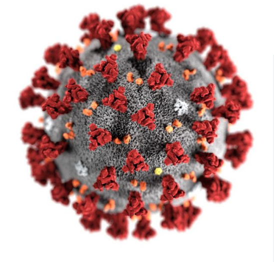


Machine Learning based Covid-19 detection using X-Ray Images

MENTOR : DR VRIJENDRA SINGH

Group Members-
Hitika Rajesh Kumar (IIB2019021)
Pranav Singhal (IIT2019050)
Rajpal Singh Shekhawat (IIT2019061)
Slok Aks (IIT2019067)
Sumit Kumar Sahu (IIT2019069)

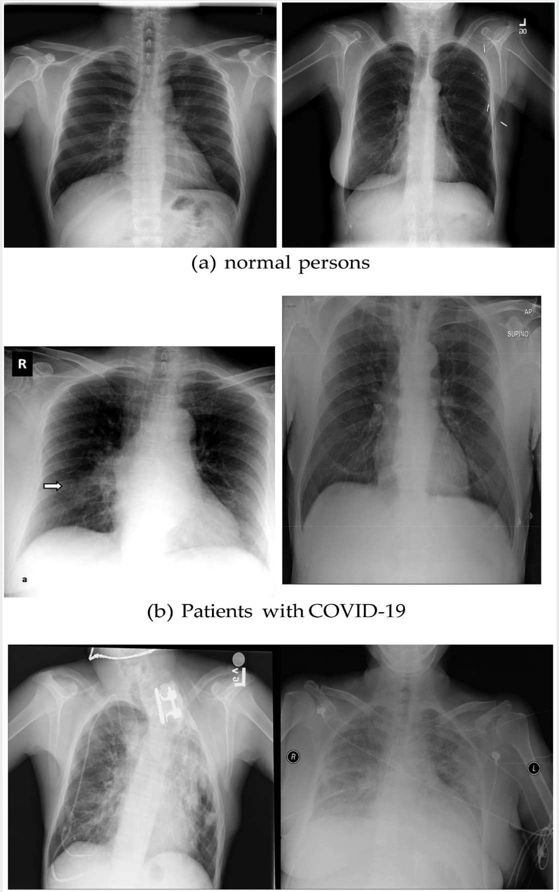
Problem Statement and Definition

Problem Statement

Given a Chest X-Ray scan as input, classify it to one of the following three classes:

1. Non-Pneumonia
2. Pneumonia
3. Covid-19 Positive

Different X-Ray labeled to specific diseases



(a) normal persons

(b) Patients with COVID-19

(c) Patients with pneumonia

Introduction to our approach

We have trained several deep convolutional networks with introduced training techniques for classifying X-ray images into three classes: normal, pneumonia, and COVID-19, based on two open-source datasets.

Our data contains 180 X-ray images that belong to persons infected with COVID-19, and we attempted to apply methods to achieve the best possible results. In this research, we introduce some training techniques that help the network learn better when we have an unbalanced dataset (fewer cases of COVID-19 along with more cases from other classes).

We also propose a neural network that is a concatenation of the Xception and ResNet50V2 networks.

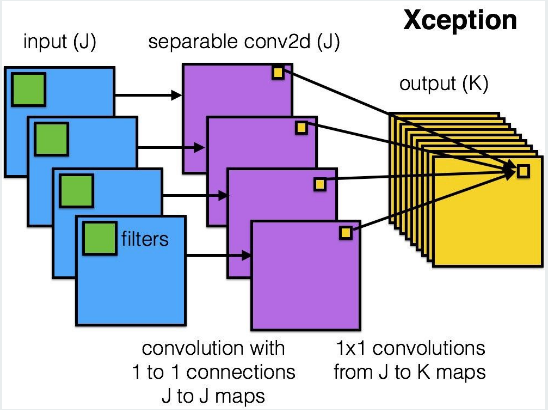
Methodology

Deep convolutional neural networks are useful in machine vision tasks. These have created advances in many field like Agriculture ; medical disease diagnosis ; and industry . The superiority of these networks comes from the robust and valuable semantic features they generate from input data.

Here the main focus of deep networks is detecting infection in X-ray images, so classifying the X-ray images into normal, pneumonia or COVID-19. Some of the powerful and most used deep convolutional networks are VGG ; ResNet ; DenseNet ; Inception ; Xception.

Xception Neural Network

Xception is a deep convolutional neural network that introduced new inception layers. These inception layers are constructed from depthwise convolution layers, followed by a point-wise convolution layer.



