<u>Classification of COVID Cases using Chest X-Rays</u>

By
Mortha Sai Sriram
B. Tech- Computer Science & Technology
6th Semester
IIEST Shibpur

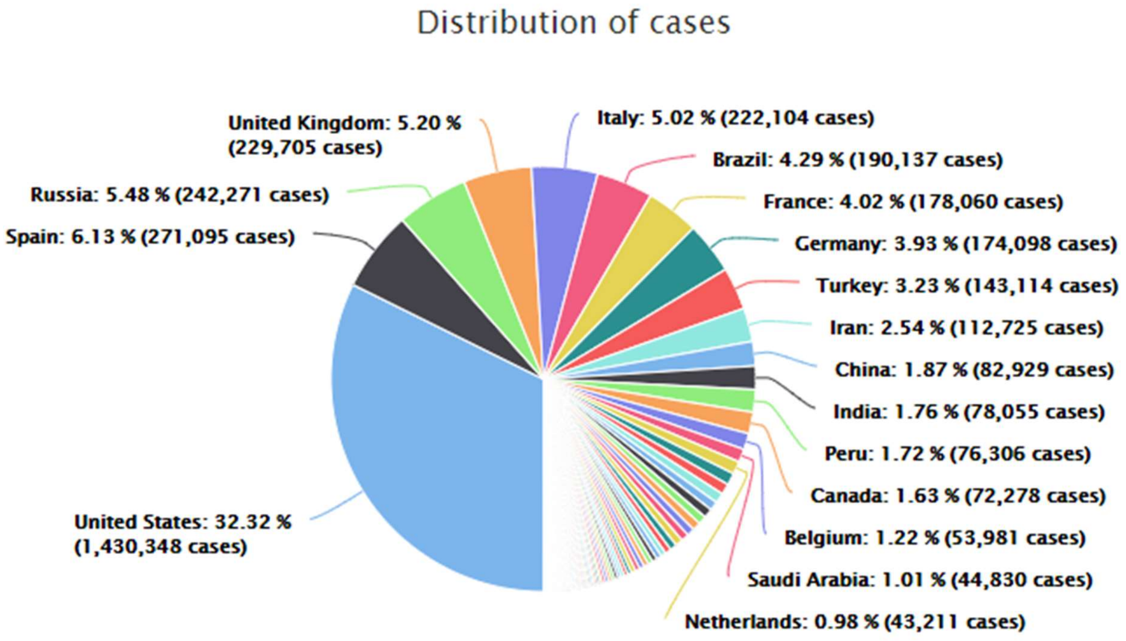
Abstract

The 2019 novel coronavirus (COVID-19), with a starting point in China, has spread rapidly among people living in other countries, and is approaching approximately 4 Million cases worldwide according to Wikipedia. There are a limited number of COVID-19 test kits available in hospitals due to the increasing cases daily. Therefore, it is necessary to implement an automatic detection system as a quick alternative diagnosis option to prevent COVID-19 spreading among people. In this study, a convolutional neural network-based model (GoogleNet) has been trained for the detection of coronavirus infected patient using chest X-ray radiographs. ROC analyses and confusion matrices by the model is given. Considering the performance results obtained, it is seen that the pre-trained GoogleNet model provides the classification performance with 96.6% accuracy.

Introduction

The 2019 novel coronavirus (COVID-19) pandemic appeared in Wuhan, China in December 2019 and has become a serious public health problem worldwide [1, 2]. The virus that caused COVID-19 pandemic disease was called as severe acute respiratory syndrome coronavirus 2, also named SARS-CoV-2 [3]. Coronaviruses (CoV) are a large family of viruses that cause diseases resulting from colds such as the Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV). Coronavirus disease (COVID-19) is a new species that was discovered in 2019 and has not been previously identified in humans. Coronaviruses are zoonotic due to contamination from animals to humans. Respiratory transmission of the disease from person to person caused rapid spread of the epidemic.

