Covid-19 and Pneumonia Detection from Chest X-ray Images using Deep Learning

Sai Spandan Reddy Jogannagari

Department of Computer Science, University of Central Florida, Orlando, Florida.

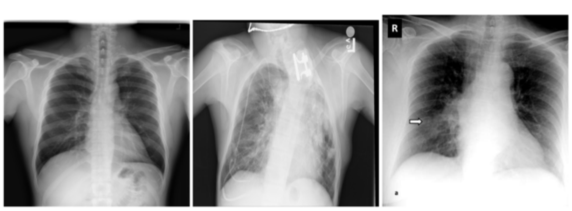
Abstract

Motivation: As both Covid -19 and Pneumonia infect human lungs causing the patient difficulty breathing it is important for doctors to detect the disease as early as possible and diagnose the patients accurately so that the infection would not spread and reduce the severity of the disease. The existing methods take a lot of time identifying the disease which could increase the severity of the disease so in this article we implement various Deep Learning Techniques which take patient's Chest X-ray images as input and predict whether the patient is Covid-19, Pneumonia, Normal and compare the accuracies of the methods implemented and choose a technique with the best accuracy.

Project Type: Implementation and Improvement of the Paper "*A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2* [1]".

1 Introduction

Covid-19 outbreak started in December 2019 and spread across the world resulting in a Pandemic. Some of the symptoms of this disease include fever, loss of taste or scent, difficulty in breathing causing Lung infection leading to the death of the patient. Whereas Pneumonia is an infection that inflames the air sacs in one or both lungs causing for severe cough, and difficult breathing. There were cases where the doctors wrongly diagnosed patients with pneumonia whereas the patient is infected with Covid-19 which they identified later and resulted in the cause of death of the patients. This happened because the existing methods of identification of the disease which is RT-PCR which takes a lot of time to identify the disease which could increase the severity of the disease until we identify the disease so we need a better approach to identify the disease faster as covid infects the patients' lungs we can use chest X-rays to identify Covid-19 as they are less invasive and more widely available than RT-PCR tests. Additionally, chest X-rays can provide valuable information about the condition of the lungs and other internal organs, which can help doctors to diagnose and treat COVID-19 more effectively but categorizing Covid-19 with Pneumonia is very difficult.



A B C

fig (1) chest X-Rays of three different patients where A: - Normal B: - Pneumonia, C: - Covid 19

From the fig (1) From the normal person (A) we can clearly see the patients' lungs and heart and there were no white patches indicating the infection whereas in the image B we can see the infections is segregated in the lungs by observing the white patches in the lungs. In the Image C where the patient is infected with covid 19 the infection is spread across the lungs and the white patches indicating the infection is not clearly visible to us, it is hard for the doctors to classify them, so I propose the use of Deep Machine learning and Artificial Intelligence to be able to identify and classify the disease by taking the input Chest X Ray images.

Using computer vision and deep learning techniques many different types of diseases can be identified. Some examples include skin diseases, such as melanoma and other types of skin cancer, which can be identified using images of the skin, eye related diseases such as glaucoma and age-related macular degeneration, which can be identified using images of the eye. Additionally, many internal diseases, such as tumors and abnormalities in the heart and lungs, can be identified using medical imaging techniques, such as X-rays, CT scans, and MRIs.

Some of the related work in detection of Covid 19 from Chest X ray images are as follows Sethy and Behra [2] propose a method founded on deep features as well as support vector machine (SVM) to detect patients with a coronavirus infection based on X-ray images. Ioannis et al. [3] assess the performance of the latest CNN architectures used in recent years for medical image classification. Narin and others solved the problem of the limited supply of COVID-19 test kits accessible in public hospitals. [4] proposes to apply an automatic detection system as another rapid diagnosis recourse to avert the spread of COVID-19 and its pressure on medical institutions. All the above methods have performed well for detecting Covid-19 but have not tried together with Pneumonia images. Also these methods used same transfer learning techniques but they have not tried concatenating the features of two networks and experimenting so In this article I have built a CNN model, using transfer learning with imagenet weights on IceptionV3 network, Resnet50v2 Network, Xception Network and in the end I have concatenated the features of both Xception and Resnet50v2 and added additional layers on top of that concatenated network and compare the performance of the individual models with the concatenated network and the network proposed by in the above article[1].

