Detection of COVID 19 from chest CT scan and X-Ray Images using Transfer learning and Stacking

Aaron Antonio Dias Barreto

September 2020

Abstract

Early detection of covid 19 is cruicial to reduce mortality rate. One of the sources for early detection of covid is by analzing Chest CT scan of an individual. This paper aims to use deep learning, transfer learning, and stacking to build a model that can be used to detect COVID 19 quickly and efficiently. The coronavirus is spreading rapidly and is threatening the lives and livelihoods of many people. The current methods of testing are time-consuming, expensive, and also produce false positives. In this paper, a deep learning model is proposed that uses pre-trained computer vision models and the concept of stacking. The proposed model achieves uniformly good performance on five different datasets which consist of CT scans as well as chest X-Ray images.

Contents

1 Introduction

1.1 Coronavirus

COVID 19 short for Coronavirus Disease 2019 is a life-threatening disease caused by the severe acute respiratory syndrome coronavirus (SARS-CoV-2). It is a disease for which currently there is no cure and it is spreading like wildfire throughout the world, the worst-hit countries being the United States of America, Brazil, and India. This virus was first detected in Wuhan, China from where it spread to the rest of the world. As of September 1st, 2020, the total number of cases worldwide was 25,803,273 and the number of deaths was 857,752. The most common symptoms of this disease include cough, fever, breathing difficulties, and loss of taste and smell. The most dangerous aspect of this virus is that most of the infected people are asymptomatic at first, and during this time may spread the virus unknowingly to the people around them. Therefore the quick detection of these infected people especially in the early stages is of paramount importance. The current method of diagnosis is by real-time reverse transcription-polymerase chain reaction (rRT-PCR) from a nasopharyngeal swab. However the results of this test take about a day or two to arrive, during which the infected person could spread this disease, therefore more efficient and faster testing methods need to be developed. Another way to detect this virus is from CT scans or Chest X-Rays of patients, however, this is also a time-consuming process as an expert is needed to read these CT scans and Chest X-Rays to determine if a person is COVID positive. Therefore a method using Deep Learning is proposed in this paper through which this virus could be detected quickly and efficiently.

1.2 Deep Learning

Deep learning is a part of machine learning that deals with algorithms that are based on the structure of the human brain called neural networks. Neural networks can be hundreds of layers deep and can contain about a million parameters. Neural networks can achieve almost human-level performance on many tasks and their performance only increases with the amount of data they are given. There are various deep learning models used specifically for image detection, these models are called Convolutional Neural Networks (CNNs). The advantages of these CNNs over traditional neural networks are that they make use of parameter sharing and therefore use relatively fewer parameters to get the same or even better performance as compared to a traditional neural network, thus saving both space and time.

1.3 Transfer Learning

Transfer learning is a deep learning technique which uses a deep learning model trained to do a specific task, to perform another related task. The parameters of the original model are fined tuned to the second task. The advantages of using transfer learning are that it saves a lot of time as training a model from scratch would take a lot longer than using a pre-trained model and just fine-tuning to the given task, also the data required to fine-tune a model is far less than the data used to train a model from scratch. Transfer learning is used from task A to task B when A and B have the same input type, images, in this case, and the amount of data available for task B is less as compared to task A.

1.4 Stacking

Stacking is an ensemble machine learning algorithm that learns how to combine the outputs of two or more models to produce the correct output. Stacking improves the overall performance of the model. The outputs of the base models are used as inputs to the meta-model. The meta-model uses these inputs to produce the correct output. The base models used in stacking generally have different network architectures, if the models used have the same network architecture with only little variation in the hyperparameters, then their predictions will also be relatively similar and stacking might not be able to achieve better performance as compared to the base models.

1.5 VGG 19

VGG 19 is a deep learning model used for image classification which consists of 19 layers (16 convolutional layers, 3 fully connected layers, 5 MaxPool layers, and 1 SoftMax layer) and has almost 144 million parameters.

1.6 ResNet 101

ResNet 101 is a deep learning model used for image classification it consists of 101 layers and almost 45 million parameters. The distinctive feature of the ResNet architecture is that it has shortcut or skip connections that are present in a Residual block. ResNet architectures consist of many of these residual blocks connected in series.

1.7 DenseNet 169

DenseNet 169 is a deep learning model used for image classification, it consists of 169 layers. The DenseNet model consists of Dense blocks, each layer in a Dense block is connected to all subsequent layers in that block.

1.8 Wide ResNet 50 2

Wide ResNet 50 2 is a deep learning model used for image classification, it is a modified version of the ResNet model, it has a depth of 50 and a width of 2. It has almost 69 million parameters.

2 Literature Review

Ever since the outbreak of the coronavirus, a quick and efficient method for its detection has been and still is a major research area. Work by Xiaowei Xu et al showed that there is evidence that the coronavirus disease can be detected from CT scans and Chest X-Rays as the lungs of the infected people are affected by this virus, the model they proposed achieved an accuracy of 0.87 on the dataset that they used. However, due to privacy reasons, the CT scans and Chest X-Ray images of COVID positive patients are not publicly available and therefore building models for the detection of COVID 19 was not an easy task. One of the first publicly available datasets was by Xingyi Yang et al. In their paper they demonstrated the usefulness of this dataset and also proposed a deep learning model that was able to achieve an accuracy of 0.89 and an F1 score of 0.90. Another dataset that was made publicly available was the COVIDx Dataset created by Linda Wang et al. when they published the COVID-Net paper, their proposed model was able to achieve an accuracy of 0.93. Muhammad Farooq and Abdul Hafeez in their paper COVID-ResNet were able to further improve the performance on this dataset, their proposed model achieved an accuracy of 0.96.