Worldwide, around 4.3 million people are affected COVID-19. Among these cases around 1.5 million have recovered and around 296K people died. Even in many developed countries, the health system has come to the point of collapse due to the increasing demand for intensive care units simultaneously. Intensive care units are filled with patients who get worse with COVID-19.



What is our Goal?

To be able to classify Chest X-ray Images of patients for COVID-Positive and Negative as correctly as possible. Because we are dealing with a pandemic here.

What we have?

We have Chest X-ray Images of patients which are positive or suspected of COVID-19 or other viral and bacterial pneumonias.

The dataset link is given here.

How are we going to do it?

We are going to train a Deep Convolutional Neural Network (GoogleNet) to classify our images into COVID Positive and COVID Negative.

What is Deep Learning?

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

What is an Artificial Neural Network?

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

In ANN implementations, the "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold.

Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.

When we have many layers, the neural network is a Deep Neural Network.

At the end, even a Deep Neural Network depends on many artificial neurons.

What does an Artificial Neuron Do?

An Artificial Neuron consists of some inputs (real numbers) and the output of each neuron is computed by some non-linear function of the sum of its inputs.

It is basically Regression.

What is Regression?

Regression takes a group of random variables, thought to be predicting Y, and tries to find a mathematical relationship between them.

Regression is of two types.

When the variable we are trying to predict is a continuous variable, the Regression is called **Linear Regression**.

When the variable we are trying to predict is a discrete variable, the Regression is called **Logistic Regression**.

What kind of Regression we need to use?

We are classifying an image as COVID Positive (1) and COVID Negative (0).

So, the regression we need to use is Logistic Regression.

In its simplest form the deep neural network is performing a Logistic Regression.

Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression).

An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is a sigmoid function, which takes any real input $t, (t \in \mathbb{R})$, and outputs a value between zero and one for the logit, this is interpreted as taking input log-odds and having output probability.

The standard logistic function $\sigma: \mathbb{R} \rightarrow (0,1)$ is defined as follows:

$$\sigma(t) = \frac{e^t}{1+e^t} = \frac{1}{1+e^{-t}}$$

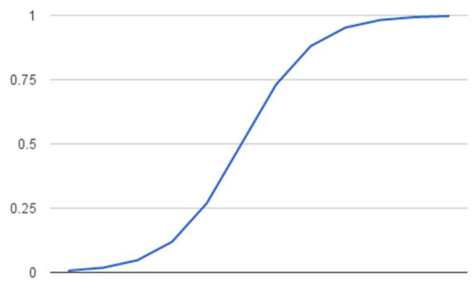


Fig: Logistic Function

Logistic regression uses an equation as the representation, very much like linear regression.

If there are **n** features of an input value **X**, then to predict an output value **y**, input values are combined linearly using weights $\boldsymbol{\theta}$. A key difference from linear regression is that the output value being modelled is a binary value (0 or 1) rather than a numeric value.

To make this feasible we apply the logistic function (sigmoid function) which outputs a value between 0 and 1.

$$X = (1, X_1, X_2, \dots\ X_n)$$

$$\theta = (\theta_0, \theta_1, \theta_2, \dots\ \theta_n)$$

$$y = \sigma(\theta * X^T)$$

The value of θ is trained using Gradient Descent algorithm.

This value can be considered as the probability that the given value belongs to positive class.

For a binary classification, we can set a threshold value by which we can classify each input.

If the output value is less than the threshold value, we classify that input as negative class and If the output value is greater than the threshold value, we classify that input as positive class.

Performance of a Classification

The performance of a classification is estimated on the value of accuracy of the Classification.

$$\text{Accuracy} = {}^{\text{Correct Prediction}}/_{\text{Size of Dataset}}$$

What if accuracy fails in predicting the validity of Classification?

Consider the following case We are trying to classify emails as spam and not-spam

Generally, the count of spam emails is very less compared to that of normal emails. If we have a classifier which classifies all emails as normal emails, the accuracy of the classifier is high even though our classifier isn’t working at all.

