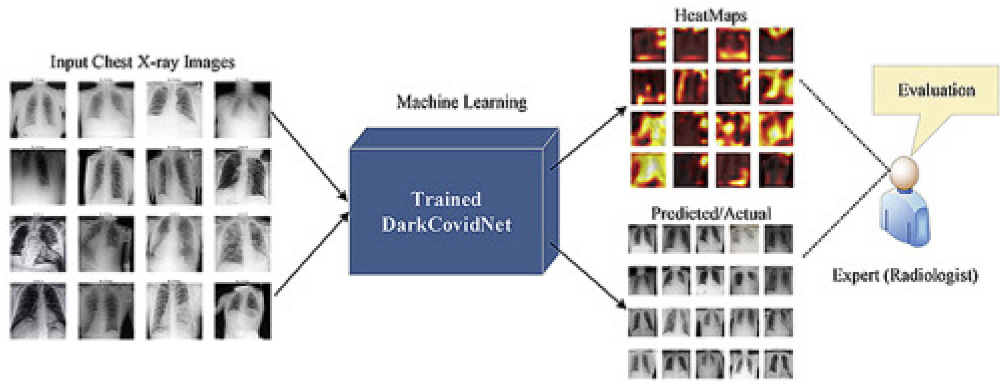
Rajvi Shah

Apr 10 · 4 min read · Listen



# Detection of COVID — 19 using Deep Learning

Numerous ways to detect COVID — 19 using Deep Learning.

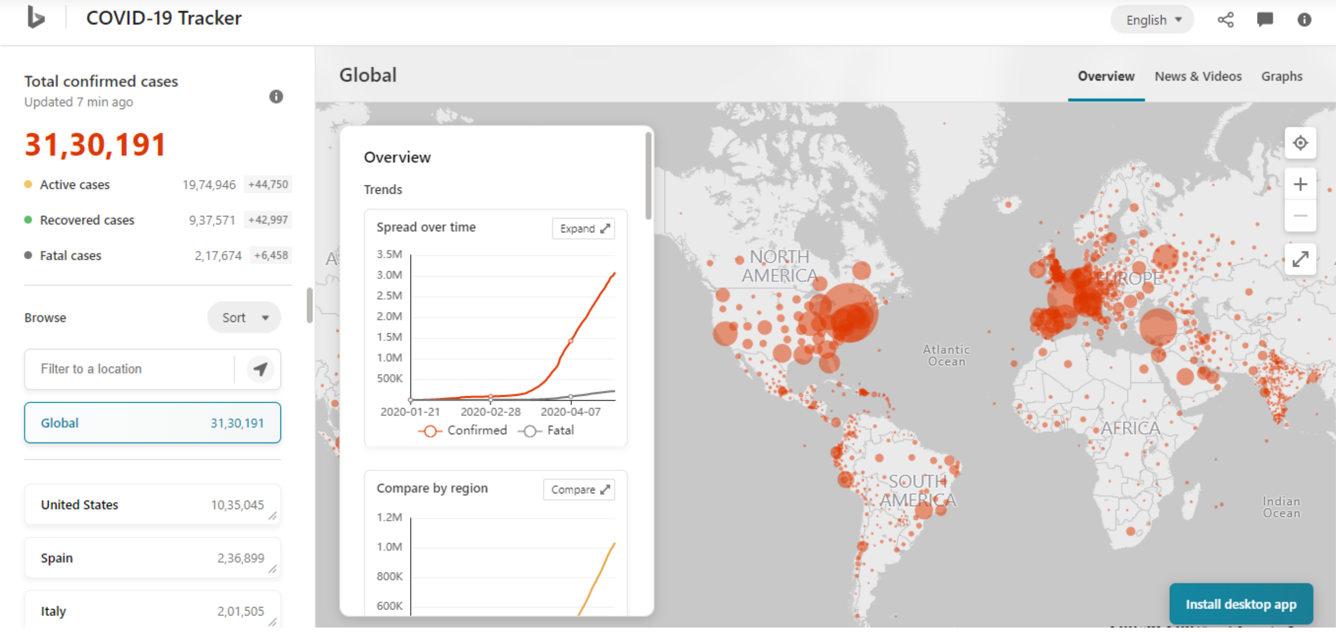



Source: https://www.sciencedirect.com/science/article/pii/S0010482520301621

"**Coronavirus disease 2019** (**COVID-19**) is a highly infectious disease caused by *severe acute respiratory syndrome coronavirus 2*".