Therefore there is evidence that CT scans and Chest X-Rays can be used to detect COVID 19. The models proposed use Convolutional Neural Networks (CNNs) as they have been very successful at many computer vision and biomedical imaging tasks.

Most of the models proposed have been trained and evaluated on only one dataset, in this paper the proposed model has been evaluated using five different datasets consisting of both CT scan images and Chest X-Ray images.

3 Proposed Model

This section discusses the proposed model. The model consists of three parts. The first part uses a pre-trained VGG 19 model and 3 fully connected layers. The VGG 19 model maps the input volume of size (3 X 224 X 224) to a column vector, consisting of 1000 rows. The first fully connected layer converts this column vector into a column vector of 500 rows. The second fully connected layer further reduces this column vector into a column vector which has 200 rows. The last fully-connected layer reduces this column vector into a column vector with as many rows as the number of classes (which is 2). The first two fully connected layers use a ReLU activation function, while the last fully connected layer uses a softmax activation function. A dropout layer with a dropout probability of 0.5 is applied between each of the fully connected layers, to prevent the model from overfitting to the training data.

The second part uses a pre-trained DenseNet 169 model and one fully connected layer. The DenseNet 169 model maps the input volume of size 3 X 224 X 224 to a column vector, consisting of 1000 rows, just like the VGG model. The fully connected layer maps this column vector to a column vector with 2 rows (equal to the number of classes). The fully connected layer uses a softmax activation function. This part also uses a dropout layer with a probability of 0.5.

The third part uses a pre-trained DenseNet 169 model and 3 fully connected layers. The third part is identical to the first part except that it uses the DenseNet 169 model instead of the VGG 19 model.

Finally, the outputs of each of the 3 parts are put through a single neuron to get the predicted class. This single neuron uses a softmax activation function.

The proposed model uses transfer learning so that the model can train faster. The weights of the pre-trained models are fine-tuned to the task at hand, which is detecting COVID 19. The three models are combined using stacking to predict the output class. In this model, a single neuron is used as the meta-model, which correctly predicts the output class based on the outputs of the three models discussed above.