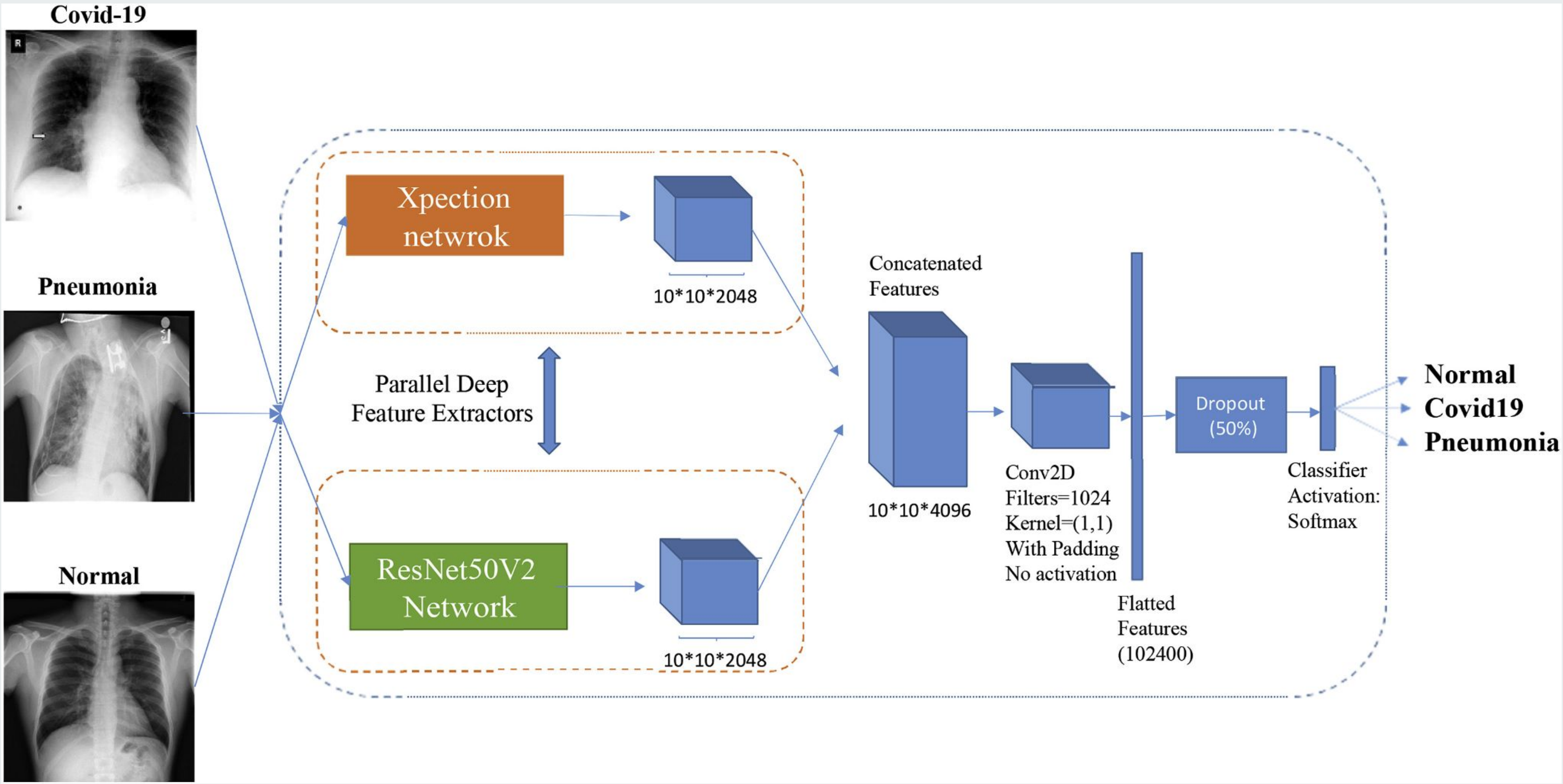
ResNet50V2

ResNet50V2 is a modified version of ResNet50 that performs better than ResNet50 and ResNet101 on the ImageNet dataset. In ResNet50V2, a modification was made in the propagation formulation of the connections between blocks. ResNet50V2 also achieves a good result on the ImageNet dataset.

Why we are using these models?

We are using Xception and ResNet50V2 models that are trained on a very large imagenet database. We have omitted the top layer of this model.
This model is used to extract features like shape, size, orientation etc in these layers.
Than, we would use these extracted features in our own fine tuning custom dense layers.
In total, we have used transfer learning.
At the end we have used the softmax layer for classification into the required three categories.

Block Diagram



Dataset Used

We have used two open-source datasets in our work. The covid chestxray dataset is taken from this GitHub repository (https://github.com/ieee8023/covid-chestxray-dataset), which has been prepared by Refs. [37]. This dataset consists of X-ray and CT scan images of patients infected to COVID-19, SARS, Streptococcus, ARDS, Pneumocystis, and other types of pneumonia from different patients. In this dataset, we only considered the X-ray images, and in total, there were 180 images from 118 cases with COVID-19 and 42 images from 25 cases with Pneumocystis, Streptococcus, and SARS that were considered as pneumonia. The second dataset was taken from (https://www.kaggle.com/c/rsna-pneumonia-detection-challenge), which contains 6012 cases with pneumonia and 8851 normal cases.