2 Data

I have used 2 different datasets in this project. For the Covid 19 images I am using Covid Chest Xray Dataset from the GitHub repository [8] which is an open public dataset contains both Chest X Rays and CT Scan images of Covid -19, Other Viral and bacterial pneumonias and this data is collected from the public sources and as well as through indirect collection from hospitals and physicians and this project is approved by the University of Montreal's Ethics Committee.

This Dataset contains the total of 585 images of Covid -19 from these images I have extracted a total of 478 Covid 19 chest X Ray images and the rest of the images were of CT scan images, so I have not considered these images. From this dataset I have also extracted the 18 Normal images and 58 Pneumonia images. So, the total of 478 Covid -19 images, 18 Normal Images and 58 Pneumonia Images.

For the Normal and Pneumonia images I have used RSNA pneumonia detection challenge from Kaggle competition [9] where RSNA is an international society of radiologists, medical physicists, and other medical professionals with more than 54,000 members from 146 countries across the globe where they see the potential for Machine Learning to automate initial detection (imaging screening) of potential pneumonia cases to prioritize and expedite their review. After extracting Normal and Pneumonia images from this dataset final dataset contains the 8869 Pneumonia Images and 6070 Normal Images, and 478 Covid 19 images. All the images are converted into .png fromat.

As we are having very less images of Covid -19 compared to the Pneumonia and Normal images this leads to the class imbalance which mean when the number of examples in one or more classes of a classification problem is significantly lower than the number of examples in other classes which makes the modal biased towards the more commonly occurring classes and not be able to accurately learn and generalize to the less commonly occurring classes. This can result in poor performance on the class with fewer examples.

At first the training data is split into 8 different sets and for each set I have considered 500 unique Pneumonia images, 500 unique Normal Images and 433 Covid 19 images. so, while training we train the modal with the similar sizes So that the modal will be able to perform better for each set. And for validation purpose I choose 400 Covid -19 images, 400 Normal Images and 45 Covid -19 Images and save the modal which performs with best accuracy. Evert model is trained with 20 epochs so for each model we trained for a total of 80 epochs. Use the rest of the images which were not trained for the testing dataset so overall I use 2070 Normal Images, 45 Covid 19 images, 4869 Pneumonia images for the testing of our modal so total we have 7334 test images on which test our model performance accuracy.

3 Methods and Implementation

3.1 Convolution Neural Network:

They often used in image classification and object detection tasks, so we use these to detect COVID-19 and pneumonia in medical images as they can automatically learn features from images, which makes them well-suited for identifying patterns and abnormalities in visual data. It is composed of many small, interconnected units called neurons, which are arranged in layers. The layers of a CNN are typically arranged in an order that corresponds to the hierarchical structure of the input data, such as an image. The first layer of a CNN typically receives the raw input data, such as the pixels of an image, and successive layers extract more abstract features, such as edges and shapes. The final layer of a CNN produces the output of the network, which is a prediction or classification based on the input data. CNNs are trained by presenting them with many examples of the inputs they are expected to process, along with the corresponding outputs, and adjusting the connections between neurons in each layer to minimize the error between the predicted and actual outputs.

For this experiment I have built a convolution neural network with the image inputs of size (256,256,3) where the first two values indicate the image pixels and the later one indicates the number of channels it is using which is basically RGB channels it is using. The network is built such a way that it contains 13 layers with each layer having “ReLU” activation in each layer, filters for this modal ranging from 8 to 512, added pooling to reduce the dimensionality of the image at the end of the 13th layer my (None,8,8,512). This output is flattened and feed into a dense layer with “ReLU” activation having 512 filters it generated overall 19,010,699 trainable parameters and the modal is compiled with Nadam optimizer with a learning rate of 0.0001 a categorical cross entropy loss function and trained with a batch size of 30 and 20 epochs.

