PROJECT REPORT

DIAGNOSING PNEUMONIA AND COVID-19 FROM CHEST X-RAYS USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

Group 6 | EEE - 312 | 25/7/2022

Submitted By:

Name: Tiasa Mondal

Student ID: 1706054

Name: Adib Md. Ridwan

Student ID: 1706055

Name: S. M. Monzurul Hoque Chowdhury

Student ID: 1706056

Name: Prithu Mahmud

Student ID: 1706057

Introduction:

Pneumonia is a life-threatening infectious disease affecting one or both lungs in humans commonly caused by bacteria called Streptococcus pneumoniae. On the other hand, COVID-19 also known as Severe Acute Respiratory Syndrome Corona virus-2, is a contagious disease that is released from tiny droplets containing saliva or mucus from respiratory system of a diseased person when he talks, sneezes, or coughs. It is a crisis on global health since it spans on a global scale as a highly contagious disease. The symptoms and indications of COVID-19 & pneumonia are almost identical. If the diagnosis is not properly done, then it can be the cause of death. In our project, we have used Convolutional Neural Networks (CNNs) to detect Covid-19 & Pneumonia from X-Ray images of lungs.

Related Works:

1. Deep learning (DL): Deep learning is a sector of AI and a type of machine learning that imitates the way human brain works by using deep layers of neurons to process input information in order to come to a decision. It includes statistics and predictive modeling. To automate predictive analysis, DL bears much importance in data science.

2. Convolutional Neural Network (CNN): In DL, CNN is a class of deep neural networks which are commonly used to analyze visual imagery and classify them accordingly. Its layers utilize the Convolution operation instead of the traditional interconnecting Dense layers to process image data better.

3. Transfer learning (TL): TL is a machine learning method that focuses on using parameters obtained after training on a large dataset and using those parameters as a starting point for training on a smaller dataset provided that both datasets have a similar distribution. The goal is to use knowledge learned from tasks where a lot of labeled data is available to where little labeled data is available. This approach generally improves training results and allows the model to converge after a smaller number of epochs.

Methodology

Our project goal is to diagnose Pneumonia & COVID-19 from chest X-rays. We have used convolutional neural networking methods in order to implement this. We used Google Colaboratory for this purpose. The whole process can be divided into some steps for proper elaboration. These steps are as follows:

Step 1: As we proceeded with the whole task in Google Colaboratory, initially x-ray image data related to Pneumonia diagnosis & COVID-19 diagnosis was transferred to the platform via Google Drive.

Step 2: Data preprocessing and augmentation was done using Keras ImageDataGenerator. We know, data augmentation is a strategy that enables one to significantly increase the diversity of data available for training models, without actually collecting new data. All images were converted to 224x224 pixels to fit the models.

Step 3: After proper data augmentation, we split the whole data into training data & validation data. The ratio was 80% to 20% where 80% of total data was training data & 20% was validation data.

Step 4: This is the most important step in this whole process where we implemented transfer learning. Here, we used five different base models. These are VGG-16, Resnet50V2, VGG-19, DenseNet121 & InceptionV3. We trained on these different models to find out the best model to diagnose the diseases perfectly. Before training, we added a GlobalAvgPooling2D and a 2-neuron Dense layer with Softmax activation. Initial weight, i.e., defining initial values for the parameters in neural network models, were obtained from training done on ImageNet dataset. The parameters of the body of the base model are frozen in order to retain the high level characterization ability obtained from training on ImageNet Dataset.

Step 5: We used Adam optimizer for stochastic gradient descent to train the networks. This algorithm combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. Moreover, we used the Categorical Cross Entropy loss function as the loss(cost) function. Batch size, i.e., the number of samples processed before updating the model was set to be 32.

Step 6: Training was completed for each model on both Pneumonia and COVID-19 datasets for a suitable number of epochs. Then we plotted accuracy & loss graphs for both training data & validation data for all five different models.

Step 7: We saved the training history as a variable with the pickle module of python. Moreover, the entire model was saved with save_model() function of Keras.

Step 8: Each model for COVID-19 classification ran for 100 epochs. Learning rate was set at 0.0001. In this case, the loss & accuracy curves showed satisfactory results.

Step 9: For Pneumonia classification, initially the learning rate was set at 0.001 & we ran the models for 50 epochs. Though this initialization provided us with good results, this resulted in unsteady accuracy & loss plots during training. They jumped too often. So, we decreased the learning rate to 0.0001. This allowed the models to learn at a slower rate and not overshoot too often. So, all five models were again trained with this learning rate of 0.0001 & for 100 epochs. This higher epoch was necessary for convergence as the learning was being done at a slower rate in this scenario. As a result, we obtained both good results and steady training plots.

Step 10: The evaluation metrices being used here were accuracy, loss, precision, recall & f1score. We found each value for each model and compared them. We found that VGG-16 provided us with the best results followed by VGG-19. This actually matches with the IEEE paper we followed which stated that VGG-16 showed the best results for this particular dataset.

Step 11: A test folder was included. New X-ray images can be uploaded to this folder for diagnosing. Two separate models classify the image as Pneumonia or not Pneumonia and COVID or not COVID respectively.