Dataset	COVID-19	Pneumonia	Normal
covid chestxray dataset	180	42	0
rsna pneumonia detection challenge	0	6012	8851
Total	180	6054	8851
Training Set	149	1634	2000
Validation Set	31	4420	6851

Training Phase

We have allocated 8 phases for training. For reporting more reliable results, we chose five folds for training, where in every fold the training set was made of 8 phases as it is mentioned. We have trained ResNet50V2 ; Xception; and a concatenation of Xception and ResNet50V2 neural networks based on the explained method. This concatenated Neural Network has shown higher accuracy compared to others. As we have tested several networks in our project, the Xception and ResNet50V2 networks work as well or better than others in extracting deep features. By concatenating the output features of both networks, we helped the network learn to classify the input image from both feature vectors, which resulted in better accuracy.

Training Parameters	Xception	ResNet50V2	Concatenated Network
Learning Rate	1e-4	1e-4	1e-4
Batch Size	30	30	20
Optimizer	Nadam	Nadam	Nadam
Loss Function	Categorical Crossentopy	Categorical Crossentopy	Categorical Crossentopy
Epochs per each Training Phase	100	100	100

Xception and ResNet50V2 layer use

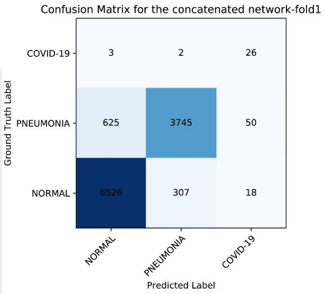
```
import keras.backend as k
k.clear_session() #Clear keras backend
try:
  os.mkdir('models')
except:
  pass
full_name='concatenate'
classes_number=3 #Number of classes
input_tensor=Input(shape=(300,300,3))
######################################################################################################
base_model1 = Xception(weights='imagenet', include_top=False, input_tensor=input_tensor)
features1 = base_model1.output
######################################################################################################
base_model2 = ResNet50V2(weights='imagenet', include_top=False, input_tensor=input_tensor)
features2 = base_model2.output
concatenated=keras.layers.concatenate([features1,features2]) #Concatenate the extracted features
####################################################################################################
```

Fine Tuning layers

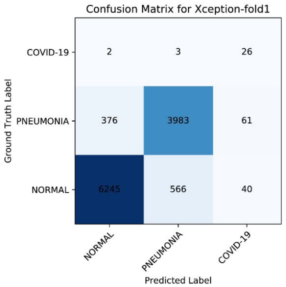
```
conv=keras.layers.Conv2D(1024, (1, 1),padding='same')(concatenated) #add the concatenated features to a convolutional layer
feature = Flatten(name='flatten')(conv)
dp = Dropout(0.5)(feature) #add dropout
preds = Dense(classes_number, activation='softmax', kernel_initializer=RandomNormal(mean=0.0, stddev=0.001))(dp)
Concatenated_model = Model(inputs=input_tensor, outputs=preds)
########################################################
```

Results

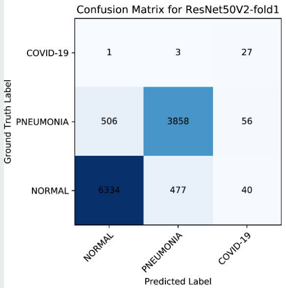
The evaluation results of the neural networks are presented in figure which shows the confusion matrices of each network for fold one and three.



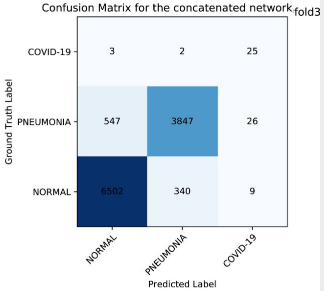
(a) Concatenated network-Fold1



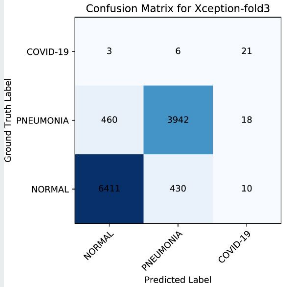
(b) Xception-Fold1



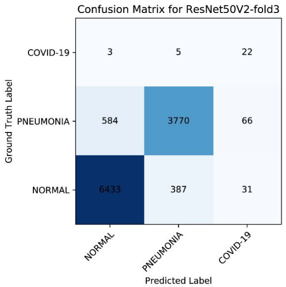
(c) ResNet50V2-Fold1



(d) Concatenated network-Fold3



(e) Xception-Fold3



(f) ResNet50V2-Fold3

Table representing the various result parameters for fold-1

Network	Accuracy	COVID-19 Sensitivity	PNEUMONIA Sensitivity	NORMAL Sensitivity	COVID-19 Specificity	PNEUMONIA Specificity	NORMAL Specificity	COVID-19 Accuracy	PNEUMONIA Accuracy	NORMAL Accuracy	COVID-19 Precision	PNEUMONIA Precision	NORMAL Precision
Xception	90.72	83.87	90.11	91.15	99.1	91.73	91.51	99.06	91.10	91.29	20.47	87.50	94.29
ResNet50V2	90.41	87.09	87.28	92.45	99.15	93.03	88.61	99.12	90.78	90.94	21.95	88.93	92.58
Concatenated	91.10	83.87	84.72	95.25	99.4	95.51	85.89	99.35	91.29	91.57	27.65	92.37	91.22

SLOK HAS MADE ONLY THIS SLIDE

THANK YOU!!!