3.2 InceptionV3 Network:

Inception v3 is a specific type of CNN that was designed by Google researchers to improve upon the original Inception architecture [5]. It uses a technique called "depthwise separable convolution" to reduce the number of parameters and improve the speed of the network, without sacrificing accuracy. A depthwise separable convolution is a way of decomposing a traditional convolution into two separate operations. In a depth wise convolution, a single filter is applied to each input channel (or "depth" dimension) to produce an output channel. This allows the network to learn spatial features independently for each input channel, rather than mixing information from different channels together.

For the Inception network I have given the Input Shape of the Image as (256,256,3) which generated the output shape of (6,6,2048) with a total trainable parameter of 21,989,539. This output is flattened feed into Dense layer with a SoftMax activation with 3 classes. The modal is compiled with the Nadam Optimizer with a Categorical cross entropy loss function.

The model is trained with a batch size of 30 and while training the modal save the model with the best validation accuracy and use this model to test the model.

3.3 Resnet50v2 Network:

Before using Resnet50 I have learnt about what residual block does it consists of two or more layers, where the input to the block is added to the output of the block, as shown in the following equation

$$Y = F(X) + x$$

where x is the input to the block and F(x) is the output of the block after passing through one or more layers. The main purpose of the residual block is to enable the network to learn residual functions, or the difference between the input and the output of the block. This can help the network learn more efficiently and can also make it easier for the network to learn deeper architectures, which can improve its performance.

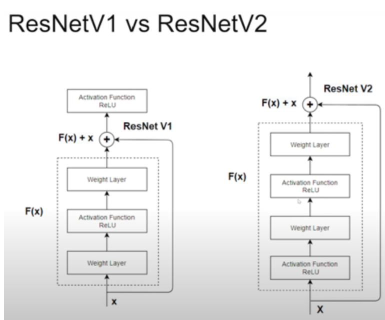


Fig (2): Comparison of ResnetV1 vs ResnetV2 Networks

ResNet50v2 is a variant of the original ResNet50 architecture [6], which uses residual blocks as a fundamental building block. These residual blocks are stacked on top of each other to form a deep network, which allows the network to learn complex and abstract features from the input images. The use of residual blocks in ResNet50V2 helps the network learn more efficiently and effectively and can also improve its performance on a wide range of vision tasks.

For the Resnet50V2 network I have followed the same approach we followed as InceptionV3 network and the Input Shape of the Image is (256,256,3) which generated the output shape of (8,8,2048) with a total trainable parameter of 23,912,579. This output is flattened feed into Dense layer with a SoftMax activation. The modal is compiled with the Nadam Optimizer with a Categorical cross entropy loss function.

3.4 Xception Network:

Xception is a variant of the Inception architecture that uses depthwise separable convolutions to reduce the number of parameters and improve performance[7]. Unlike Inception, which uses a combination of regular convolutions, average pooling, and maximum pooling, Xception uses depthwise separable convolutions in all its layers, which allows it to learn spatial hierarchies of features more efficiently. This makes it particularly well-suited for tasks such as image classification and object detection. It also has the advantage of being easier to train than Inception, thanks to its use of batch normalization and skip connections. Overall, It has proven to be a powerful and efficient architecture for deep learning tasks involving image data.

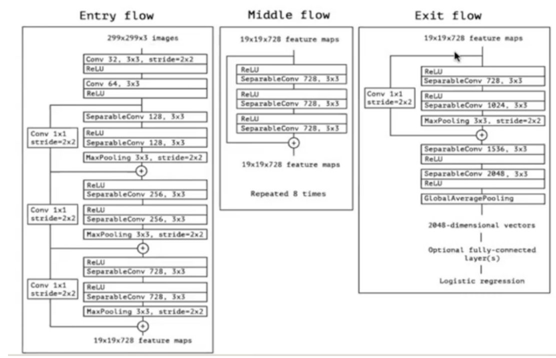


Fig (2): Structure of Xception Network

Entry flow block: This is the initial block of the network, and it typically consists of a series of convolutional layers that are used to extract low-level features from the input data.