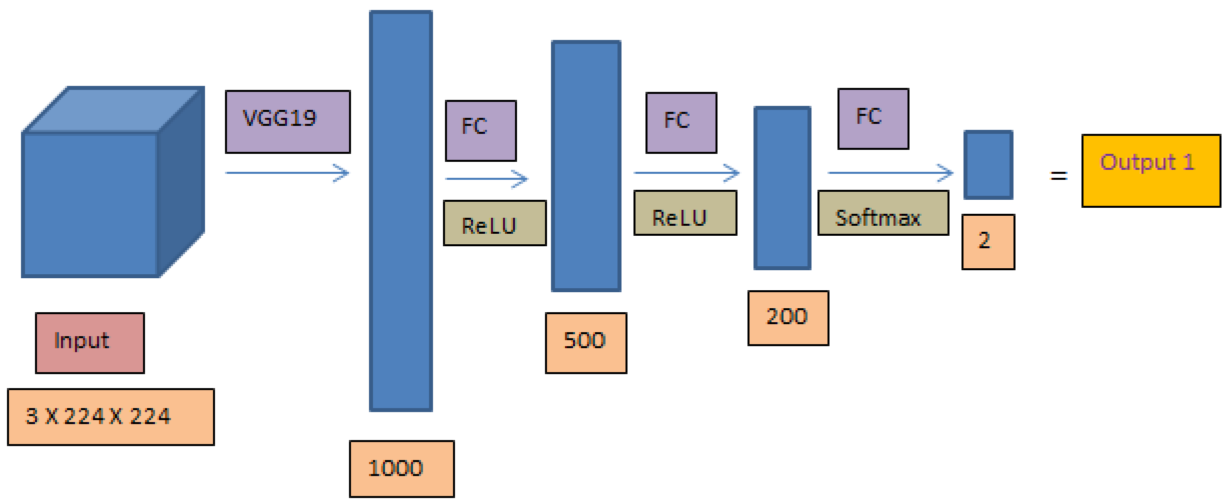


Figure 1: Part 1 Model Architecture

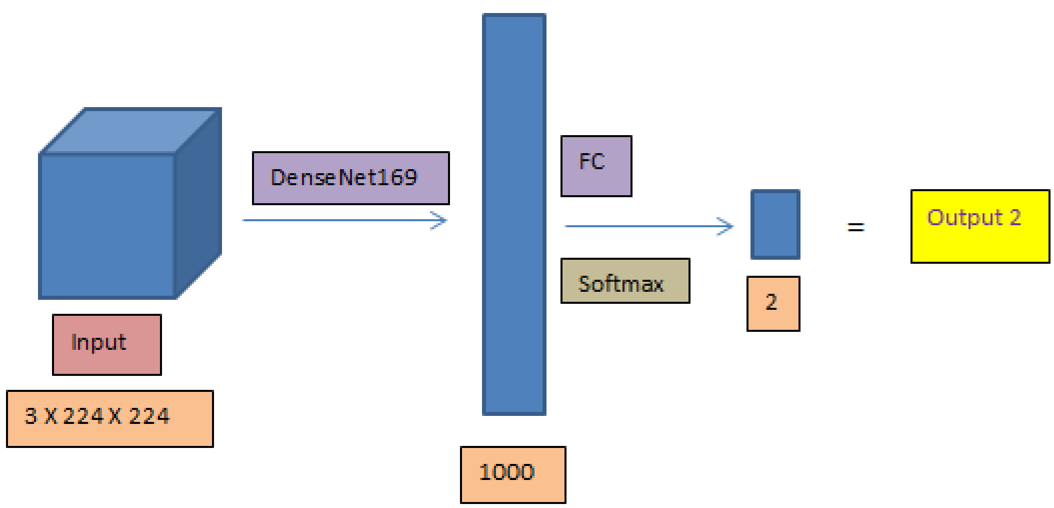


Figure 2: Part 2 Model Architecture

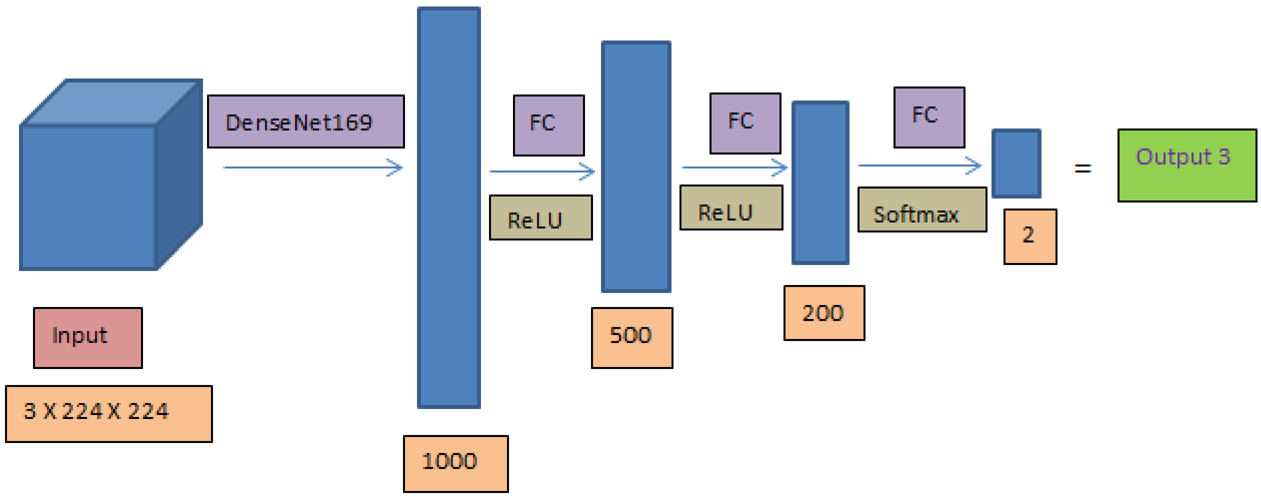


Figure 3: Part 3 Model Architecture

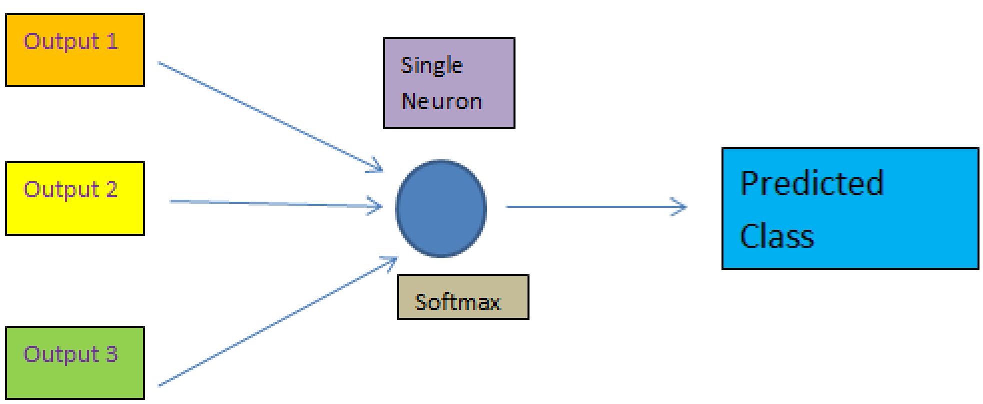


Figure 4: Combined Model Architecture

4 Data Augmentation Techniques

The data augmentation techniques used are:

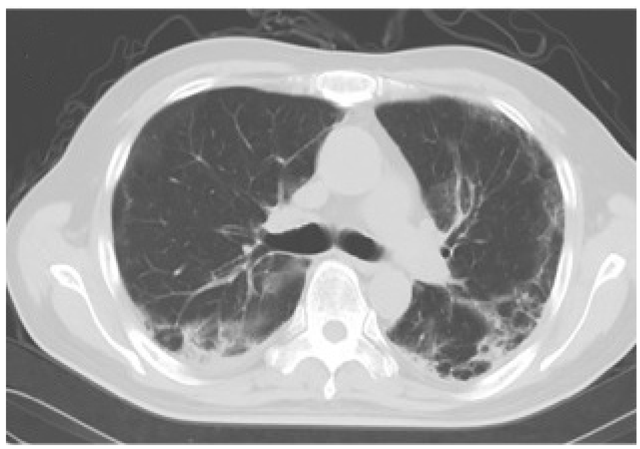
- Random Resized Crop: Crop the given image to a random size and aspect ratio.
- Random Rotation: Randomly rotate the image by an angle in the given range.
- Random Horizontal Flip: Horizontally flip the given image randomly with a given probability.
- Colour Jittering: Randomly change the brightness, contrast, and saturation of an image.