Overall workflow is summarized in this flow chart:

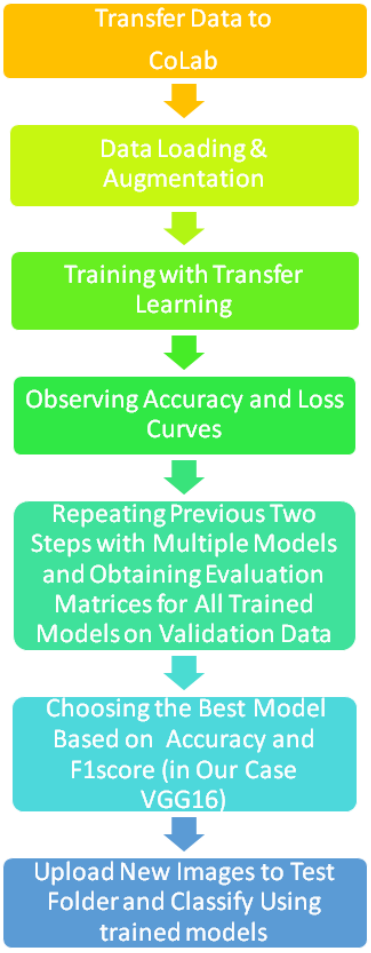


Table I Vgg-16 CNN Model Architecture

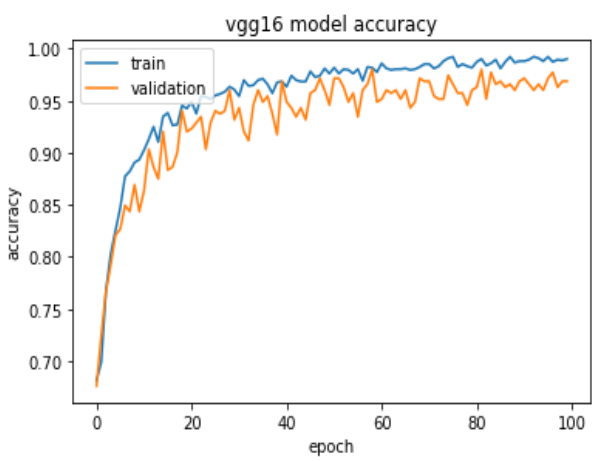
Layer (type)	*Output Shape*	*Param #*
Input	(224, 224, 3)	0
Convolution	(224, 224, 32)	896
Batch Normalization	(224, 224, 32)	128
Activation	(224, 224, 32)	0
Max Pooling	(112, 112, 32)	0
Convolution	(112, 112, 64)	18496
Batch Normalization	(112, 112, 64)	256
Activation	(112, 112, 64)	0
Max Pooling	(112, 112, 32)	0
Convolution	(112, 112,64)	36928
Batch Normalization	(112, 112, 64)	256
Activation	(112, 112, 64)	0
Max Pooling	(56, 56, 64)	0
Convolution	(56, 56, 128)	73856
Batch Normalization	(56, 56, 128)	512
Activation	(56, 56, 128)	0
Max Pooling	(56, 56, 64)	0
Convolution	(56, 56, 128)	147584
Batch Normalization	(56, 56, 128)	512
Activation	(56, 56, 128)	0
Flatten	100352	0
Dense	512	51380736
Batch Normalization	512	2048
Activation	512	0
Dropout	512	0
Dense	2	1026
Activation	2	0

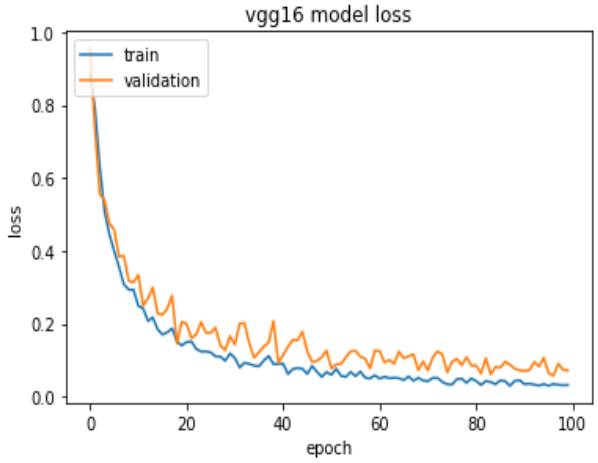
Results

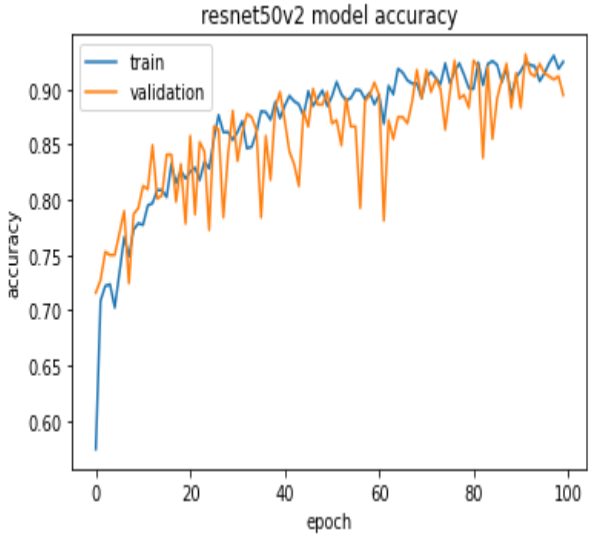
COVID-19 CLASSIFICATION

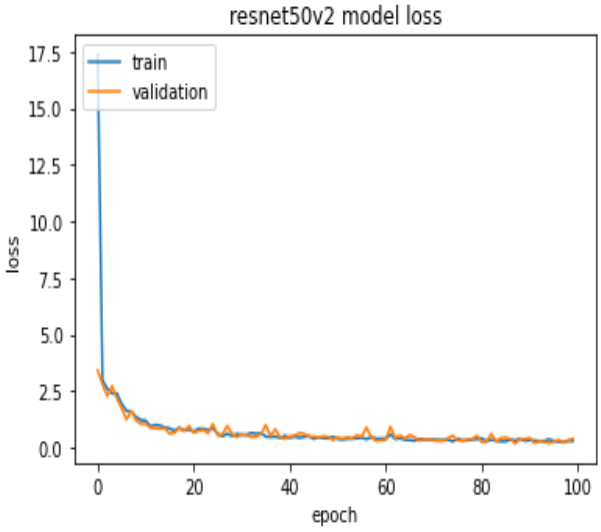
Training History:

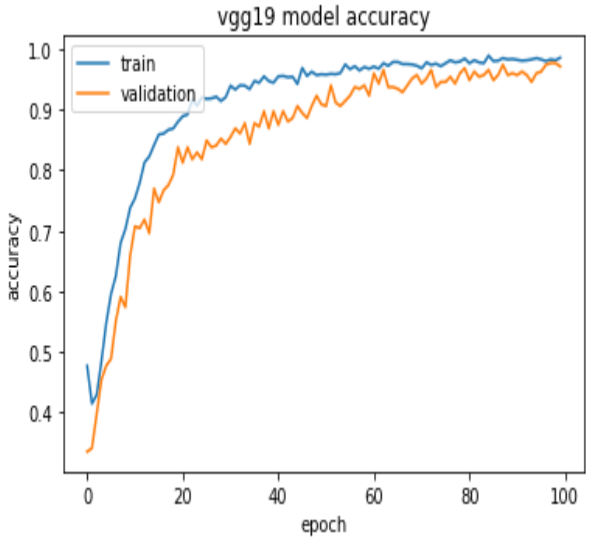
batch size=32, learning rate=0.0001, epoch=100

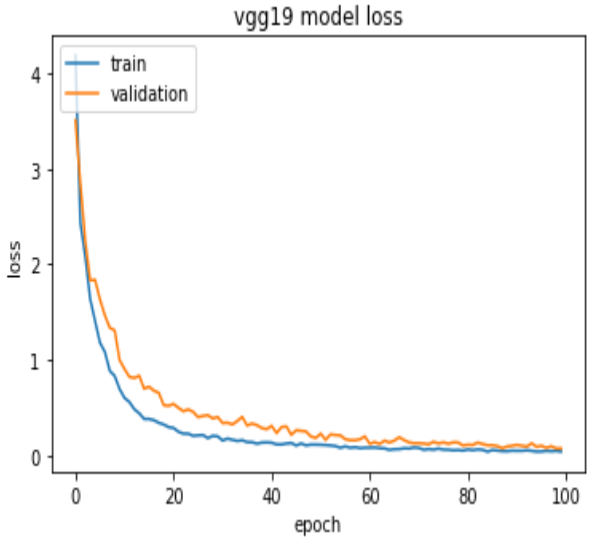




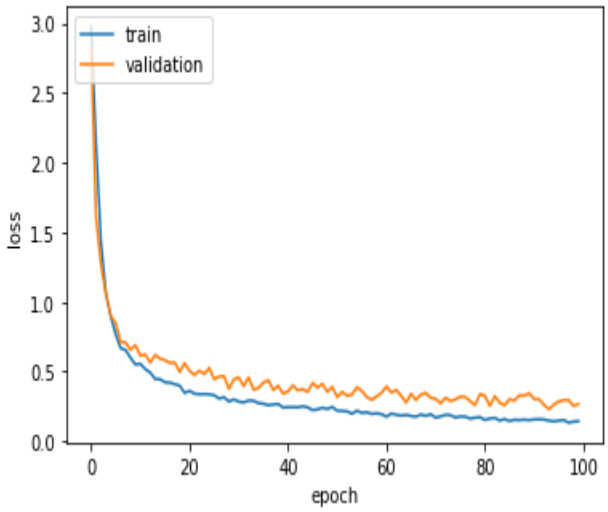


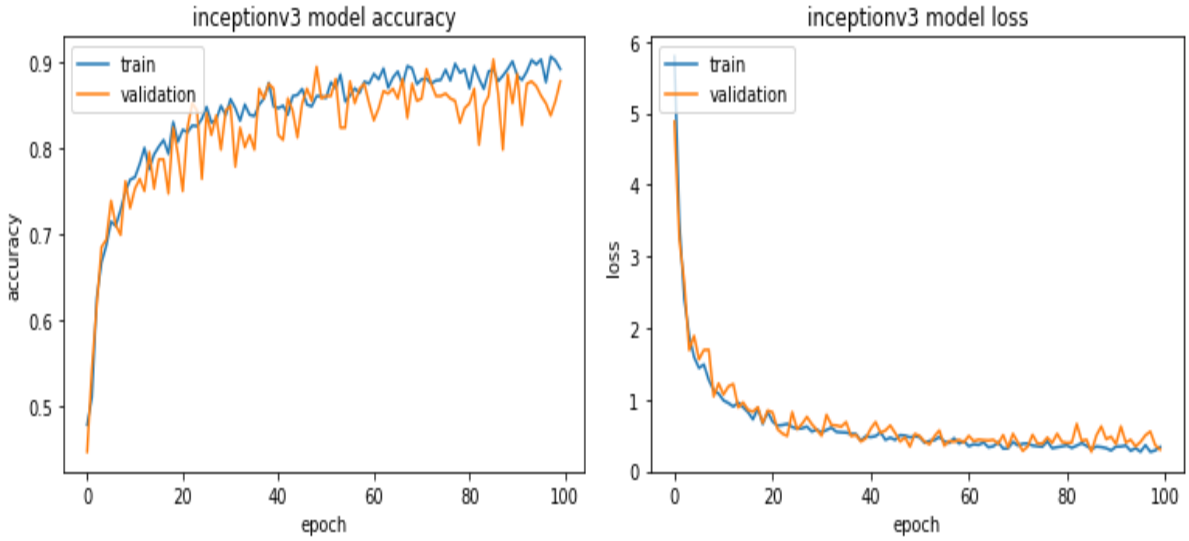



vgg19 model accuracy
train
validation
accuracy
epoch
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
20
40
60
80
100


vgg19 model loss
train
validation
loss
epoch
4
3
2
1
0
0
20
40
60
80
100


densenet121 model accuracy
train
validation
accuracy
epoch
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
20
40
60
80
100


densenet121 model loss
train
validation
loss
epoch
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100



Evaluation Metrics on Validation Data:

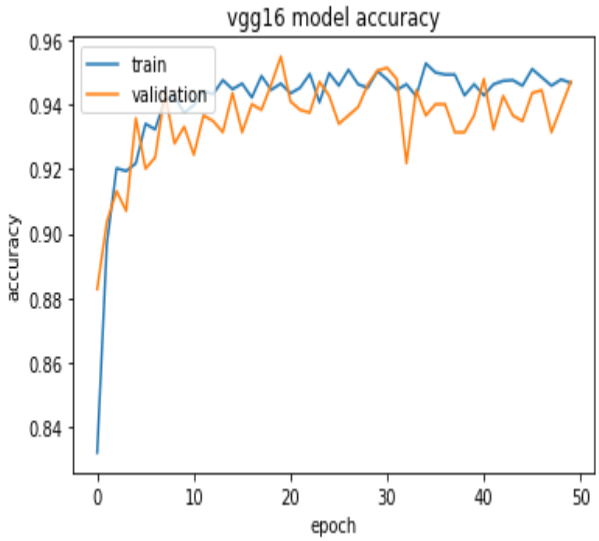
batch size=32, learning rate=0.0001, epoch=100

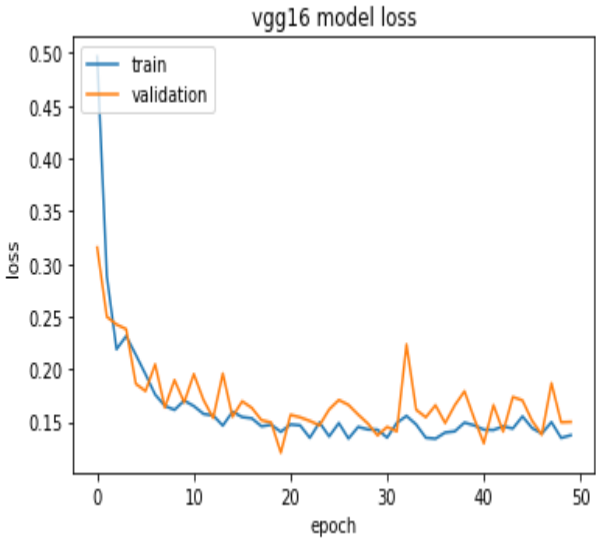
	loss	accuracy	precision	recall	f1_score
VGG16	0.084387	0.978022	1.000000	0.967033	0.983240
Resnet50V2	0.412650	0.901099	0.967949	0.829670	0.893491
VGG19	0.099641	0.950549	0.977011	0.934066	0.955056
DenseNet121	0.268713	0.901099	0.948718	0.813187	0.875740
InceptionV3	0.275648	0.895604	0.889535	0.840659	0.864407

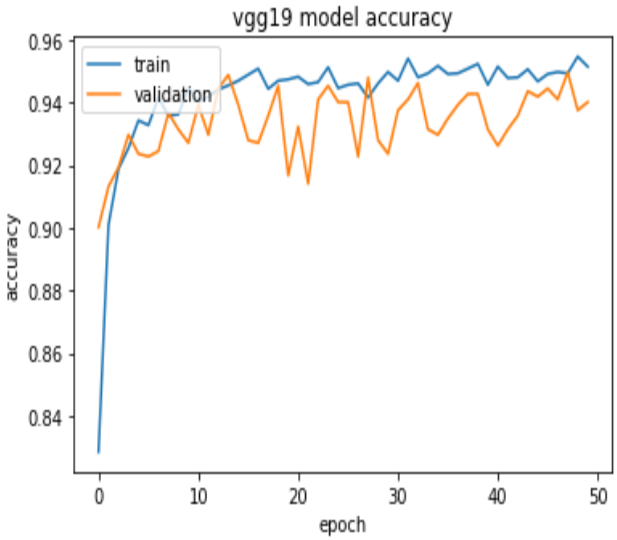
PNEUMONIA CLASSIFICATION

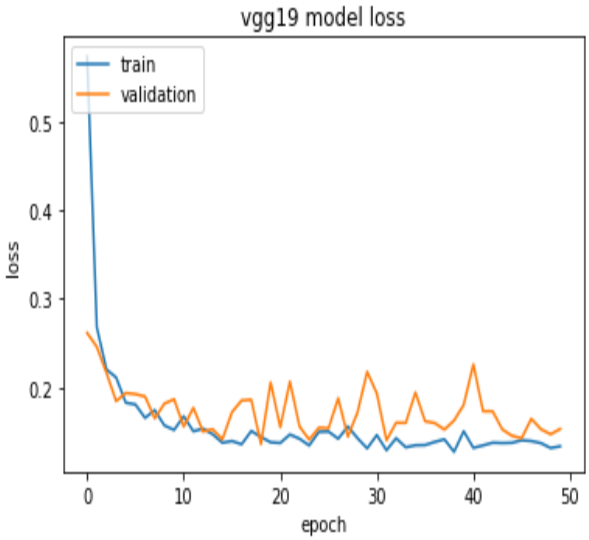
Training History

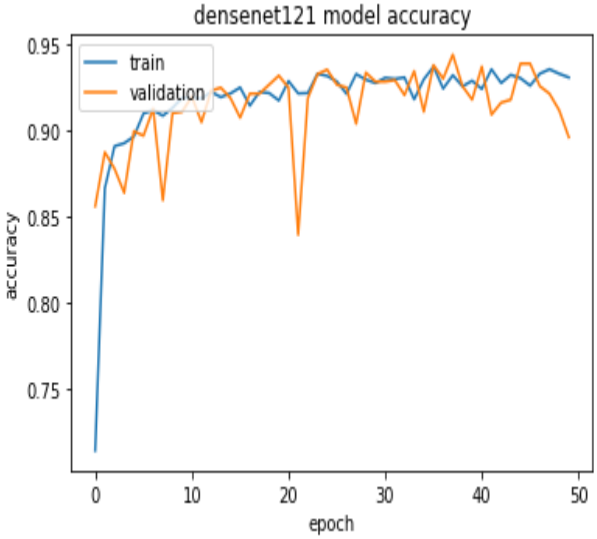
1 . batch size=32, learning rate=0.001, epoch=50

