Detection of COVID-19 affected patients from Chest X-Rays using Deep Convolution Neural Networks and Transfer Learning Approach

Artificial Intelligence I Final Project Report

Fall 2020

Datta, Debayan
UTA ID : 1001851239
dxd1239@mavs.uta.edu and

Saber, Nudrat Nawal
UTA ID : 1001733394
nxs3394@mavs.uta.edu

Abstract

The Novel Coronavirus 2019 (nCov-19) is one of the most impacting disease which brought the entire world to a standstill. The world was locked down and all forms of physical communication disappeared for a span of 6 months in different locations around the globe. So, the only suitable way to control the spread of this disease and bring back life to normal was rapid testing that can identify people with the disease and isolate them from normal people to avoid contagiosity. The already existing testing standards being new to people became difficult to sulk in, moreover, the technical expertise was minimum and the time taken by the virtue of the investigation was pretty high. And in this moment of emergency it became even more challenging. So, we tried to implement another way of testing for Coronavirus by identification through Chest X-rays. The already existing frameworks are really good, but always have a room for improvements and modifications that actually happen for the good. So, we implemented a baseline model as well as few transfer learning approaches to mark the beginning for our research in this area. The proposed model or the experiments are not ready for bench-marking, but on the way, we will be working for a number of suitable modifications that will finally help us derive the conclusion better.

Introduction

Now, as we already understand what is already happening with the ever changing world absorbing the new considerations that were listed by the outbreak of the deadly Coronavirus disease and imagine the consequences that we need to face in the coming days, we anyway had to bring life to normal by adding this disease to the already existing and semi-life threatening diseases. So, the first step to do so can be by trying to detect this by the already existing chest disease detecting framework like Chest X-Rays. The lungs as a part of the entire chest region develops a number of complications and changes due to the disease that can be identified by trained radiologists who have their expertise in this area. Now, being some budding researchers in the field of Artificial Intelligence, it was very well necessary to bring about some changes in the conventional X-ray identification by using Convolutional Neural Network and also the help of pre-trained model that could classify the chest x-rays using the image classification techniques that are commonly used by different researches to classify other objects.

So, we basically identified the affected and normal x-rays and then with the help of a baseline convolutional deep neural network to classify the x-ray images. As a convolutional neural network uses very sleek ideologies like probabilities and max pooling, it was absolutely perfect to begin the research. The examples of both affected X-rays and Normal X-rays are given below: Then we basically decided to use

Figure 1: Affected X-Ray

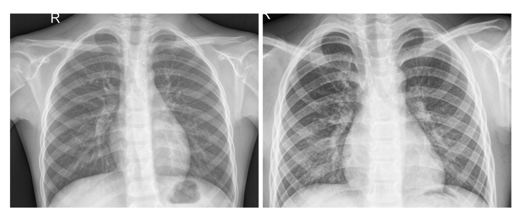

Figure 2: Normal X-Ray

the pre-trained models on the data to give the outputs in a better way.

All these classification tasks that we mentioned above was impossible without proper data that in fact drives most of the task. So, in the phase of the data acquisition, we basically depended on the crowd resources publicly available datasets and did the pre-processing to finally get our desired data.

The dataset is an amalgamation of different inputs from Kaggle and GitHub. We have collected a set of Normal Chest

X-Rays from The Kaggle Chest X-Ray Images (Pneumonia)(Kermany, Zhang, and Goldbaum 2018) Dataset, URL: https://data.mendeley.com/datasets/rscbjbr9sj/2 and the affected patients X-Rays from COVID-19 Image Data Collection(Cohen et al. 2020) Dataset, URL: https://github.com/ieee8023/covid-chestxray-dataset. The acquired dataset was then divided into training and validation datasets. As we have nearly 6000 Normal X-Ray Samples, so we decided to randomize the sample space and then get the first 600 Images. The COVID-19 Dataset is a public sourced dataset and still increasing in number. During the compilation of the project report, there were 196 Samples. So, we took all the samples for our classification. Later we have plans to look into the class imbalance issue. All the images were finally divided into two sets in a ratio of 70:30.