5 Training Details

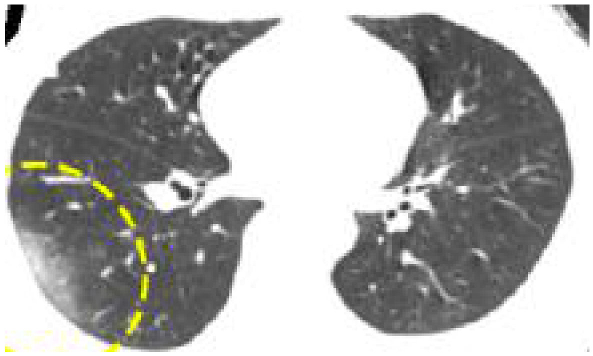
- Deep Learning Framework = PyTorch
- Number of epochs = 100
- Optimizer = Adam
- Learning rate = 1e-3
- Loss function = Cross Entropy Loss
- Batch size = 16
- Random Resized Crop size = 224
- Random Resized Crop Scale = (0.5, 1.0)
- Random Rotation angle range = [-5 degrees, 5 degrees]
- Random Horizontal Flip probability = 0.5

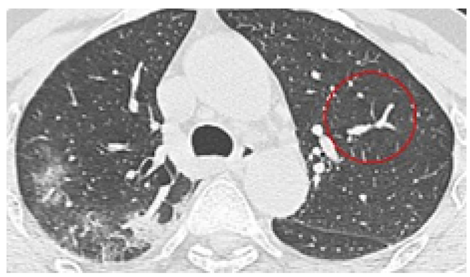
6 Images from the datasets

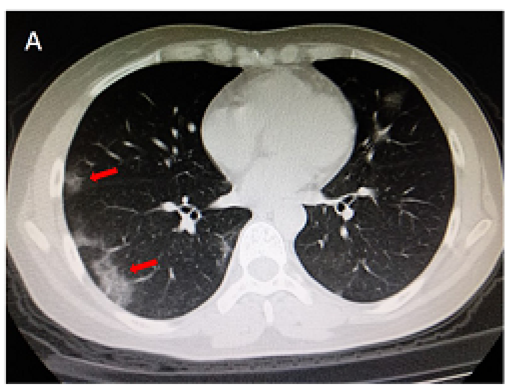
6.1 COVID Negative CT scan images



6.2 COVID Positive CT scan images







7 Experimental Results

7.1 Datasets

- Dataset 1
  - Source: https://github.com/UCSD-AI4H/COVID-CT
  - Type of images: CT scans
  - Dataset size: 746 images
  - No. of COVID positive images: 349
  - No. of COVID negative images: 397
  - Train set size: 425 images
  - Validation set size: 118 images
  - Test set size: 203 images

  Note: To prevent the model from overfitting to the training data, the size of the training set is increased to 1275 images using data augmentation techniques like random rotation, horizontal flip, and color jittering.

- Dataset 2
  - Source: https://github.com/ieee8023/covid-chestxray-dataset
  - Type of images: Chest X-Rays
  - Dataset size: 514 images
  - No. of COVID positive images: 320
  - No. of COVID negative images: 194
  - Train set size: 200 images
  - Validation set size: 114 images
  - Test set size: 200 images

- Dataset 3
  - Source: https://github.com/mr7495/COVID-CTset
  - Type of images: CT scans
  - Dataset size: 12058 images
  - No. of COVID positive images: 2282
  - No. of COVID negative images: 9776
  - Train set size: 11400 images
  - Validation set size: 258 images
  - Test set size: 200 images

- Dataset 4
  - Source: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
  - Type of images: Chest X-Rays
  - Dataset size: 6940 images
  - No. of COVID positive images: 219
  - No. of COVID negative images: 6721
  - Train set size: 4009 images
  - Validation set size: 1426 images
  - Test set size: 1505 images
- Dataset 5
  - Source: https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset
  - Type of images: CT scans
  - Dataset size: 2481 images
  - No. of COVID positive images: 1252
  - No. of COVID negative images: 1229
  - Train set size: 1800 images
  - Validation set size: 281 images
  - Test set size: 400 images

7.2 Comparative Analysis

The tables below show the comparison in the performance of the proposed model against other models in terms of precision, recall, accuracy, and F1 score on different datasets.