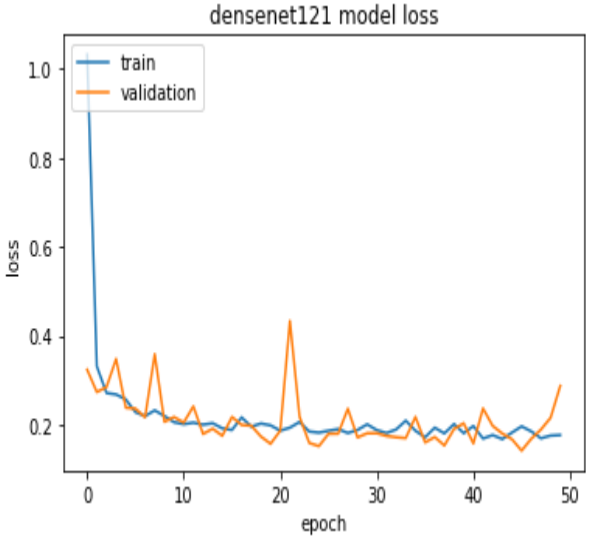


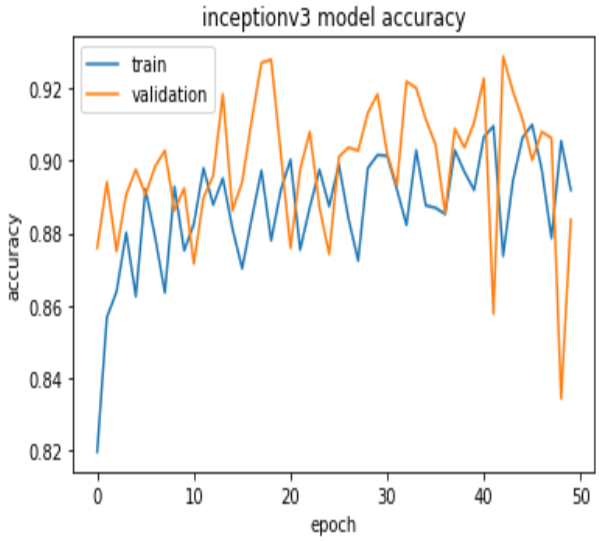


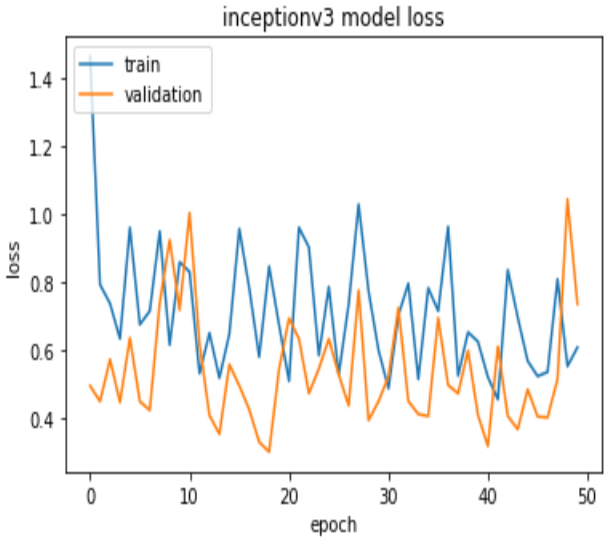

vgg19 model accuracy
train
validation
accuracy
epoch


vgg19 model loss
train
validation
loss
epoch

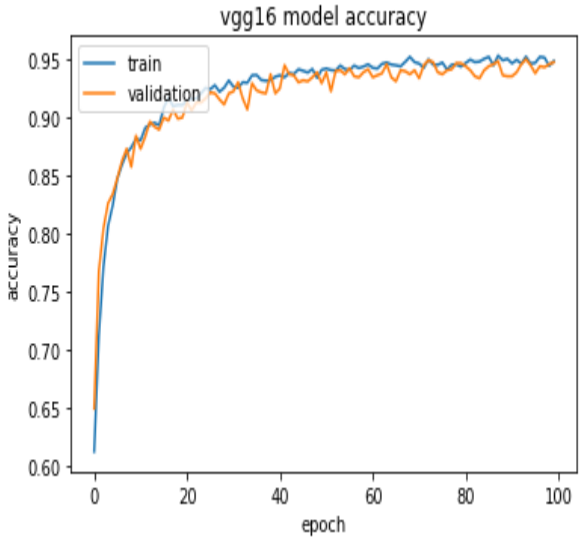

densenet121 model accuracy
train
validation
accuracy
epoch

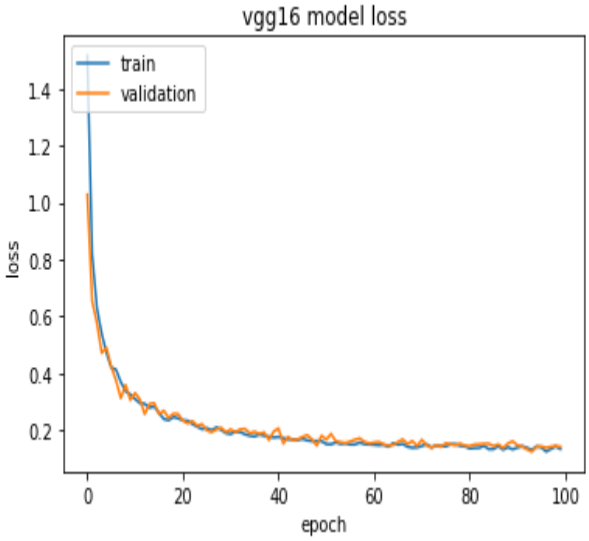

densenet121 model loss
train
validation
loss
epoch



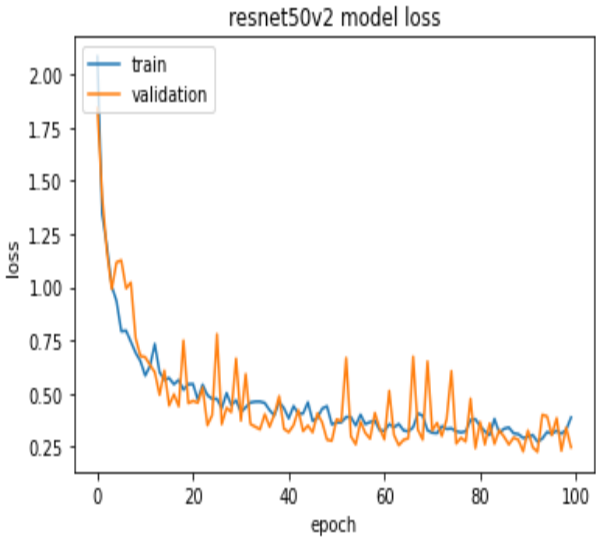


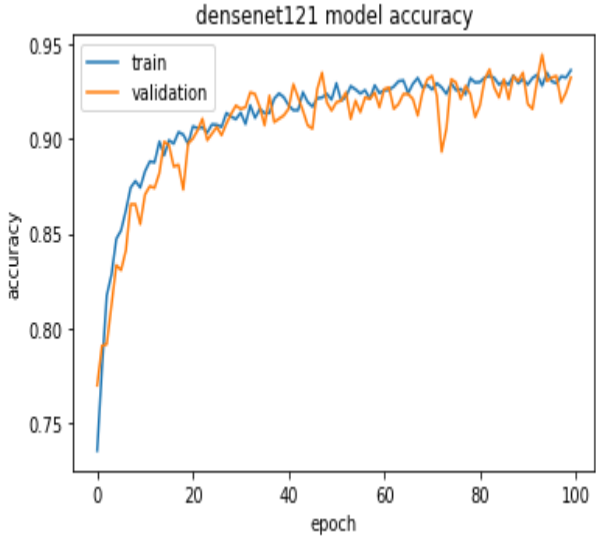
2. batch size=32, learning rate=0.0001, epoch=100

