

### ResNet50 Architecture

A ResNet model version called ResNet50 contains 48 Convolution layers, 1 MaxPool layer, and 1 Average Pool layer. ResNet50 contains a convolution with 64 different kernels, each having a stride of size 2, and a kernel size of 7 * 7, giving us 1 layer. Following that, we witness max pooling with a stride size of 2. The following convolution consists of three layers: a 1*1,64 kernel, a 3*3,64 kernel, and a 1*1,256 kernel. These three levels have been repeated a total of three times, giving us nine layers in this phase. The kernel of 1 * 1,128 is next, followed by a kernel of 3 * 3,128, and finally a kernel of 1 * 1,512; this phase was repeated for 12 layers. Following that, we have a kernel of size 1*1,256, followed by two more kernels of size 3*3,256 and 1*1,1024; this is repeated six times, giving us a total of 18 layers. Finally, a 1 * 1,512 kernel was added, followed by two more kernels of 3 * 3,512 and 1 * 1,2048. This process was done three times, giving us a total of nine layers. Then, we perform an average pool, finishing it with a fully connected layer that has 1000 nodes and a softmax function so that this.

**InceptionV3 Architecture**

The third iteration of Google's DL convolutional architectures is called Inception V3. A CNN tool called Inception V3 aids in the analysis of the image and item finding. The major goal is to reduce the amount of processing power used by enhancing and altering earlier Inception architectural designs. It was initially honed using a 1000-class original ImageNet dataset. More than a million training photos were used to train. The RMSProp Optimizer in Inception V3 is factorized 7x7. The auxiliary classifiers' BatchNorm, convolutions, and Label Smoothing (a regularizing element that is included in the loss formula to prevent overfitting and the network from being overconfident about a class. InceptionV3 has 42 layers.

**Xception Architecture**

Depthwise Separable Convolutions are used in the deep convolutional neural network architecture known as Xception. The "extreme" version of an Inception module is referred to as Xception. Xception, which means "extreme inception," pushes the fundamental ideas of Inception to the limit. In Xception, the filters are first applied to each depth map individually, and only then is the input space compressed using 1X1 convolution by applying it across the depth. This process is nearly equivalent to a depthwise separable convolution, a technique that was first applied to the building of neural networks in 2014. Between Inception and Xception, there is yet another distinction. whether a non-linearity exists or not after the initial operation. Xception is a deep CNN design that does not introduce any non-linearity.

We have built four CNN models, namely: VGG19, ResNet50, InceptionV3, and Xception. We decided to add 3 custom layers (Flatten, Dropout & Dense) to the pre-trained models so that the models mentioned above can be trained well according to our dataset. We chose to use pre-trained weights from ImageNet in all four models. The fully connected (FC) output layers of the model, which are used to produce predictions, are not included because the included top parameter has been set to false. When the included top is set to false, the final convolution layer or the pooling layer is used to derive the output activations. The images in the dataset are of different sizes. Then, a tensor of shape (224, 224, 3), where 3 is a number, was added. 224 x 224 is the size of the image after being scaled, and of channels. Softmax was employed as the activation function. We used Image Data Generator to train the models at modified versions of the images, such as at different angles, flips, rotations, or shifts. We trained the models with 500 epochs and 32 images for each batch size.

## RESULTS

The outcomes of the CNN models that we examined are shown below. The percentage of COVID in a certain patient is computed using the results and is displayed. As a result, we made predictions for each model using the entire dataset.