Table 1: Comparision between the proposed model and other baseline models on Dataset 1

Model	Precision	Recall	Accuracy	F1 Score
$VGG19_0$	0.7479	0.8476	0.7734	0.7946
$VGG19_1$	0.7583	0.8667	0.7882	0.8089
$VGG19_2$	0.75	0.8571	0.7783	0.8
$ResNet101_0$	0.6612	0.7619	0.6748	0.7080
$ResNet101_1$	0.7788	0.7714	0.7685	0.7751
$ResNet101_2$	0.6351	0.8952	0.6798	0.7431
$DenseNet169_0$	0.7385	0.9143	0.7882	0.8170
$DenseNet169_1$	0.8	0.7238	0.7635	0.76
$DenseNet169_2$	0.775	0.8857	0.8079	0.8267
$WideResNet50\ 2_0$	0.7523	0.7810	0.7537	0.7664
$WideResNet50\ 2_1$	0.7241	0.8	0.7389	0.7602
$WideResNet50\ 2_2$	0.7766	0.6952	0.7389	0.7337
$Combination1$	0.7769	0.8952	0.8128	0.8319
$Combination2$	0.7815	0.8857	0.8128	0.8303
$Proposed\ Model$	0.7951	0.9238	0.8374	0.8533

Table 2: Comparision between the proposed model and other baseline models on Dataset 2

Model	Precision	Recall	Accuracy	F1 Score
$VGG19_0$	0.7953	0.9714	0.805	0.8746
$VGG19_1$	0.8299	0.8714	0.785	0.8502
$VGG19_2$	0.7688	0.95	0.765	0.8498
$ResNet101_0$	0.7771	0.9214	0.76	0.8431
$ResNet101_1$	0.8299	0.8714	0.785	0.8502
$ResNet101_2$	0.8462	0.8643	0.795	0.8551
$DenseNet169_0$	0.8468	0.75	0.73	0.7955
$DenseNet169_1$	0.7949	0.8857	0.76	0.8378
$DenseNet169_2$	0.8301	0.8071	0.75	0.8188
$WideResNet50\ 2_0$	0.7881	0.85	0.735	0.8179
$WideResNet50\ 2_1$	0.7902	0.8071	0.715	0.7986
$WideResNet50\ 2_2$	0.8356	0.8714	0.79	0.8531
$Combination1$	0.8207	0.85	0.765	0.8351
$Combination2$	0.7950	0.9142	0.775	0.8505
$Proposed\ Model$	0.9618	0.9	0.905	0.9299

Table 3: Comparision between the proposed model and other baseline models on Dataset 3

Model	Precision	Recall	Accuracy	F1 Score
$VGG19_0$	0.9949	0.98	0.9875	0.9874
$VGG19_1$	0.9896	0.955	0.9725	0.9720
$VGG19_2$	0.9897	0.96	0.975	0.9746
$ResNet101_0$	0.9948	0.95	0.9725	0.9719
$ResNet101_1$	1	0.935	0.9675	0.9664
$ResNet101_2$	0.975	0.975	0.975	0.975
$DenseNet169_0$	0.9946	0.915	0.955	0.9531
$DenseNet169_1$	0.9896	0.95	0.97	0.9694
$DenseNet169_2$	0.9897	0.965	0.9775	0.9772
$WideResNet50\ 2_0$	0.9843	0.94	0.9625	0.9616
$WideResNet50\ 2_1$	0.9947	0.93	0.9625	0.9612
$WideResNet50\ 2_2$	0.9948	0.955	0.975	0.9745
$Combination1$	0.9898	0.975	0.9825	0.9824
$Combination2$	1	0.975	0.9875	0.9873
$Proposed\ Model$	1	0.98	0.99	0.9899

Table 4: Comparision between the proposed model and other baseline models on Dataset 4

Model	Precision	Recall	Accuracy	F1 Score
$VGG19_0$	1	0.8852	0.9954	0.9391
$VGG19_1$	0.9818	0.9	0.9953	0.9391
$VGG19_2$	0.9464	0.8833	0.9934	0.9138
$ResNet101_0$	0.9615	0.8333	0.9920	0.8929
$ResNet101_1$	0.9464	0.8833	0.9934	0.9138
$ResNet101_2$	0.9773	0.7167	0.9880	0.8269
$DenseNet169_0$	0.9815	0.8833	0.9947	0.9298
$DenseNet169_1$	0.9808	0.85	0.9934	0.9107
$DenseNet169_2$	1	0.8852	0.9954	0.9391
$WideResNet50\ 2_0$	0.9818	0.9	0.9953	0.9391
$WideResNet50\ 2_1$	0.9815	0.8833	0.9947	0.9298
$WideResNet50\ 2_2$	0.9818	0.9	0.9953	0.9391
$Combination1$	0.9815	0.8833	0.9947	0.9298
$Combination2$	0.9818	0.9	0.9953	0.9391
$Proposed\ Model$	1	0.9333	0.9973	0.9655

Table 5: Comparision between the proposed model and other baseline models on Dataset 5

Model	Precision	Recall	Accuracy	F1 Score
$VGG19_0$	0.8389	0.885	0.8575	0.8613
$VGG19_1$	0.8691	0.83	0.8525	0.8491
$VGG19_2$	0.8960	0.775	0.8425	0.8311
$ResNet101_0$	0.9341	0.78	0.8625	0.8501
$ResNet101_1$	0.8848	0.73	0.8175	0.8
$ResNet101_2$	0.8942	0.845	0.8725	0.8689
$DenseNet169_0$	0.9270	0.825	0.88	0.8730
$DenseNet169_1$	0.9045	0.71	0.8175	0.7955
$DenseNet169_2$	0.9096	0.855	0.885	0.8814
$WideResNet50\ 2_0$	0.9223	0.89	0.9075	0.9059
$WideResNet50\ 2_1$	0.9179	0.895	0.9075	0.9063
$WideResNet50\ 2_2$	0.8653	0.835	0.8525	0.8499
$Combination1$	0.9302	0.8	0.87	0.8602
$Combination2$	0.9171	0.83	0.8775	0.8714
$Proposed\ Model$	0.9137	0.9	0.9075	0.9068

Note:

- $Model_0$ represents the Model with a softmax layer.
- $Model_1$ represents the Model with a fully connected layer and a softmax layer.
- $Model_2$ represents the Model with two fully connected layers and a softmax layer.
- The different types of Models used are $VGG19$, $ResNet101$, $DenseNet169$, $WideResNet50$ 2.
- The softmax layer is used to map the output to the required number of classes.
- Combination1 represents $DenseNet169_0$, $DenseNet169_1$, $DenseNet169_2$ used as an ensemble.
- Combination2 represents $DenseNet169_0$, $DenseNet169_2$, $WideResNet50$ 2_2 used as an ensemble.