"The disease first originated in December 2019 from *Wuhan, China* and since then it has spread globally across the world affecting more than **200 countries**. The impact is such that the ***World Health Organization(WHO)*** has declared the ongoing pandemic of **COVID-19** a Public Health Emergency of International Concern."

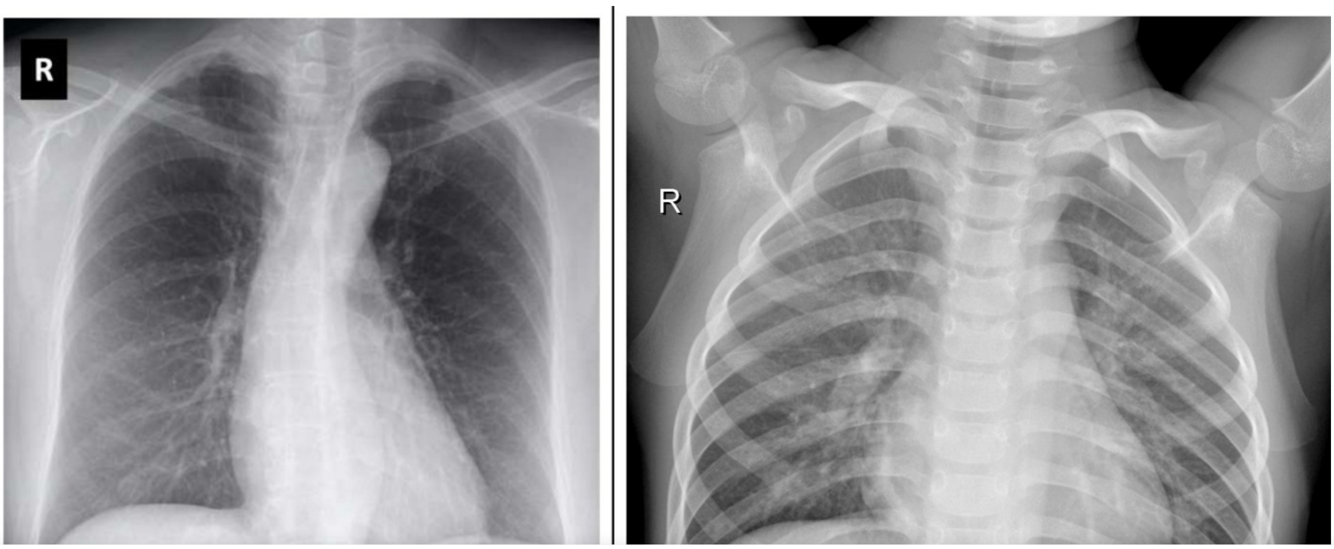
As of 29th April, there are a total of 31,30,191 cases with 2,17,674 deaths in more than 200 countries across the world.




So, in this particular scenario, one primary thing that needs to be done and has already started in the majority of the countries is *Manual testing*, so that the true situation can be understood and appropriate decisions can be taken.

But the drawbacks of manual testing include sparse availability of testing kits, costly and inefficient blood tests; a blood test takes around 5–6 hours to generate the result.

So, the idea is to overcome these circumstances using the Deep Learning technique for better and efficient treatment. Since the disease is highly contagious therefore as early as we generate the results the fewer cases in the city that's why we can use Convolution Neural Network to get our job done.



Can you distinguish between both X-rays if they haven't been labeled? I bet you can't, but a CNN can.

The problem is a binary classification problem where we classify Normal vs COVID-19 cases.

There are several pros and cons of using Deep Learning to tackle such kinds of situations:

1. Pros: More time saving; less expensive; easy to operate

2. Cons: Practically we need ~100% accuracy as we can't wrongly identify the patients as it might lead to further spread of disease which is highly discouraged.

But still, this model can return good accuracies and can further be enhanced.

## Dataset

- The dataset used is obtained from a GitHub repository owned by UCSD-AI4H.

- The dataset consists of 349 images of chest CT-Scan from infected patients (labeled as pos, indicating "positive") and 397 images from non-infected patients (labeled as neg, indicating "negative").

- The dataset is split into two categories, the training dataset, and the testing dataset.

- The training dataset is used to train the deep learning model while the testing dataset is used to test the model's performance after the training process can be seen in diagrams given below.

- The images from the training dataset are resized to a 1:1 ratio.

- Then every one of the images is rotated 10° to the right, 10° to the left, 20° to the right, and 20° to the left, thus generating four new images.

- This expands the dataset size to five times bigger before the addition of rotated images.

- The sum of images in the testing dataset is 74.

*A. Samples of the original dataset*

Original Dataset

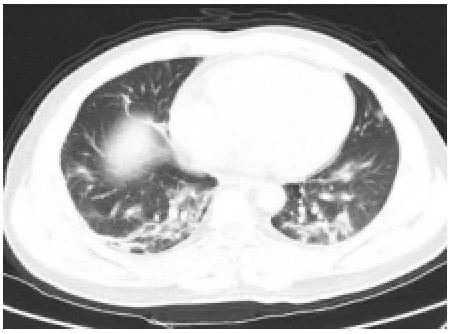


Fig. 1. A sample image from the "pos" label

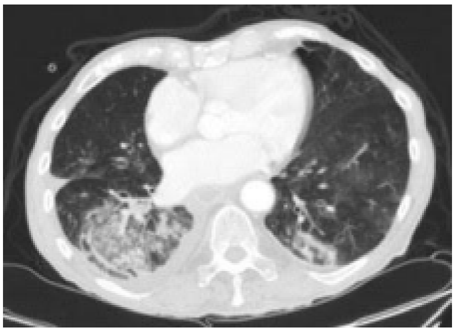


Fig. 2. A sample image from the "neg" label