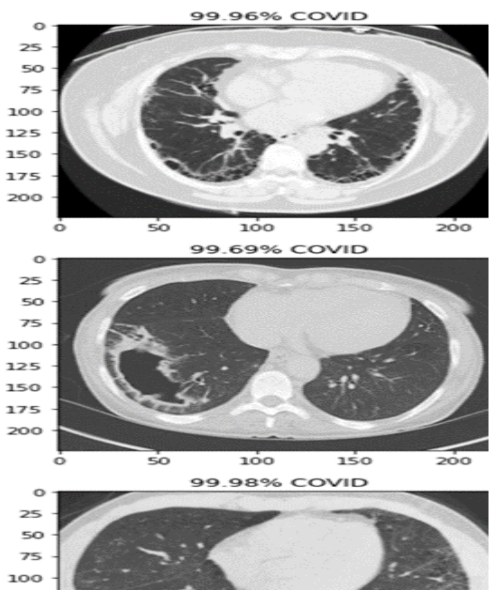



**Fig 2(a)** Prediction visualizations of VGG19

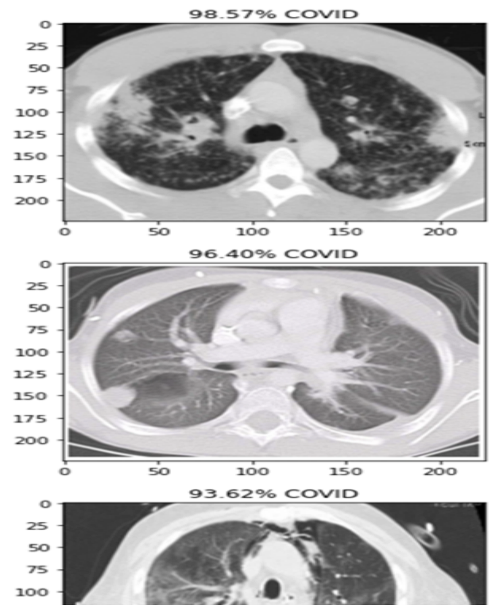



**Fig 2(b)** Prediction visualizations of ResNet50

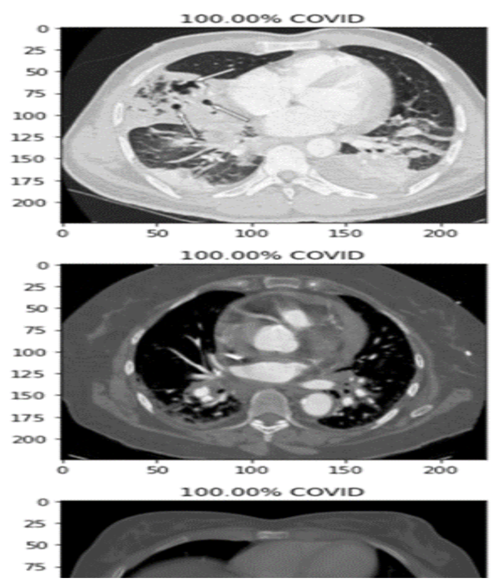



**Fig 2(c)** Prediction visualizations of InceptionV3

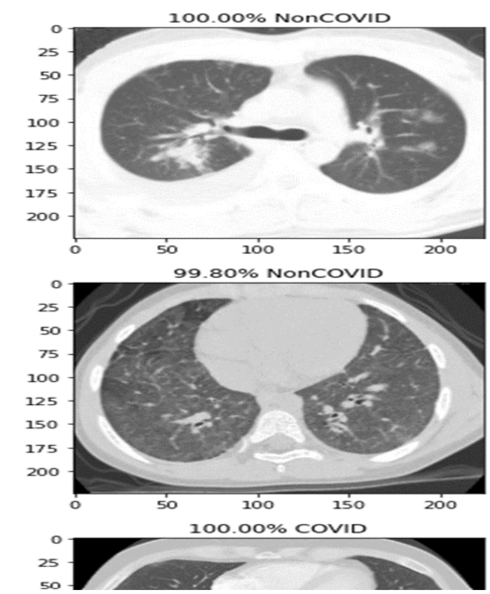



**Fig 2(d)** Prediction visualizations of Xception

The classification reports for the four models are provided below. They include metrics like Accuracy, Support, Recall, F1-score, and Precision. As we can see, Xception, InceptionV3, ResNet50, and VGG19 have accuracy rates of 79%, 76%, 79%, and 80%, respectively. We can determine the precise percentage of predictions that were accurate. With a macro average of 0.82, VGG19 has the highest precision value, followed by Xception and ResNet50, which has a precision rating of 0.79. Recall allows us to calculate the proportion of correctly identified positives. The weighted average of recall for VGG19 is the highest at 0.80.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.87 | 0.67 | 0.76 | 70 |
| 1 | 0.76 | 0.91 | 0.83 | 80 |
| accuracy | | | 0.80 | 150 |
| macro avg | 0.82 | 0.79 | 0.79 | 150 |
| weighted avg | 0.81 | 0.80 | 0.80 | 150 |

**Fig 3(a)** Classification Report of VGG19

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.78 | 0.76 | 0.77 | 70 |
| 1 | 0.79 | 0.81 | 0.80 | 80 |
| accuracy | | | 0.79 | 150 |
| macro avg | 0.79 | 0.78 | 0.79 | 150 |
| weighted avg | 0.79 | 0.79 | 0.79 | 150 |

**Fig 3(b)** Classification Report of ResNet50

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.71 | 0.83 | 0.76 | 70 |
| 1 | 0.82 | 0.70 | 0.76 | 80 |
| accuracy | | | 0.76 | 150 |
| macro avg | 0.77 | 0.76 | 0.76 | 150 |
| weighted avg | 0.77 | 0.76 | 0.76 | 150 |

**Fig 3(c)** Classification Report of InceptionV3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.76 | 0.80 | 0.78 | 70 |
| 1 | 0.82 | 0.78 | 0.79 | 80 |
| accuracy | | | 0.79 | 150 |
| macro avg | 0.79 | 0.79 | 0.79 | 150 |
| weighted avg | 0.79 | 0.79 | 0.79 | 150 |

**Fig 3(b)** Classification Report of Xception

## DISCUSSION

In this study, we examined four CNN models to categorize COVID-19 patients based on their lung CT scans. With 80% accuracy, our VGG19 model had the best performance. Xception and ResNet50 achieved a 79% accuracy. 76% accuracy was attained by the InceptionV3 model. The best outcomes came from the VGG19 model. Its weight serialization is the smallest. ResNet50 includes numerous layers as well but uses less memory because we use average pooling rather than completely connected layers. The organizational complexity of InceptionV3 is moderate. To improve its performance, we need to comprehend it even better. We've finally categorized COVID-19 scans, and this demonstrates the potential for automated diagnosis in the future.

## References


[1] https://iq.opengenus.org/resnet50-architecture/

[2]https://iq.opengenus.org/vgg19-architecture/#:~:text=VGG19%20is%20a%20variant%20of,VGG19%20has%2019.6%20billion%20FLOPs.

[3]https://iq.opengenus.org/inception-v3-model-architecture/

[4] https://iq.opengenus.org/xception-model/

[5]https://ieeexplore.ieee.org/document/9432134

[6] https://www.geeksforgeeks.org/detecting-covid-19-from-chest-x-ray-images-using-cnn/

[7]https://www.kaggle.com/code/vafaknm/covid-detection-by-ct-scan-images-using-cnn?scriptVersionId=104246511

[8]https://github.com/UCSD-AI4H/COVID-CT