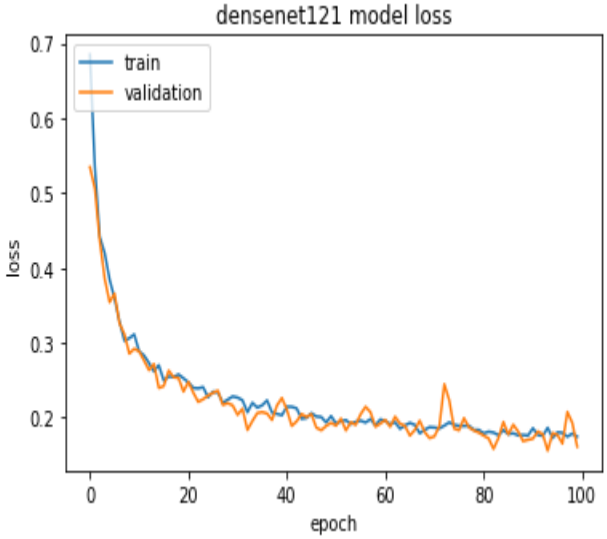


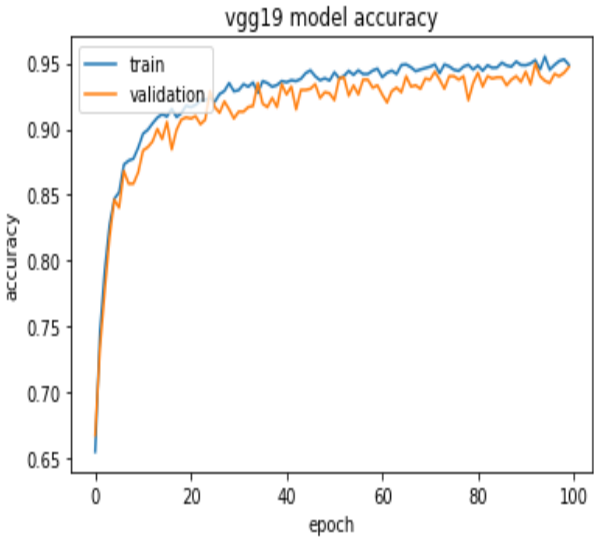


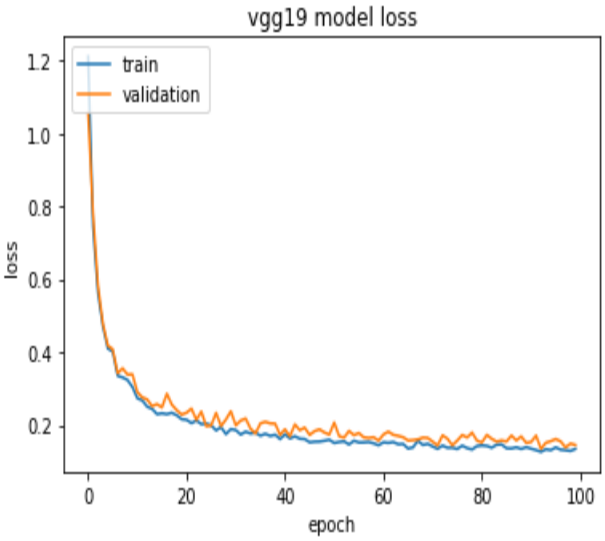

resnet50v2 model accuracy
train
validation
accuracy
epoch
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0
20
40
60
80
100


resnet50v2 model loss
train
validation
loss
epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0
20
40
60
80
100

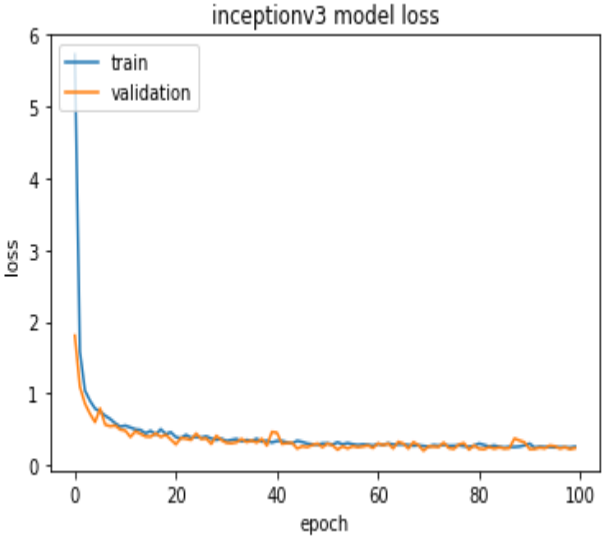

densenet121 model accuracy
train
validation
accuracy
epoch
0.95
0.90
0.85
0.80
0.75
0
20
40
60
80
100


densenet121 model loss
train
validation
loss
epoch
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100


vgg19 model accuracy
train
validation
accuracy
epoch
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0
20
40
60
80
100


vgg19 model loss
train
validation
loss
epoch
1.2
1.0
0.8
0.6
0.4
0.2
0
20
40
60
80
100


inceptionv3 model accuracy
train
validation
accuracy
epoch
0.90
0.85
0.80
0.75
0.70
0.65
0
20
40
60
80
100


inceptionv3 model loss
train
validation
loss
epoch
6
5
4
3
2
1
0
0
20
40
60
80
100

Evaluation Metrics on Validation Data

1.batch size=32, learning rate=0.001, epoch=50

	loss	accuracy	precision	recall	f1_score
VGG16	0.147635	0.945299	0.977327	0.959016	0.968085
Resnet50V2	1.078044	0.871795	0.995745	0.822014	0.900577
VGG19	0.162498	0.938462	0.955451	0.954333	0.954892
DenseNet121	0.290966	0.893162	0.989218	0.859485	0.919799
InceptionV3	0.776927	0.885470	0.884043	0.973068	0.926421

2.batch size=32, learning rate=0.0001, epoch=100

	loss	accuracy	precision	recall	f1_score
VGG16	0.144859	0.944444	0.959491	0.970726	0.965076
Resnet50V2	0.207803	0.935043	0.958432	0.944965	0.951651
DenseNet121	0.171671	0.923077	0.938426	0.963700	0.950895
VGG19	0.153860	0.933333	0.959064	0.960187	0.959626
InceptionV3	0.211841	0.918803	0.974233	0.929742	0.951468

Analysis

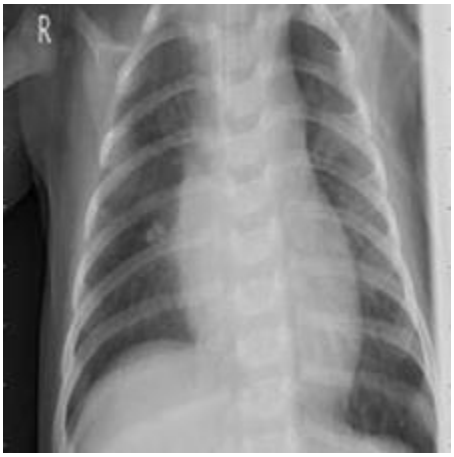
For COVID-19 Classification, the VGG-16 model gave the best results. All the models converged properly for a learning rate of 0.0001, as can be seen from the accuracy and loss plots obtained during training.

For Pneumonia Classification as well, VGG-16 model gave the best results. At first training was done for 50 epochs using a learning rate of 0.001 in order to speed up training on the large dataset. All 5 models showed erratic behavior during training and neither the accuracy nor the loss converged properly. So, the learning rate was decreased to 0.0001 and the training was again done for 100 epochs. This time both accuracy and loss were seen to converge. This kind of high learning rate problem represents a classic reason for convergence failure of Deep Learning models. Using a high learning rate speeds up training but causes the loss function to jump around near the convergence point but never actually converge. Using a smaller learning rate causes the Gradient Descent algorithm to take smaller steps and thus come closer to the convergence point. But this slows down training. So, a proper trade-off is required when choosing the learning rate.

The results obtained for the VGG-19 model is very close to VGG-16. This is because they have similar architecture.

Test Examples

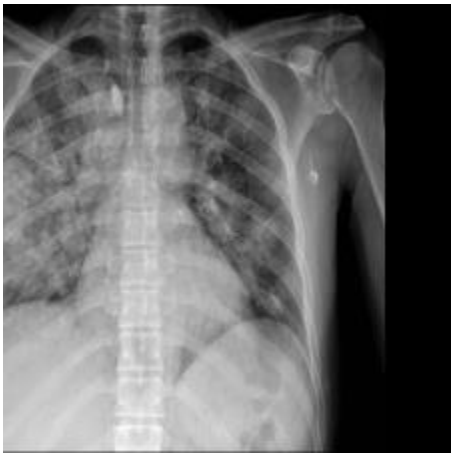
```
#During training, the second folder in dataset was Pneumonia. That is why 1 represents pneumonia positive

#During training, the first folder in dataset was COVID. That is why 0 represents covid positive


images given for testing are,

['Non-COVID-19 (7).jpg', 'COVID-19 (376).jpg', 'person3_bacteria_11.jpeg', 'IM-0003-0001.jpeg']

evaluating on VGG16 model to test for Pneumonia...

[1 1 1 0]

evaluating on VGG16 model to test for COVID-19...

[1 0 0 1]
```



Non-COVID-19 (7).jpg

pneumonia detected

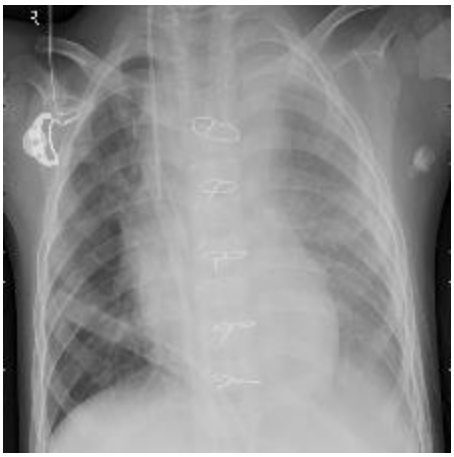
COVID-19 not detected



COVID-19 (376).jpg

pneumonia detected

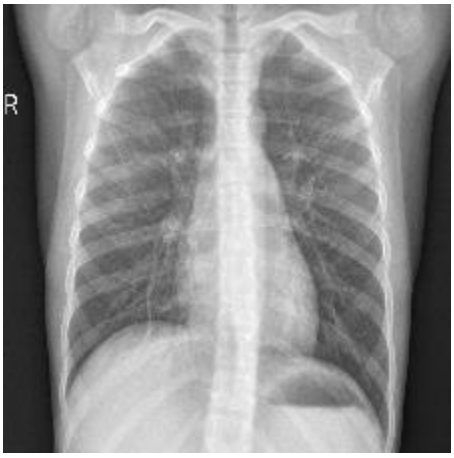
COVID-19 detected



person3_bacteria_11.jpeg

pneumonia detected

COVID-19 detected



IM-0003-0001.jpeg

pneumonia not detected

COVID-19 not detected

Conclusion:

Presence of expert radiologists is the topmost necessity to properly diagnose any kind of thoracic disease. This paper primarily aims to improve the medical adeptness in areas where the availability of radiotherapists is still limited. Our study facilitates the early diagnosis of Pneumonia & Covid-19 to prevent adverse consequences (including death) in such remote areas. The development of algorithms in this domain can be highly beneficial for providing better health-care services.

Future Plans:

COVID is a relatively new disease. So we had to work with a small dataset and rely on augmentation techniques. Our model could be significantly improved if more COVID positive chest X-rays could be collected. Furthermore, COVID is a highly mutative virus. Many new strains of the virus are plaguing the world. If the dataset could be diversified by including chest X-rays of patients infected by more diverse strains of COVID, our model should perform a lot better. In this case some hyperparameter tuning might be necessary.

Acknowledgement:

We are thankful to our course teachers for their guidelines. We would like to express our gratitude to Kaggle community to share their datasets publicly.

Reference:

1. https://ieeexplore.ieee.org/document/9243290?denied=&fbclid=IwAR3d5V3dypGfwKKMUixcEeOtXrvzphM63n1nxHZaS6eWKD7oqoT2SQsnVHg

2. https://ieeexplore.ieee.org/document/8869364?fbclid=IwAR2ynw0_hHzUzH6_EfhYJa2l8SIcreS015Fz_ocSsCDFx1amNHpEezlY_2c