*B. Samples of the pre-processed dataset*

Rotated Dataset

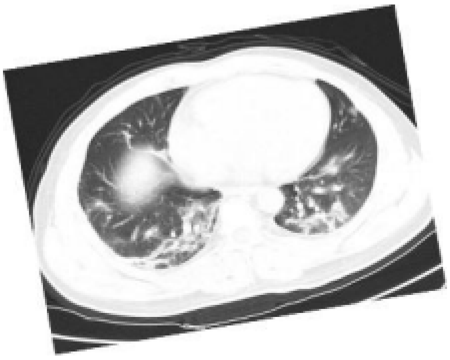


Fig. 3. A sample "pos" image rotated -10° to the left

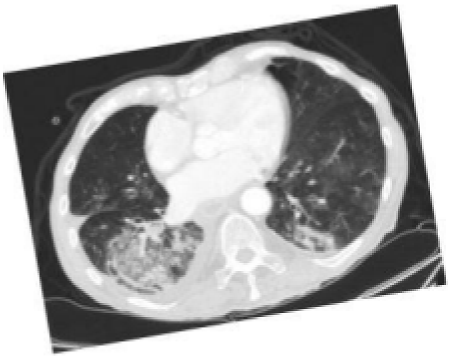


Fig. 4. A sample "neg" image rotated -10° to the left

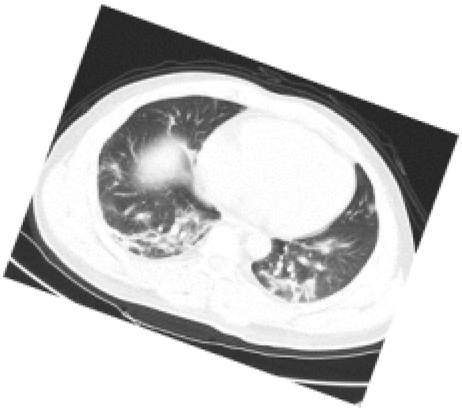


Fig. 5. A sample "pos" image rotated 20° to the right

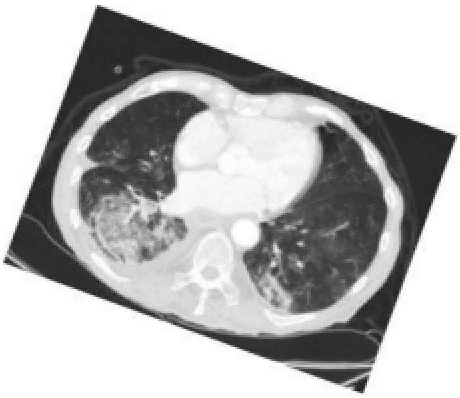


Fig. 6. A sample "neg" image rotated 20° to the right

## Methodology

- The proposed model named FJCovNet is based on the DenseNet 121 pretrained model.
- The application of transfer learning was used in building FJCovNet, which is excluding the top layer of the DenseNet 121 pretrained model.
- This enables the pretrained model to work as a feature extractor but yet for classifying.
- To enable the classifying function to our model, we add layers on top of the DenseNet121 layer, which works as the classification layers, to perform classification.
- These layers are Batch Normalization, Global Average Pooling, Dense, and Dropout.
- The paper used the Batch Normalization layer for its great benefits for increasing the model's performance.
- Batch Normalization resets the parameters of the optimization problem of the model and makes it more stable and smoother.
- This leads to faster and more effective optimization.

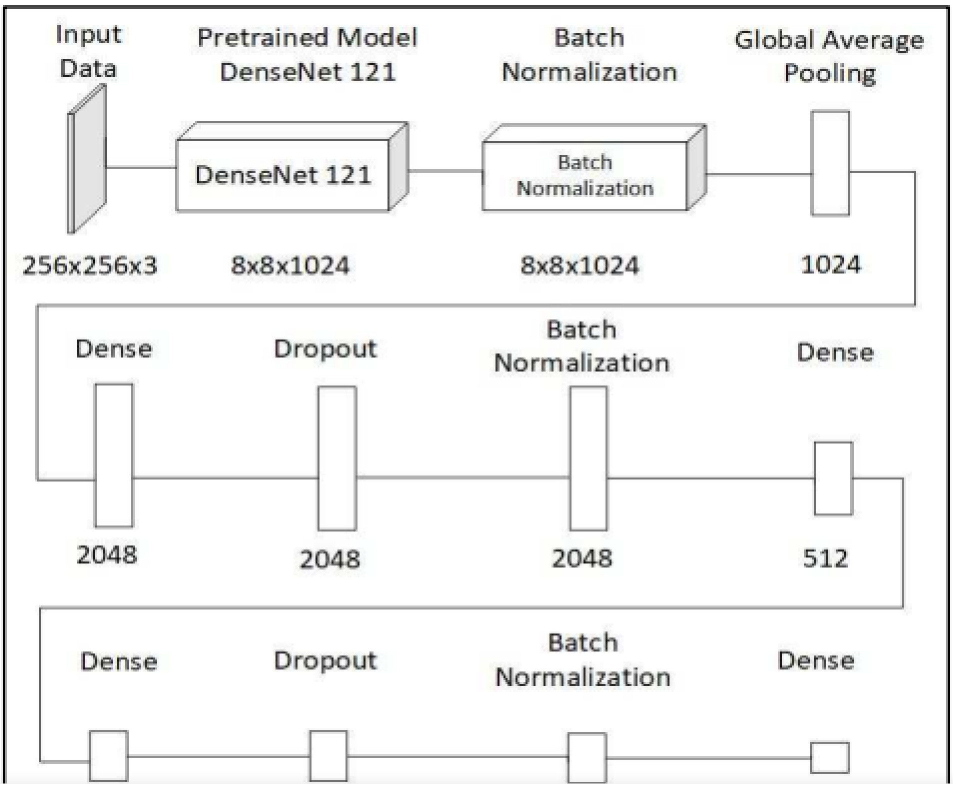
- The way the dropout layer works is by randomly dropping units, along with the network connection during the training process.
- This reduces overfitting significantly, thereby enhancing the model's performance and increasing accuracy.
- The hyperparameters we set for building the FJCovNet are described in Table I.