Middle or Inception block: This is the core building block of the Inception architecture, and it consists of a series of parallel convolutional layers that are used to extract different spatial scales and combinations of features from the data.

Exit Flow block: This is the last block of the network, and it typically consists of a series of fully connected layers that are used to make predictions based on the features extracted by the inception blocks.

For the Xception network I have followed the same approach for the above two networks and the Input Shape of the Image is (256,256,3) which generated the output shape of (8,8,2048) with a total trainable parameter of 21,200,171. This output is flattened and feed into Dense layer with a SoftMax activation. The modal is compiled with the Nadam Optimizer with a Categorical cross entropy function and trained with a batch size of 30.

3.5 Concatenated Network:

For the concatenated network we extract the layers from both Resnet50V2 and Xception Network and concatenate the layers of both the network. By doing this we can improve the performance by taking advantage of the strengths of both architectures and potentially improve the performance of the model on the task, concatenation introduce additional diversity and regularization into the model, which can help to prevent overfitting and improve the generalization performance of the model. By using features from multiple networks, you can make the model more robust to changes in the input data, such as variations in lighting and pose. This can improve the performance of the model on real-world data.

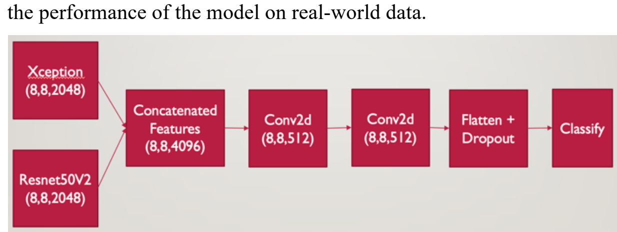


Fig (4) Structure of Concatenated Model

Fold	Network	COVID-19 Correct detected	COVID-19 Not detected	COVID-19 Wrong detected	PNEUMONIA Correct detected	PNEUMONIA Not detected	PNEUMONIA Wrong detected	NORMAL Correct detected	NORMAL Not detected	NORMAL Wrong detected
	Xception	26	5	101	3983	437	569	6245	606	378
1	ResNet50V2	27	4	96	3858	562	480	6334	517	507
	Concatenated	26	5	68	3745	675	309	6526	325	628

Fig (5): Predictions obtained in the referred Paper

	name	correct_covid	not_detected_covid	wrong_covid	correct_pneu	not_detected_pneu	wrong_pneu	correct_normal	not_detected_normal	wrong_normal
0	CNN	37	8	183	1845	575	316	4455	414	498
1	InceptionV3	39	6	38	2184	236	199	4649	220	225
2	Resnet50V2	38	7	32	2194	226	289	4565	304	216
3	Xception	42	3	35	2162	258	188	4664	205	243
4	Concatinated-Old	40	5	16	2236	184	313	4548	321	181
5	Concatinated	44	1	36	2243	177	369	4479	390	163

Fig (6): Predictions Obtained after my experiments

As you can see from the above figure 4 both the networks take the same input image shape of (256,256,3) and both the networks produces the same output shape which is of (8,8,2048) hence we were able to concatenate these two networks with resulting in the shape of (8,8,4096) in order to get the better predictions from this network I have added two additional convolution layers on top of the concatenated model and these two layers have the shape of (8,8,512) then the output of these two layers is them flatten and have added dropout of 50 percent and then feed into a last dense layer which classifies the images with SoftMax function. The learning rate of this model is .0001 and used Nadam optimizer with Categorical Cross entropy loss function. I have trained the model with 20 epochs and a batch size of 20.