The tables below show the performance of the proposed model on different datasets as the threshold above which an image is predicted positive is varied.

Table 6: Performance of the proposed model on Dataset 1 as the threshold is varied

Threshold	Precision	Recall	Accuracy	F1 Score
0.1	0.6459	0.9905	0.7143	0.7820
0.2	0.7007	0.9810	0.7734	0.8175
0.3	0.7279	0.9429	0.7882	0.8216
0.4	0.7597	0.9333	0.8128	0.8376
0.5	0.7951	0.9238	0.8374	0.8546
0.6	0.8	0.9142	0.8374	0.8533
0.7	0.7965	0.8571	0.8128	0.8257
0.8	0.8218	0.7905	0.8030	0.8058
0.9	0.9054	0.6381	0.7783	0.7486

Table 7: Performance of the proposed model on Dataset 2 as the threshold is varied

Threshold	Precision	Recall	Accuracy	F1 Score
0.1	0.8961	0.9857	0.91	0.9388
0.2	0.9172	0.95	0.905	0.9333
0.3	0.9286	0.9286	0.9	0.9286
0.4	0.9416	0.9214	0.905	0.9314
0.5	0.9618	0.9	0.905	0.9299
0.6	0.9609	0.8786	0.89	0.9179
0.7	0.9752	0.8429	0.875	0.9042
0.8	0.9746	0.8214	0.86	0.8915
0.9	0.9794	0.6786	0.765	0.8017

Table 8: Performance of the proposed model on Dataset 3 as the threshold is varied

Threshold	Precision	Recall	Accuracy	F1 Score
0.1	0.985	0.985	0.985	0.985
0.2	0.9949	0.985	0.99	0.99
0.3	1	0.985	0.9925	0.9924
0.4	1	0.98	0.99	0.9899
0.5	1	0.98	0.99	0.9899
0.6	1	0.97	0.985	0.9848
0.7	1	0.965	0.9825	0.9822
0.8	1	0.965	0.9825	0.9822
0.9	1	0.945	0.9725	0.9717

Table 9: Performance of the proposed model on Dataset 4 as the threshold is varied

Threshold	Precision	Recall	Accuracy	F1 Score
0.1	1	0.9333	0.9973	0.9655
0.2	1	0.9333	0.9973	0.9655
0.3	1	0.9333	0.9973	0.9655
0.4	1	0.9333	0.9973	0.9655
0.5	1	0.9333	0.9973	0.9655
0.6	1	0.9333	0.9973	0.9655
0.7	1	0.9333	0.9973	0.9655
0.8	1	0.9333	0.9973	0.9655
0.9	1	0.9333	0.9973	0.9655

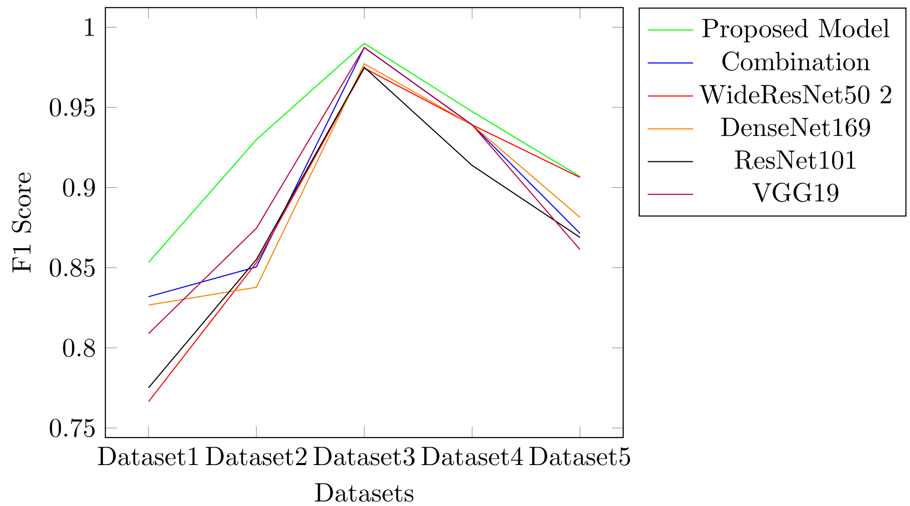
Table 10: Performance of the proposed model on Dataset 5 as the threshold is varied

Threshold	Precision	Recall	Accuracy	F1 Score
0.1	0.8527	0.955	0.895	0.9009
0.2	0.8942	0.93	0.91	0.9118
0.3	0.915	0.915	0.915	0.915
0.4	0.9146	0.91	0.9125	0.9123
0.5	0.9137	0.9	0.9075	0.9068
0.6	0.9293	0.855	0.895	0.8906
0.7	0.9486	0.83	0.8925	0.8853
0.8	0.9586	0.81	0.8875	0.8780
0.9	0.9682	0.76	0.8675	0.8515