TABLE I. HYPERPARAMETERS USED IN FJCOVNET

| | |
|---|---|
| Batch size | 64 |
| Epochs | 100 |
| Optimizer | Adam |
| Loss function | Sparse Categorical Crossentropy |
| Step per epoch | 37 |

Source: https://ieeexplore.ieee.org/document/9431887

## Proposed Model



## Results

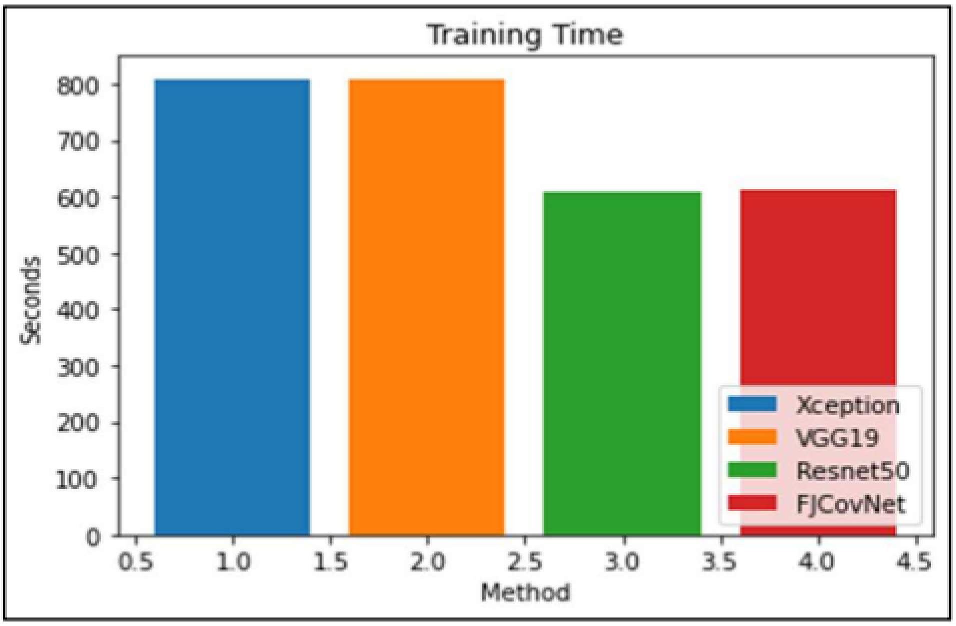



Fig. 12. Comparison of Training Times

Source: https://ieeexplore.ieee.org/document/9431887

TABLE II. COMPARISON VALUES OF FJCOVNET AND OTHER MODELS

| Model | Training Time (in seconds) | Validation Accuracy | Validation Loss |
|---|---|---|---|
| Xception | 809 | 84.24% | 0.7687 |
| VGG19 | 808 | 95.25% | 0.1644 |
| Resnet50 | 609 | 91.53% | 0.3814 |
| FJCovNet | 612 | 98.14% | 0.1556 |

Source: https://ieeexplore.ieee.org/document/9431887

## References

- https://ieeexplore.ieee.org/document/9431887
- https://towardsdatascience.com/detecting-covid-19-using-deep-learning-262956b6f981

## Get an email whenever Rajvi Shah publishes.

You cannot subscribe to yourself