At first, I have replicated the same architecture as proposed in the paper saved the model as concatenated-old model and then modified the network further and performed experiments on concatenated network. The comparison between my model and the proposed model in the paper is I have reduced the size of the input image doing so I have reduced the number of parameters in the model. This can help prevent overfitting and improve the generalization of the model on new data. Added additional Layers after the concatenation of our pretrained models so that the model can learn new features from the concatenated output of the pre-trained models reducing the number of filters and improve the overall performance of the model this also gave me the flexibility to further fine tune the model by varying the number of neurons, activation function etc.

4 Results

The testing is performed on rest of the images which were not used for training, so we have a total of 7334 images for testing of three different classes which are 45 images of Covid -19, 4869 Normal Images and 2420 pneumonia positive images.

The model performance is identified by the all the correct predictions are marked by the True Positives (TP) and all the Wrong predictions and Marked by the False Negative (FN) also calculated False positive and False negative for each class in our predictions.

Overall Accuracy = Total correct Predictions / Total Predictions
Accuracy of the Class: (TP +TN) / (TP+TN+FP+FN)
Sensitivity = Correct Class prediction/ Total class elements

From the above fig (5) indicates the results obtained for a single fold in the original paper and fig (6) we can observe the predictions obtained for the total 7334 images and classified into three categories into correct predictions, wrong predictions and not detected predictions. And using these predictions I have calculated the accuracies. In the Fig (8) below I have displayed the accuracies obtained in my experiments.

For the CNN model in the Fig (8) have the least accuracy with 86% among the rest of the models given the model is trained on just 13 layers this accuracy could be improved if I had increased the number of layers but that would increase the number of parameter and the model would take a lot of time to train.

Comparing the percentage of accuracy except CNN all the rest of the models have outperformed the previous accuracies represented by the author in the paper[1] comparatively whereas in Xception Model which I have designed has the best accuracy overall but it is not able to detect the Covid-19 and pneumonia samples properly compared to the concatenated models, but the difference comes with the normal image predictions as it has predicted normal images more accurately and the number of normal images is more compared to the other classes.

The Concatenated model - old has performed similar as Xception network but it has outperformed the Xception network in detecting both Covid-19 and Pneumonia Accuracies and If we observe the specificity of this model, it has the highest specificity comparatively to all other models. The concatenated model which I have further modified is able to predict the Covid

Fold	Network	Accuracy	COVID-19 Sensitivity	PNEUMONIA Sensitivity	NORMAL Sensitivity	COVID-19 Specificity	PNEUMONIA Specificity	NORMAL Specificity	COVID-19 Accuracy	PNEUMONIA Accuracy	NORMAL Accuracy
	Xception	91.31	73.35	88.95	92.91	99.55	93.17	89.63	99.48	91.52	91.62
Average	ResNet50V2	89.79	74.02	85.54	92.60	99.33	92.98	86.64	99.26	90.07	90.25
	Concatenated	91.40	80.53	87.35	94.06	99.56	94.32	88.09	99.50	91.60	91.71

Fig (7): Results obtained in Paper

	name	Modal_Accuracy	covid_Sensitivity	Pnemonia_Sensitivity	Normal_Sensitivity	covid_Specificity	Pnemoia_Specificity	Normal_Specificity	Covid_Accuracy	Pnemonia_Accuracy	Normal_accuracy
0	CNN	86.405781	82.222222	76.239669	91.497227	97.489368	93.569394	79.797160	97.395691	87.851104	87.564767
1	InceptionV3	93.700573	86.666667	90.247934	95.481618	99.478666	95.950346	90.872211	99.400055	94.068721	93.932370
2	Resnet50V2	92.677938	84.444444	90.661157	93.756418	99.560982	94.118844	91.237323	99.468230	92.977911	92.909735
3	Xception	93.646032	93.333333	89.338843	95.789690	99.519824	96.174196	90.141988	99.481865	93.918735	93.891464
4	Concatinated-Old	93.046087	88.888889	92.396694	93.407270	99.780491	93.630444	92.657201	99.713662	93.223343	93.155168
5	Concatinated	92.255250	97.777778	92.685950	91.990142	99.506105	92.490842	93.387424	99.495500	92.555222	92.459776

Fig (8): Results Obtained my experiments

19 better compared to all other Models having the best sensitivity of 97.7% and pneumonia with 92.6% though this model did not perform as expected as previous concatenated-old model but I have trained with very less parameters compared to the parameters generated by the authors method.