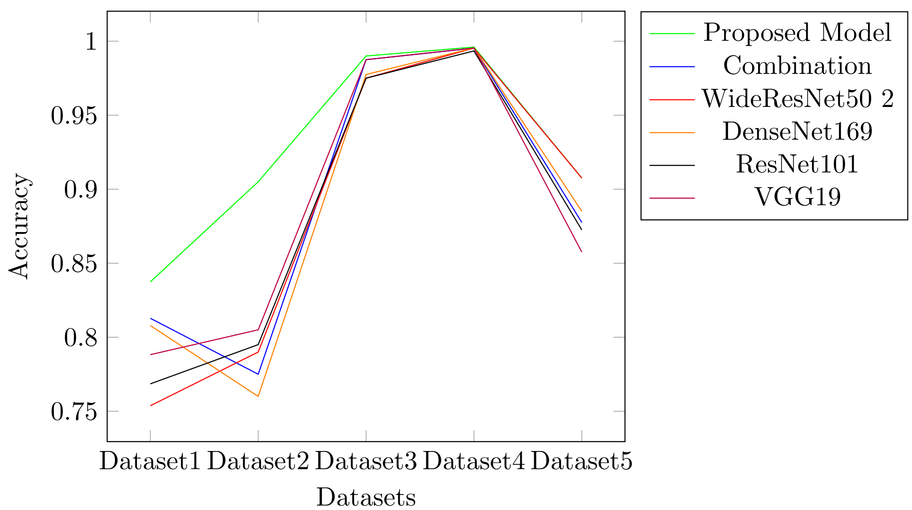
Table 11: Comparison between the proposed model and models proposed in previous research papers.

Model	Dataset 1 (F1 Score)	Dataset 2 (Accuracy)	Dataset 3 (Accuracy)	Dataset 4 (Accuracy)
Model[1]	0.8114	N/A	N/A	N/A
Model[2]	0.85	N/A	N/A	N/A
Model[3]	0.8333	N/A	N/A	N/A
Model[4]	0.7150	N/A	N/A	N/A
Model[5]	N/A	0.719	N/A	N/A
Model[6]	0.794	N/A	N/A	N/A
Model[7]	N/A	N/A	0.9849	N/A
Model[8]	N/A	N/A	N/A	0.9720
Model[9]	N/A	N/A	N/A	0.9969
Model[10]	N/A	N/A	N/A	0.9935
Proposed Model	0.8533	0.905	0.99	0.9973

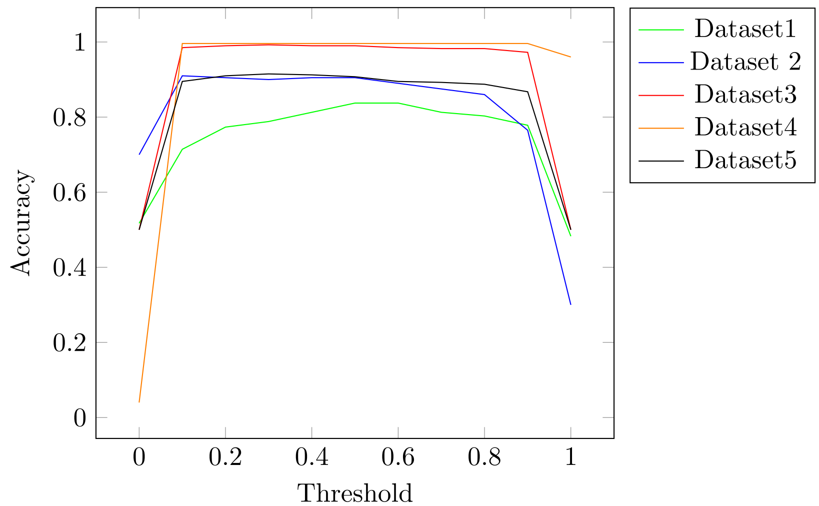
Grpah 1: F1 Score of Different models on different datasets



Graph 2: Accuracy of Different models on different datasets



Graph 3: Variation of Accuracy with threshold



8 Result Analysis

From tables 1 to 5 and graphs 1 and 2, it can be clearly seen that the proposed model performs better than the other models in terms of accuracy and F1 score on all the datasets. Since the proposed model uses the concept of stacking to combine three different models, any misclassification made by one model can be compensated by the other two models; hence increasing the accuracy and F1 score of the proposed model. Therefore, the proposed model is able to perform better than the individual models.

The proposed model consists of three different models, each of these three models consist of a pre-trained model and additional fully connected layers, these additional fully connected layers help the model learn features specific to a particular dataset. This is another reason for the improved performance of the proposed model.

From table 11, it can also be seen that the proposed model does better than the models proposed in previous research papers.

The threshold above which a positive prediction should be made varies with different datasets. The threshold also depends on what one wants to give more preference, if better precision is preferred i.e fewer false positives, the threshold should be higher, whereas if a better recall is preferred i.e fewer false negatives, the threshold should be lower.

In the detection of COVID 19, it is important to not make false negative predictions, as it can have major consequences like the increased chance of an infected person spreading the disease to other people.