As soon as the collection and stocking of the dataset of confirmed, we started looking for different other researches that were ongoing in this field as well as already accomplished results and bechmarkings in the field.

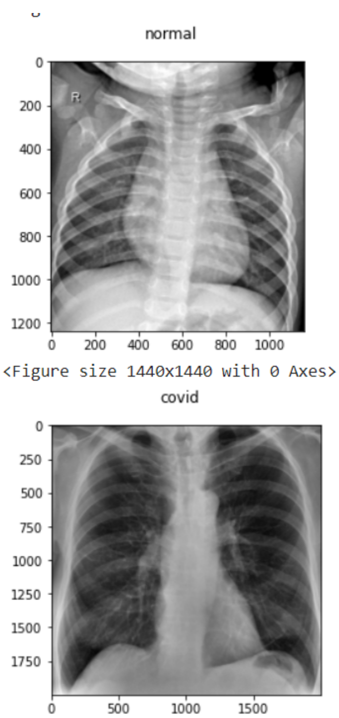

Figure 3: Example of dataset

A number of quality rich research works (few listed in the reference section) have already been performed in the past few months and gave awe-inspiring results. But, there is always room for improvement of the previously used strategies and developments that can be carried out in the existing systems to produce even more fruitful outcomes. That's what encouraged us to build a model using Deep Convolution Neural Networks and Transfer Learning Approaches to detect COVID-19 affected patients using radiological imaging, primarily X-Rays. We have pointed out a number of techniques in Transfer Learning Approaches as well as Deep Learning that can have credible outputs. For example, ResNeXt, etc.

Literature Review

Researchers around the world are trying to detect coronavirus by building a model that is not only time-efficient but also cost-effective. In this section, we will be trying to present the recent existing works on COVID-19 classification and describe the characterizations in short. Apostolopoulos and Mpesiana(Apostolopoulos and Mpesiana 2020) collected a 1427 thoracic x-ray scan and proposed a model using the strategy of transfer learning to perform feature extraction. They pre-trained the CNN model for feature extractor. They performed tenfold cross-validation for training and evaluation procedures. Though they finally got a remarkable result, they didn't focus to handle the transfer that is negative. In the paper(Wang and Wong 2020), the authors proposed a diverse and selective long-range connectivity method from CXR images. Some improvement of sensitivity and PPV to COVID-19 infections need to be done as new data is collected. Ozturk et al.(Ozturk et al. 2020) proposed a 19-layer CNN model for the binary classification of COVID-19 .

They designed DarkCovidNet architecture by using the

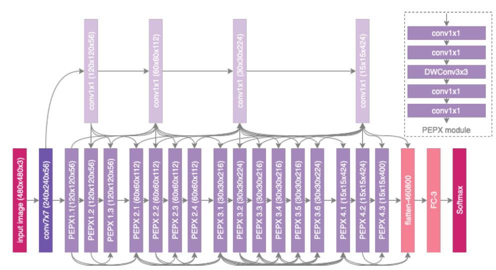

Figure 4: CovidNet Architecture(Wang and Wong 2020)

Darknet classifier that is based on a real time object detection system YOLO. Rach of the convolutional layer was developed by batch normalization and LeakReLU operation.87.02% accuracy for multiclass cases and 98.08% accuracy for binary classes were achieved by the authors. The authors from the paper(Narin, Kaya, and Pamuk 2020) worked on some pre-trained models ResNet50, InceptionV3 and Inception ResNetV2. They used transfer learning to overcome the time to train and deficiency as they used small a set of data. They input the images into pre-trained models incorporating transfer learning. Then they employed Global Average Pooling, Fully Connected Layer with ReLU in the training phase. Finally, they connected the layer fully with softmax .Accuracy of 97%, 98% and 87% was achieved by performing InceptionV3, ResNet50 and Inception-ResNetV2 architecture respectively. The model generated some overfitted data. Zhang et al.(Zhang et al. 2020) proposed a general flatten and fully connected convolutional network with a PEPX module((conv1×1 + conv1×1 + DWconv3×3 + conv1×1 + conv1×1).It got 93.3% accuracy but, the long-range connections in the DNN gave rise to memory over-

head.
The authors from paper (Pandit and Banday 2020) considered around 1500 X-ray report data which were defined as COVID positive, bacterial pneumonia, or stable condition. They applied a pre-trained model of VGG-16 over the dataset for classifying and labeling. They achieved 96% and 92.5% accuracy over a few production classes. Researchers tried to improve the model with the use of Artificial Intelligence. Where (Das, Santosh, and Pal 2020) used an AI-based model named CovidAID. The authors used a deep neural network model over a public dataset of chest X-rays for detecting COVID. The model resulted in 90.5% accuracy.
As for the scarcity of test data, researchers tried to use similar data which can be regarded as identifying COVID. Also, they tried to apply new algorithms that can increase the overall performance. Researchers in (Basu and Mitra 2020) applied transfer learning with Gradient class activation Cam for detecting the virus. The authors used 20000 X-rays of 4 categorical data which include stable, Pneumonia, COVID, and other diseases. Applying 5 fold cross-validation COVID and stable datasets provided with 95.3% overall accuracy where the other 2 datasets resulted in misclassification. Authors in (Minaee et al. 2020) used four kinds of pre-trained CNN models. Which includes, ResNet18, ResNet50, SqueezeNet, and DenseNet-161 over COVID 19 and non-COVID 19 data. They had 14 subclasses consisting CheXPert image dataset. Which resulted in a sensitivity range of 97.5%.

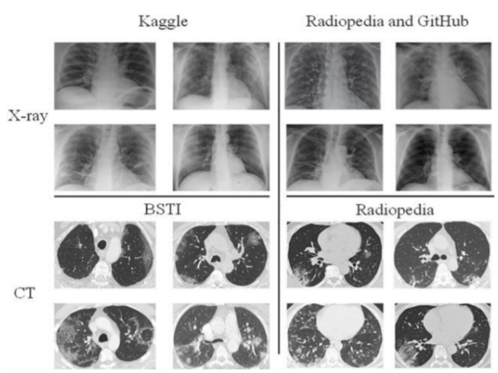

Figure 5: Samples from Datasets used by (Maghdid et al. 2020)

A generative adversarial network was constructed on research (Maghdid et al. 2020) with deep transfer learning. Here the authors used limited x-ray image data for 4 different groups. 3 transfer learning models were selected. Different conditions were applied where 3 groups, 3 groups, and 2 groups were selected accordingly. Among all the models GoogleNet performed the best in 2 conditions AlexNet performed better in one condition. where the accuracies were as follows. 80.6% (4 classes with GoogleNet). 85.62% (3 classes with AlexNet), 99 100 % (2 classes with GoogleNet). Authors from (Hemdan, Shouman, and Karar 2020) used the CovidXNet model over 50 equally divided COVID positive and Negative patients' chest X-ray They achieved 90% accuracy over the small dataset. Also in research [79], the authors use a similar structure of the dataset with RestNet50 SVM. They achieved 93.28% accuracy.

Problem Statement

“COVID-19 —The pandemic of the $21^{st}Century''$

The COVID-19 pandemic is continuing its devastating effect on earth and causing global health crisis of our time(Wang and Wong 2020). World's scientific community is working relentlessly to prevent the spreading of the virus.
One of the most important steps is identifying the patients diagnosed with COVID-19, just to stop the spreading. This resulted in the developing new and advanced detection methods like the Anti-gen Test, RT-PCR Test, etc. But, the readiness of testing can also be enriched by using conventional methods for the detection of lung diseases like Chest X-Rays.
In the Computer Science field, the researchers are using Artificial Intelligence, Data Science, etc. approaches and trying to make computational tools to detect, diagnose, and treat the world. Recent findings suggest that using the application of advanced Artificial Intelligence(AI) techniques and chest X-rays images help clinicians and radiologists in detecting Coronavirus.
The already achieved accuracy points from the existing researches were pretty high but the approaches that were seen in this researches were unconventional techniques. We infact tried to do it with conventional and old-school techniques that may also have potentials.

Problem Solution

The project that we have decided to work on is itself a challenge, only because the dataset that is required to do a suitable classification or prediction is hard to accumulate. The internet stores a world of data, but most of it is scattered. To accumulate the required data with meaningful information was our challenge, and that is where we invested most of the time. The dataset was mostly scrapped from Open GitHub repositories and Kaggle Dataset.
So, basically below is the detailed exposure of the formal steps of implementation:

Data Pre-processing : Data, which in this case are images of X-rays needs to be pre-processed because, these images were being resourced by the crowd who had different standards while recording. Most of which was not necessary for the classification. Then, all the data that we collected being X-Rays that were taken from different views, we filtered on the Posteroanterior (PA) view of all X-Rays, because this was something that could give a lot of points of relation. All these x-rays were classified under different classes of the disease. As we primarily focused on COVID-19 affected patients, we extracted only the COVID-19 positive X-rays. Then we noticed that all these images are in a different color format as well as dimensions. So, we converted all of them into a similar color format and an image of 224 x 224. To be precise, we converted the images to RGB and then finalized the images. Then, we noticed that there were a number of images that had blue filters, noise and other minute causes that result in misclassification. So, as a measure to avoid that, we discarded the images with blue filters, and finally removed the noise from the images. This noise removal was mostly the Gaussian Blur that existed in the images. There are many other pre-processing techniques that we plan to implement in the future. Finally, we created separate folders for the images and grouped them as Covid-19 Affected and Normal X-rays.

Removal of Outliers : Outliers in case of images are nothing but the area that are responsible to distract the model to give undesired results. In our case, outliers were the outer rib cage that was present in all the X-rays, and if at all we would segment the images, the rib cages constituted of another class of existence which deviated the analysis. So, we cropped the images in such a way that most of the sections that were present in all the immages were cropped out.

Image Segmentation : Image segmentation is basically the process that partitions the image into multiple segments depicting different regions in the image. This is a very essential phase in case of classification using deep convolutional neural networks. And this is because, if the image have not segments, it cannot understand the importance of the layers in the images. In most of the classification, the basic information is being conveyed by the foreground, and the background is neglected.
To achieve this we have a pool of techniques, and some of which lies in the core understanding of Convolutional Neural Networks, but the already existing conventional modes of image segmentation are pretty good and convey great information. We decided to go forward with the old school techniques because, our images are X-rays images, which doesn't have a lot of logic to segment. This can easily be segmented with the placement of the lungs and the bones. The methods that we applied to segment our images are given below:

1. **Histogram Analysis** : An image histogram is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.
   Image histograms are present on many modern digital cameras. Photographers can use them as an aid to show the distribution of tones captured, and whether image detail has been lost to blown-out highlights or blacked-out shadows.[2] This is less useful when using a raw image format, as the dynamic range of the displayed image may only be an approximation to that in the raw file.Histogram Analysis is done with the help of intensity values of each pixels that formally constitute the image. This histogram analysis is done to identify the image curves which segment the images. The threshold that separates the two segments can be observed. If there exist two types of values from different pixels, then we can conclude that there exist two segment that can be identifies with the help of thresholding. The image histogram pattern is given below: From this diagram, we decided the value of the threshold

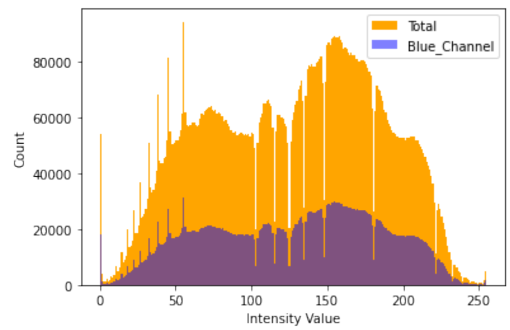

Figure 6: Image Histogram of a Random X-ray sample

   that separates the two areas of the image.

2. **Thresholding** : The threshold represents the value above which a data point is considered in the positive class. If we have a model for identifying a disease, our model might output a score for each patient between 0 and 1 and we can set a threshold in this range for labeling a patient as having the disease (a positive label).
   Thresholding is the simplest method of image segmentation and one of the oldest too. The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity $I_{i,j}$ is less than some fixed constant T (that is, $I_{i,j} < T$, or a white pixel if the image intensity is greater than that constant. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming completely white. There are multiple types of thresholding that can be used to segment the images. In all the procedures we only used the invert thresholding, only because, in the x-ray images, the darker regions were the most informative regions, which could only be highlighted in the inverse markings. The examples of this thresholdings are given below:

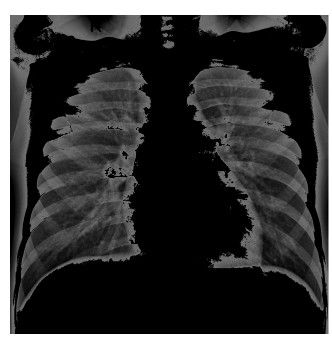

Figure 7: Type-I Thresholding

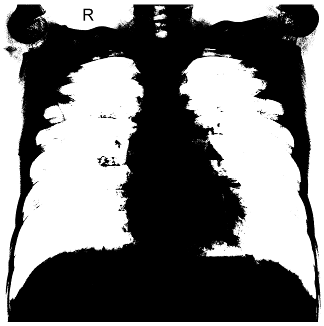

Figure 8: Type-II Thresholding

3. **K-Means** : K-means is a clustering algorithm that clusters similar attributes into a same cluster and groups them together. So, we could implement this clustering algorithm to segment the images as the different sections can be acted as clusters and pointed by the algorithm. K-means clustering is one of the most commonly used clustering algorithms. Here, k represents the number of clusters. In our case we considered k value, 2 and 3. The below images depict the same.

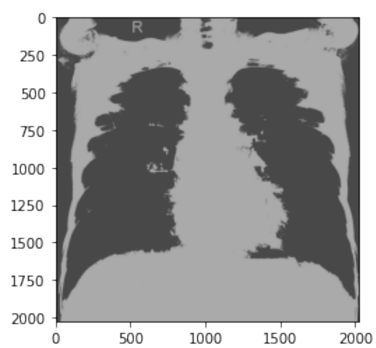

Figure 9: K-means with k = 2

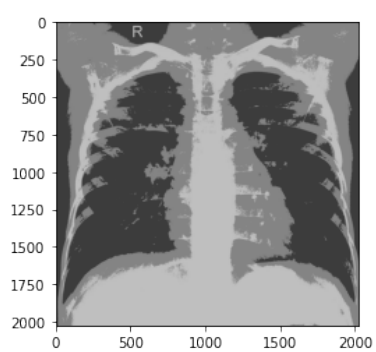

Figure 10: K-means with k = 3

We haven't explored any other segmentation techniques other than these. But have few in the list which we are going to implement in the near future.

Data Augmentation : Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a machine learning model. It is closely related to oversampling in data analysis. In our case, we anyway had to augment the data due to the lack of dataset that could be fetched form the public resourced repositories. So, what we basically did was:

1. Rescale : We re-scaled the images to a canvas of 256 px. This helped us to deal with multiple canvas size and have better augmented results.
2. Zooming : We zoomed to a power of 2 and fetched results.
3. Brightness : We distributed the images to a range of brightness levels to generate different images.
4. Horizontal Flip : We flipped the images horizontally to generate a whole new batch of augmented images. The example is given below:

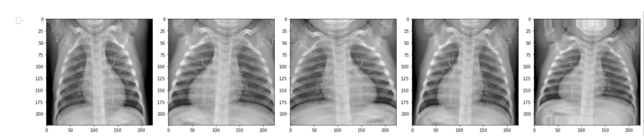

Figure 11: Augmented Images

Finally, it was time to implement the CNNs that would give the results. So, to begin, we developed a baseline convolution neural network that 4 convolution layers.

Baseline CNN : Computer vision is evolving rapidly day-by-day. Its one of the reason is deep learning. When we talk about computer vision, a term convolutional neural network(abbreviated as CNN) comes in our mind because CNN is heavily used here. Examples of CNN in computer vision are face recognition, image classification etc. It is similar to the basic neural network. CNN also have learnable parameter like neural network i.e, weights, biases etc.
Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field(it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then we slide the filter over the next receptive field of the same input image by a Stride and do the same operation again. We will repeat the same process again and again until we go through the whole image. The output will be the input for the next layer. Our Model had 4 Convo + ReLU Layers.
Pooling layer is used to reduce the spatial volume of input

image after convolution. It is used between two convolution layer. If we apply FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive and we don't want it. So, the max pooling is only way to reduce the spatial volume of input image. In the above example, we have applied max pooling in single depth slice with Stride of 2. You can observe the 4 x 4 dimension input is reduce to 2 x 2 dimension. We employed Max-Pooling at every layer interactions to give the intermediate output of each layer.
Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different category by training. We had a 64 channel FC Layer that connects all the features after flatting them and passing it to the softmax layer.
Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification. Finally, we passed it through a softmax Layer to give the final output of two label classification namely, Covid-19 and Normal. The architecture of our baseline model is given below:

Figure 12: Baseline Convolutional Deep Neural Network Architecture

This was really not enough to give a final say on the classification. So, we decided to implement the transfer learning approach to have an ordered classification.

Transfer Learning : Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in skill that they provide on related problems. We basically implemented, two approaches to classify the images out of the range of available ones. This was only because of the short time capacity we worked on the project. We have thoughts to implement better ones in the future.

1. **ResNet50** : ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8 x 10^9 Floating points operations. It is a widely used ResNet model and we have explored ResNet50 architecture in depth.
2. **InceptionResnetV2** : Inception-ResNet-v2 is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 164 layers deep and can classify images into 1000 object categories, such as the keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 299-by-299, and the output is a list of estimated class probabilities.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 13: ResNet50 Architechture

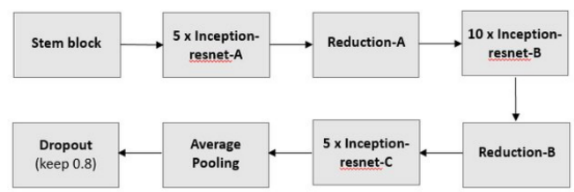

Figure 14: Basic Architecture of InceptionResnetV2

We, were able to derive decent outputs from the experiments conducted using these algorithms.

Experiments Performed

As we finally decided over the algorithms and the models that we were using in this project, we started the experiments by the first implementation of the baseline CNN. We performed all the basic segmentation and finally augmented the data to prepare for the final set of input that were images of 224 x 224 pixels. Figure 15 is the curve that we obtained for training and validation accuracy.

Then, we implemented the transfer learning approaches and we were able to derive the following accuracy points. The table below contains the accuracy of the baseline approach too.

Model	Training Accuracy	Validation Accuracy
Baseline CNN	88.87%	93.75%
ResNet50	82.39%	86.61%
InceptionResnetV2	81.95%	79.47%

Table 1: Accuracy Table

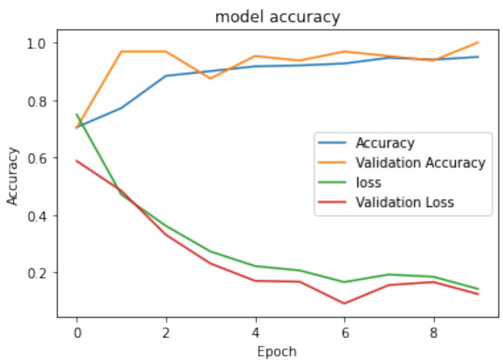

Figure 15: Training (Accuracy , Loss) vs. Validation (Accuracy , Loss) of Baseline CNN

Also, the bar plot of the accuracy that marks the trend of the drop and hike is given below:

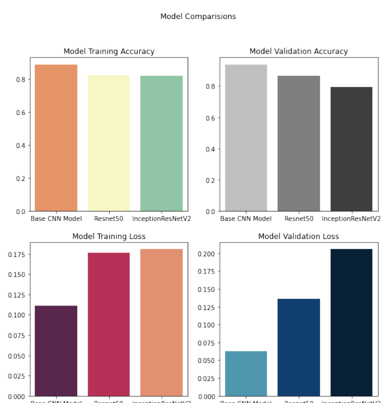

Figure 16: Barplot of the Accuracy Points

Conclusion

As a conclusion, we can only say that, the range of experiments that we were able to conduct in this short time span cannot declare this projects finding in a final proposed state. This project surely has great potential if the research is being carried out forward, but at this point of time the proposed model has minimum credibility to replace any working framework or it cannot be used for benchmarking.
The experiments performed desirably, but the accuracy was not so great. This was mostly because of the Pooling function that we trained after the transfer learning. Also, the segmentation techniques used by us are outdated and there exists state of the art techniques currently that can improve the segmentation process. This could not be explored due to lack of knowledge.
There exists a lot of possible outcomes as a scope of this project in future, which we will adapt to have a superior conclusion to the statements we proposed in this report.
But in a satisfactory note, we were able to classify the images almost correctly and finally generate a heat map that concludes the classification task.
This project helped us a lot to learn about the disease as well as methods of its identification. Also, a lot of image processing techniques were being cultured. An exempler heat map is given below:

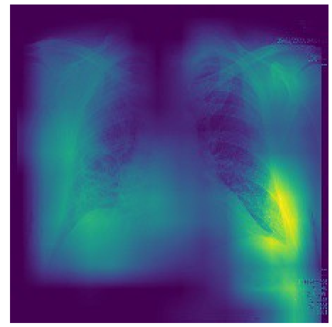

Figure 17: Heat Map of Predicted Covid X-Ray

References

Apostolopoulos, I. D., and Mpesiana, T. A. 2020. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. *Physical and Engineering Sciences in Medicine* 1.

Basu, S., and Mitra, S. 2020. Deep learning for screening covid-19 using chest x-ray images. *arXiv preprint arXiv:2004.10507*.

Cohen, J. P.; Morrison, P.; Dao, L.; Roth, K.; Duong, T. Q.; and Ghassemi, M. 2020. Covid-19 image data collection: Prospective predictions are the future. *arXiv preprint arXiv:2006.11988*.

Das, D.; Santosh, K.; and Pal, U. 2020. Truncated inception net: Covid-19 outbreak screening using chest x-rays. *Physical and engineering sciences in medicine* 43(3):915–925.

Hemdan, E. E.-D.; Shouman, M. A.; and Karar, M. E. 2020. Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*.

Kermany, D.; Zhang, K.; and Goldbaum, M. 2018. Labeled optical coherence tomography (oct) and chest x-ray images for classification. *Mendeley data* 2.

Maghdid, H. S.; Asaad, A. T.; Ghafoor, K. Z.; Sadiq, A. S.; and Khan, M. K. 2020. Diagnosing covid-19 pneumonia from x-ray and ct images using deep learning and transfer learning algorithms. *arXiv preprint arXiv:2004.00038*.

Minaee, S.; Kafieh, R.; Sonka, M.; Yazdani, S.; and Soufi, G. J. 2020. Deep-covid: Predicting covid-19 from chest x-ray images using deep transfer learning. *arXiv preprint arXiv:2004.09363*.

Narin, A.; Kaya, C.; and Pamuk, Z. 2020. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*.

Ozturk, T.; Talo, M.; Yildirim, E. A.; Baloglu, U. B.; Yildirim, O.; and Acharya, U. R. 2020. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in Biology and Medicine* 103792.

Pandit, M. K., and Banday, S. A. 2020. Sars n-cov2-19 detection from chest x-ray images using deep neural networks. *International Journal of Pervasive Computing and Communications*.

Wang, L., and Wong, A. 2020. Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *arXiv preprint arXiv:2003.09871*.

Zhang, J.; Xie, Y.; Li, Y.; Shen, C.; and Xia, Y. 2020. Covid-19 screening on chest x-ray images using deep learning based anomaly detection. *arXiv preprint arXiv:2003.12338*.