COVID19 Detection

The aim of this project was to compare state-of-the-art deep learning architectures for image classification to determine the model architecture best suited for X-Ray images. At the beginning of the pandemic, various attempts were made to detect the presence of the COVID19 virus from X-Ray scans of the lungs. This project revolved comparing the various architectures to determine the best architecture for classification on X-Ray images.
This problem was further divided into a binary classification and a multi-class classification problem. THe binary problem distinguished X-Ray scans into two categories - COVID19 patient and healthy. The multi class problem being slightly more intricate, tried to differentiate COVID19 patients, patients suffering from Viral Pneumonia, Patients suffering from bacterial Pneumonia and people who are healthy. The data used was collected by Adrian Xu, combining the Kaggle Chest X-ray dataset with the COVID-19 Chest X-ray dataset collected by Dr. Joseph Paul Cohen of the University of Montreal.

For the binary problem, I worked on designing and training my own model. In addition to that I used various state-of-the art models pretrained on the imagenet ImageNet dataset. For the binary problem, the various architectures I used were VGG16, VGG19 and ResNet50. After evaluating baselines for all these models, I decided to go ahead and tune the VGG16 models as it had the best performance compared to the complexity of the model. I tuned various hyperparameters of the model such as the learning rate, the activation functions, the dropout parameters and the number of epochs. For the parameter tuning, I used sklearn's GridSearch method with 3 fold cross validation. I also generated a t-Distributed Stochastic Neighbor Embedding (tSNE) plot to visualize the lower dimension representation of the data from my first dense layer.

Inferences: The best model, the tuned VGG16 model had a training and validation accuracy of over 95%. The model I created myself performed fairly well (70% + on both training and validation) considering the size of the model, and the small dataset. This model could have been further improved by using a much simpler network and more data points. The VGG19 did have similar results compared to VGG16 however, it had 26% more parameters than the VGG16. The Resnet50 had very good training accuracy, the best among all the models, but it had very low validation accuracy. This could be due to the fact that architectures based on residual blocks have been known to be data hungry and our dataset was fairly small for the size of the model.

For the multi-class classification problem, I started by creating my own model followed by using state of the art models with different architectures. The models I used for this problem were the VGG16 model, InceptionResnetV2, Resnet50 and Xception models. I chose these models due to the difference in their architectures. The model I created was a simple model with a few Convolution layers followed by pooling layers. The VGG16 has a similar linear architecture but significantly more parameters. The ResNet50 is a model that is built on the concept of residual blocks which acts as a solution to the vanishing gradient problem in very deep networks. The InceptionResnetV2 is a model that is built using the architecture of residual inception blocks. The Xception model is similar to the InceptioResnet50 model but without the residual blocks and uses depth wise separable convolution instead of the Inception blocks. After running baselines, I decided to tune the VGG16 model further based on the performances compared to the complexity of the model. For tuning the model, I used sklearn's Grid Search method and tuned the learning rate, activation function and the dropouts. I also generated a t-Distributed Stochastic Neighbor Embedding (tSNE) plot to visualize the lower dimension representation of the data from my first dense layer.

Inferences: For this task, out of the 5 models that we used, the VGG16 again had the best baselines overall.The Resnet50 again had a very good performance on the training set(better than the VGG16) but had poor generalization on the validation set similar to the binary task. The Xception and InceptionResnetV2 model both suffered from overfitting. However both these models could have been further optimized by more aggressive dropouts. This multi class problem was significantly harder as the differences in features between the viral pneumonia and Covid 19 patients were not significant and this can be corroborated by research in the COVID19 virus.

Overall this project gave me an understanding of the various state of the art model architecture for image classification tasks. It gave me a thorough understanding of the strengths and weaknesses of all these architectures. Although the models were pretrained, I was made aware of how data hungry most deep learning models were. Another key insight that I took from this project was the cost to performance ratio for training these models.