Therefore recall should be given more preference. Therefore from tables 5 to 10 and graph 3, it can be seen that the best threshold is between 0.3 and 0.5.

References

[1] Mohamed Loey, Gunasekaran Manogaran, and Nour Eldeen M. Khalifa. A Deep Transfer Learning Model with Classical Data Augmentation and CGAN to Detect COVID-19 from Chest CT Radiography Digital Images. May 2020.

[2] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans. *medRxiv*, page 2020.04.13.20063941, April 2020.

[3] Matteo Polsinelli, Luigi Cinque, and Giuseppe Placidi. A Light CNN for detecting COVID-19 from CT scans of the chest. *arXiv:2004.12837 [cs, eess]*, April 2020. arXiv: 2004.12837.

[4] Ahmed I.Saleh Warda M.Shaban, Asmaa H.Rabie and M.A.Abo-Elsoud. A new COVID-19 Patients Detection Strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier. *Knowledge-Based Systems*, 205:106270, October 2020.

[5] Mohd Zulfaezal Che Azemin, Radhiana Hassan, Mohd Izzuddin Mohd Tamrin, and Mohd Adli Md Ali. COVID-19 Deep Learning Prediction Model Using Publicly Available Radiologist-Adjudicated Chest X-Ray Images as Training Data: Preliminary Findings, August 2020.

[6] Rohit Lokwani, Ashrika Gaikwad, Viraj Kulkarni, Aniruddha Pant, and Amit Kharat. Automated Detection of COVID-19 from CT Scans Using Convolutional Neural Networks. *arXiv:2006.13212 [cs, eess]*, June 2020. arXiv: 2006.13212.

[7] Mohammad Rahimzadeh, Abolfazl Attar, and Seyed Mohammad Sakhaei. A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset. *medRxiv*, page 2020.06.08.20121541, September 2020.

[8] CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images. *Chaos, Solitons & Fractals*, 140:110245, November 2020.

[9] Mundher Taresh, Ningbo Zhu, and Talal Ahmed Ali Ali. Transfer learning to detect COVID-19 automatically from X-ray images, using convolutional neural networks. *medRxiv*, page 2020.08.25.20182170, August 2020.

[10] Samir S. Yadav, Mininath R. Bendre, Pratap S. Vikhe, and Shivajirao M. Jadhav. Analysis of deep machine learning algorithms in COVID-19 disease diagnosis. *arXiv:2008.11639 [cs, eess]*, August 2020. arXiv: 2008.11639.

[11] Chest X-ray scanning based detection of COVID-19 using deepconvolutional neural network. August 2020.

[12] Coronavirus disease 2019, September 2020. Page Version ID: 978125859.

[13] Mohana Basu. RT-PCR, antigen, antibody, TrueNAT — all you need to know about the different Covid tests, June 2020.

[14] Jason Brownlee. What is Deep Learning?, August 2019.

[15] Jason Brownlee. A Gentle Introduction to Transfer Learning for Deep Learning, December 2017.

[16] UCSD-AI4H. UCSD-AI4H/COVID-CT, September 2020. original-date: 2020-03-27T23:29:54Z.

[17] Joseph Paul Cohen. ieee8023/covid-chestxray-dataset, September 2020. original-date: 2020-02-14T23:22:23Z.

[18] Mohammad Rahimzadeh. mr7495/COVID-CTset, September 2020. original-date: 2020-06-01T15:41:22Z.

[19] COVID-19 Radiography Database.

[20] SARS-COV-2 Ct-Scan Dataset.

[21] Xingyi Yang, Xuehai He, Jinyu Zhao, Yichen Zhang, Shanghang Zhang, and Pengtao Xie. COVID-CT-Dataset: A CT Scan Dataset about COVID-19. *arXiv:2003.13865 [cs, eess, stat]*, June 2020. arXiv: 2003.13865.

[22] Linda Wang and Alexander Wong. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. *arXiv:2003.09871 [cs, eess]*, May 2020. arXiv: 2003.09871.

[23] Muhammad Farooq and Abdul Hafeez. COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs. *arXiv:2003.14395 [cs, eess]*, March 2020. arXiv: 2003.14395.

[24] Xiaowei Xu, Xiangao Jiang, Chunlian Ma, Peng Du, Xukun Li, Shuangzhi Lv, Liang Yu, Yanfei Chen, Junwei Su, Guanjing Lang, Yongtao Li, Hong Zhao, Kaijin Xu, Lingxiang Ruan, and Wei Wu. Deep Learning System to Screen Coronavirus Disease 2019 Pneumonia. *arXiv:2002.09334 [physics]*, February 2020. arXiv: 2002.09334.

[25] How to write the literature review of your research paper, November 2018.

[26] torchvision.transforms — PyTorch 1.6.0 documentation.

[27] Jason Brownlee. Stacking Ensemble Machine Learning With Python, April 2020.

[28] Coronavirus Update (Live): 28,988,031 Cases and 925,320 Deaths from COVID-19 Virus Pandemic - Worldometer.

[29] Understanding the VGG19 Architecture, February 2020.

[30] ResNet (34, 50, 101): Residual CNNs for Image Classification Tasks, January 2019.

[31] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. *arXiv:1608.06993 [cs]*, January 2018. arXiv: 1608.06993.

[32] What are Wide Resnets? - Quora.

[33] Wide ResNet-50-2 - Wolfram Neural Net Repository.