5 Modal Parameters and Key Words

Learning Rate:

It determines the step size at which the algorithm updates the weights of the network during training. It is a crucial parameter that influences the performance of the network, and it determines how quickly the algorithm converges to a solution.

Epochs:

It determines how many times the algorithm will iterate over the training dataset during training. Increasing the number of epochs increases improves the performance of the network. But it also increases the training time, algorithm may overfit the data and fail to generalize to new examples.

Batch Size:

It is the number of training samples that are processed by the network before the weights are updated. Increasing the batch size can improve the performance of the network, However, it also increases the memory requirements of the network ant the algorithm may not be able to fit the data in memory, leading to out-of-memory errors.

Nadam Optimizer:

Nadam uses an adaptive learning rate that is adjusted based on the past gradient values. This allows the algorithm to automatically tune the learning rate for each parameter, which helps to improve the performance of the network.

Categorical cross-entropy:

It is a a loss function that is used in classification tasks where the classes are exclusive. The cross-entropy loss is calculated as the negative log likelihood of the true class, given the predicted probabilities for each class.

6 Challenges

The limited availability of covid 19 images is the biggest challenge for me as we have very less covid positive image compared to normal and pneumonia positive it is hard to train a model to get unbiased results is a hard to achieve. So I have split the training data in 8 batches and trained the model varying pneumonia and Normal images but if we would have more covid -19 images it would be a lot easier and the model would perform better as the neural networks performs better when we have large amount of data.

As we are using the public dataset the data used in the paper varies with the current data it so we use only the accuracies percentage the metric to evaluate rather than the number of images detected.

Training a Neural Network requires a lot of computation power I have tried using my desktop at first but It could not be able to handle that so I have used google Collaboratory as we are training a large data the free GPU provided by them is stopping while training so I had to train from the scratch again but collab pro dis the work for me but each model took more than half a day to train and test any change in the parameter leads to retraining of the model which took me a lot of time for me to achieve these results.

7 Conclusions

In this paper we have we have explored the ways to identify whether a patient is infected with Covid-19, Pneumonia or Normal using deep neural networks. The data is fetched from two different publicly available datasets but as we have limited Covid-19 images, so I have split the training data into 8 different phases and trained the model. I have implemented CNN, InceptionV3, Resnet50V2, Xception, Concatenated Models. The proposed Concatenated model has the best sensitivity for Covid-19 with 97.77% and Pneumonia with 92.6%. and has the best prediction towards both Covid-19 and Pneumonia images compared to the rest of the Models. These accuracies can be improved if we have more Covid-19 images as this diversifies the model and features extratcted are learnt accurately.

8 References

[1] Mohammad Rahimzadeh , Abolfazl Attar(2020) A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2

[2] Sethy PK, Behera SK (2020) Detection of coronavirus disease (covid-19) based on deep features

[3] Apostolopoulos ID, Mpesiana TA (2020) Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks.
[4] Narin A, Kaya C, Pamuk Z (2020) Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks
[5] Christian Szegedy Google Inc, Vincent Vanhoucke , Sergey Ioffe , Jonathon Shlens , Zbigniew Wojna (2015) Rethinking the Inception Architecture for Computer Vision
[6] Kaiming He , Xiangyu Zhang , Shaoqing Ren , Jian Sun(2015) Deep Residua Learning for Image Recognition
[7] Francois Chollet(2016) Xception: Deep Learning with Depthwise Separable Convolutions
[8] Covid ChestxRay dataset https://github.com/mr7495/covid19
[9] RSNA Dataset RSNA Pneumonia Images: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge