CCC, IIIT Allahabad - Computer Vision Research Internship

Automatic Detection of Coronavirus Disease (COVID-19) Using Chest X-ray Images and Deep Convolutional Neural Networks

FINAL REPORT

By-Krishna Pal Deora

Building the Model

I first added **3 custom layers** to the pretrained models so that they can be trained on our dataset. The code for adding custom layers, for example, to the **ResNet50** model is shown below. The code for the rest of the models remains the same. One just needs to change ResNet50 in the first line to the name of the desired model.

```
res = ResNet50(weights="imagenet", include_top=False,
input_tensor=Input(shape=(224, 224, 3)))
outputs = res.output
outputs = Flatten(name="flatten")(outputs)
outputs = Dropout(0.5)(outputs)
outputs = Dense(2, activation="softmax")(outputs)
model = Model(inputs=res.input, outputs=outputs)
for layer in res.layers:
  layer.trainable = False
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

The images in the dataset were of different sizes. Thus, I needed to resize them to a fixed size before they can be fed to the deep learning models for training. I resized the images to a size of **224 x 224 px** which is considered to be the ideal size for the ResNet50 model. Therefore, I added the input tensor of shape **(224, 224 , 3)** to the pretrained ResNet50 model, 3 being the number of channels.

Next, I added a **Flatten** layer to flatten all our features and a **Dropout** layer to overcome **overfitting**. Finally, I added the **Dense** output layer using **softmax function** as the activation function. Since the first half of the model is already pretrained, the trainable attribute of the previous layers was set to **False**. Finally, I compiled the model with the **adam optimizer** and using **categorical crossentropy** as the **loss function**.

Training the Model

I first defined an **Image Data Generator** to train the models at modified versions of the images, such as at different angles, flips, rotations or shifts.

```
train_aug = ImageDataGenerator(rotation_range=20,
width_shift_range=0.2, height_shift_range=0.2, horizontal_flip=True)
```

Next, training of the model was performed, with all the required parameters specified as follows:

```
history = model.fit(train_aug.flow(X_train, y_train,
batch_size=32),validation_data=
(X_test,y_test),validation_steps=len(X_test) / 32,
steps_per_epoch=len(X_train) / 32, epochs=500)
```

As you can see, I have trained the model for **500 epochs** with a batch size of **32 images**.

Making Predictions

Predictions were generated by running the trained models on images of the test set. The predictions for the first 10 images of the dataset were plotted as shown below:

```
y_pred = model.predict(X_test, batch_size=batch_size)
prediction=y_pred[0:10]
for index, probability in enumerate(prediction):
  if probability[1] > 0.5:
    plt.title('%.2f' % (probability[1]*100) + '% COVID')
  else:
    plt.title('%.2f' % ((1-probability[1])*100) + '% NonCOVID')
  plt.imshow(X_test[index])
  plt.show()
```

The following snippet shows the plots of first 10 predictions of Chest X-rays:

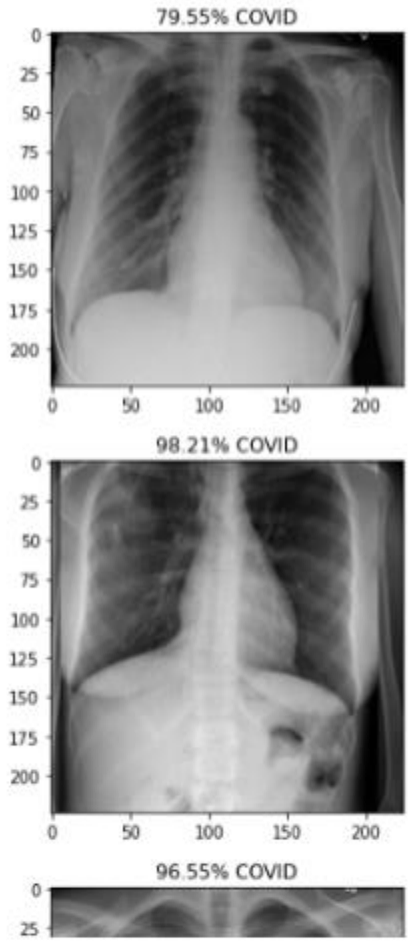

Fig 2: Visualizing first 10 predictions of Chest X-rays

Evaluation & Results of Machine Learning

Following are a few important results and plots that help estimate the accuracy of the models and get insights their performance.

Sample output of test set images

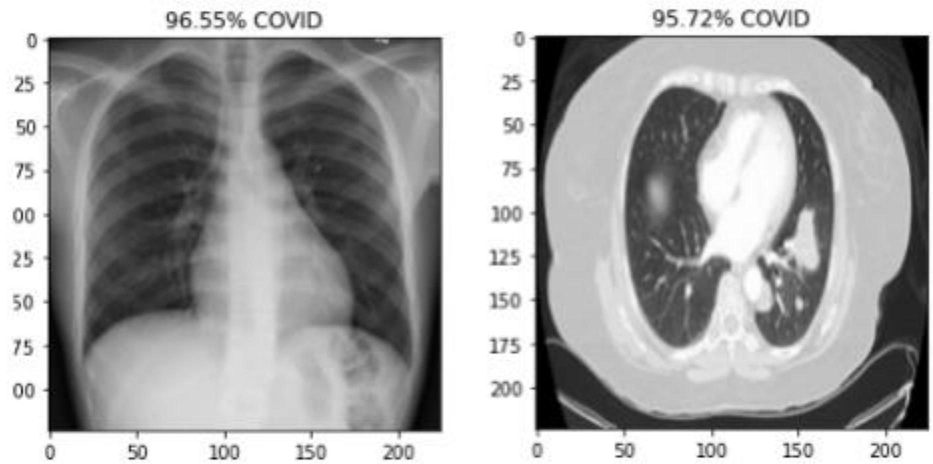

Fig 3: Sample output of test set images

Classification Report

Classification Reports of VGG16 model for Chest X-rays and CT scans

	precision	recall	f1-score	support
0	0.88	0.99	0.93	87
1	0.99	0.88	0.93	101
accuracy			0.93	188
macro avg	0.93	0.93	0.93	188
weighted avg	0.94	0.93	0.93	188

	precision	recall	f1-score	support
0	0.93	0.93	0.93	70
1	0.94	0.94	0.94	80
accuracy			0.93	150
macro avg	0.93	0.93	0.93	150
weighted avg	0.93	0.93	0.93	150

Fig 4: Classification Reports of VGG16 model for Chest X-rays (Left) and CT scans (Right)

Classification Reports of InceptionV3 model for Chest X-rays and CT scans

	precision	recall	f1-score	support
0	0.92	0.99	0.96	87
1	0.99	0.93	0.96	101
accuracy			0.96	188
macro avg	0.96	0.96	0.96	188
weighted avg	0.96	0.96	0.96	188

	precision	recall	f1-score	support
0	0.88	0.97	0.93	70
1	0.97	0.89	0.93	80
accuracy			0.93	150
macro avg	0.93	0.93	0.93	150
weighted avg	0.93	0.93	0.93	150

Fig 6: Classification Reports of InceptionV3 model for Chest X-rays (Left) and CT scans (Right)

Confusion Matrix

Confusion Matrix of VGG16 model for Chest X-rays and CT scans

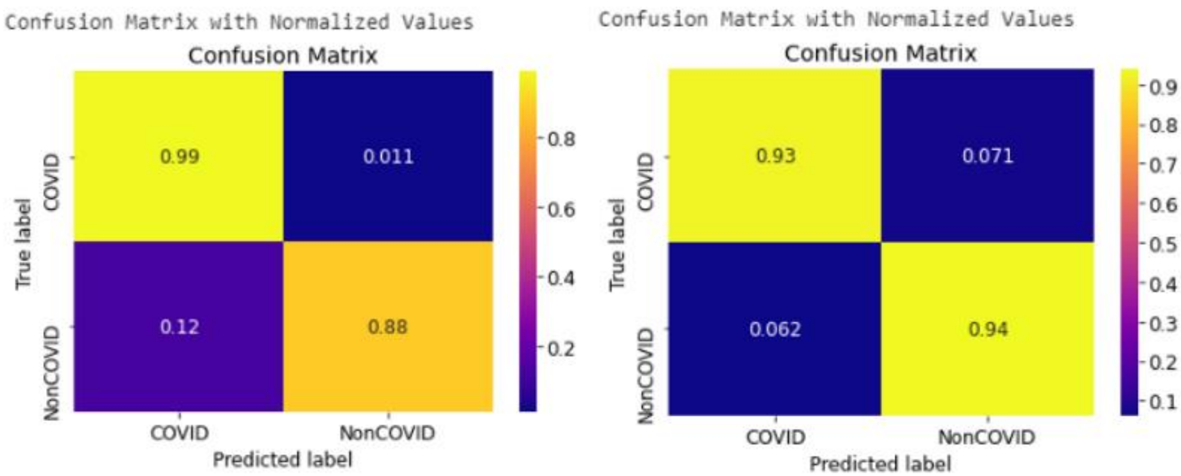

Fig 8: Confusion Matrix of VGG16 model for Chest X-rays (Left) and CT scans (Right)

Confusion Matrix of ResNet50 model for Chest X-rays and CT scans

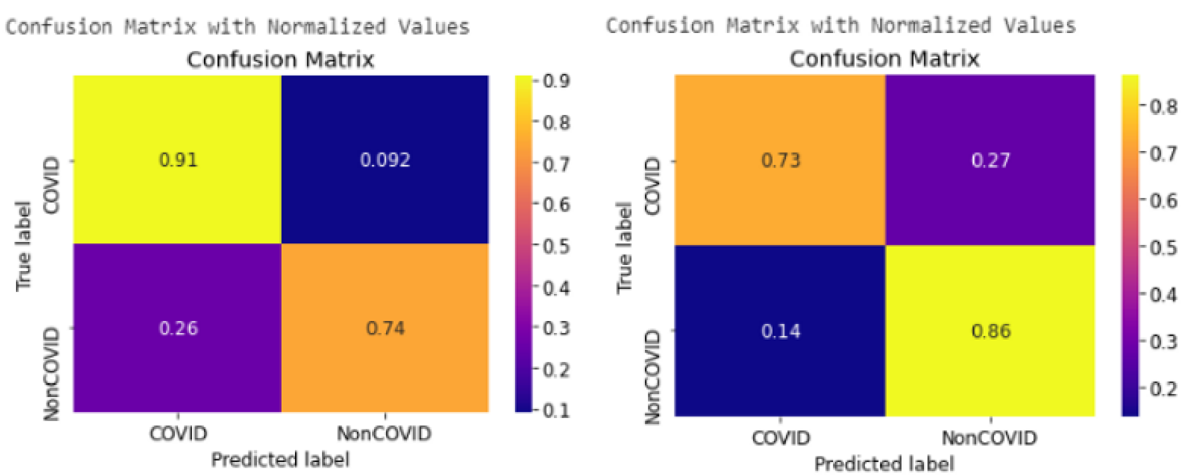

Fig 9: Confusion Matrix of ResNet50 model for Chest X-rays (Left) and CT scans (Right)

Confusion Matrix of InceptionV3 model for Chest X-rays and CT scans

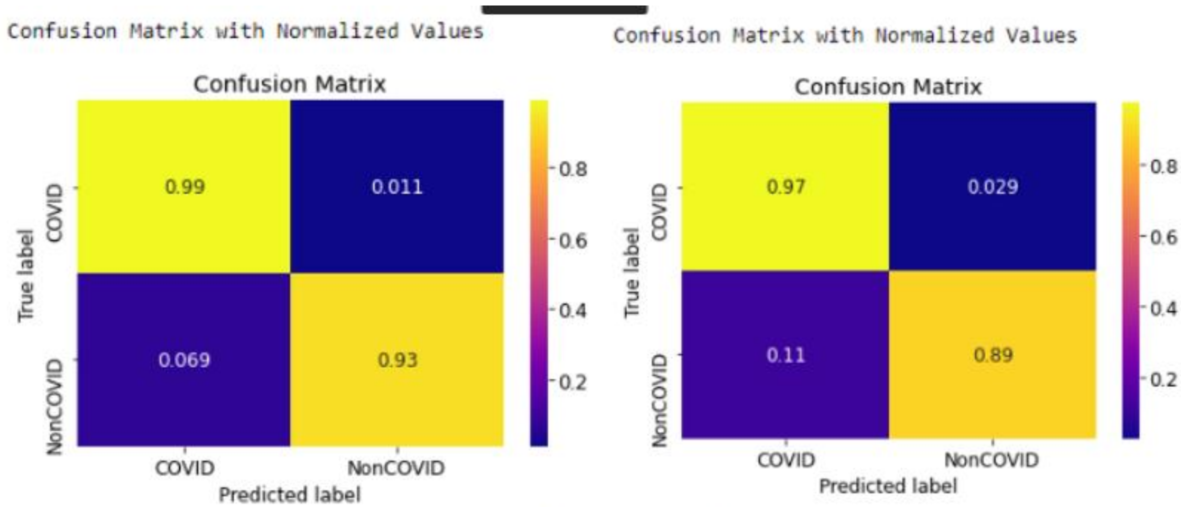

Fig 10: Confusion Matrix of InceptionV3 model for Chest X-rays (Left) and CT scans (Right)

Confusion Matrix of Xception model for Chest X-rays and CT scans

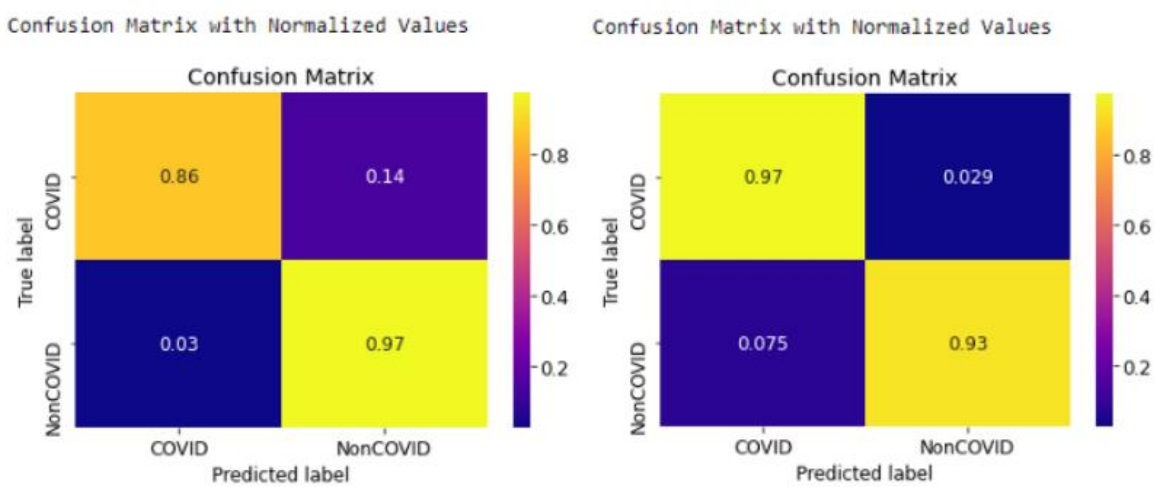

Fig 11: Confusion Matrix of Xception model for Chest X-rays (Left) and CT scans (Right)

Source Code

The source code for the entire project along with the datasets and models a are available on my GitHub repo :

https://github.com/krishnapalS/Automatic-Detection-of-Coronavirus-Disease-COVID-19-Using-Chest-X-ray-Images-and-Deep-CNNs

Conclusion

In conclusion, I would like to throw light on the fact that the analysis has been done on a limited dataset and that the results are preliminary. Medical validations have not been done on the approach and hence the results might differ from those observed in practical use cases.

In future, I plan to improve the performance of the models by training them on more images and possibly including other factors like age, nationality, gender, etc.