When the data is sensitive towards one class, accuracy usually can’t decide a better classifier.

To overcome this, we form a **Confusion Matrix** for the outputs of the classification and derive some formulas that gives better measure of performance of our classifier.

Confusion Matrix

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of a classifier.

Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).

		Actual class	
		Positive	**Negative**
Predicted class	**Positive**	**True Positive**	False Positive
	Negative	False Negative	**True Negative**

In accuracy we consider only True Positive and True Negative cases for measuring the performance of a classifier. That doesn’t work in all the cases.

So, we define new terms from the confusion matrix.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

$$\text{True Negative Rate} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

Recall is also called Sensitivity and True negative rate is also called Specificity.

Types of Errors in Classification

- **False Positive Error** also known as **Type 1 Error,** occurs when a classifier outputs the prediction as Positive but in reality, it is Negative.
- **False Negative Error** also known as **Type II Error,** occurs when a classifier outputs the prediction as Negative but in reality, it is Positive.

ROC Curve

For the classification tasks a new characteristic (ROC Curve) has been introduced.

A receiver operating characteristic curve, or ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied.

The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as probability of false alarm and can be calculated as (1 − specificity).

It can also be thought of as a plot of the power as a function of the Type I Error of the decision rule (when the performance is calculated from just a sample of the population, it can be thought of as estimators of these quantities).

When using normalized units, the area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative'). This area is called ROC AUC Score.

The ROC AUC Score lies between 0.5 to 1 where 0.5 denotes a bad classifier and 1 denotes an excellent classifier.

Which Error is more dangerous in Medical Domain?

In medical domain, using classification we usually predict whether a person is affected with a particular disease based on some inputs.

False Negative Error is more dangerous in that case compared to False Positive Error.

Consider that the disease we are trying to predict is contagious.

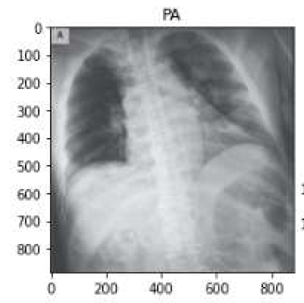
- False Negative Error (type II error) is the case in which the person actually has the disease and our classifier is predicting that he is not infected with the disease. So, he will not take precautions of staying away from people. This makes him a super spreader.
- False Positive Error (type I error) is the case in which the person actually doesn’t have the disease and our classifier is predicting that he is infected with the disease. This makes the person to take extra care by isolating himself from people and take necessary treatment from the doctors. In further medical tests, he will come to know that he doesn’t have the disease. This error is not dangerous.

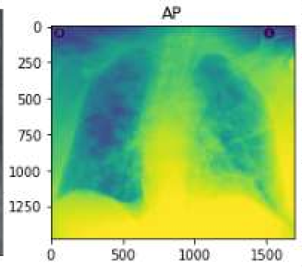
What we have done

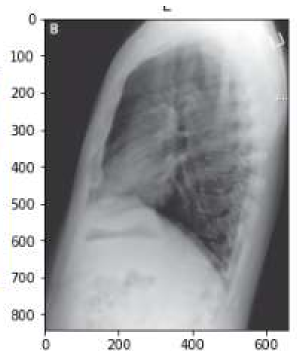
- Pre-processing Images
- Defining the Model
- Training the Model
- Predicting the outputs

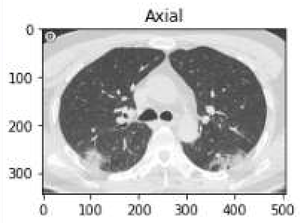
Dataset Description

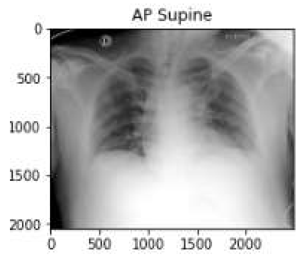
- This dataset is a public open dataset of Chest X-Ray images of patients which are positive or suspected of COVID-19 or other viral and bacterial pneumonias.
- The victims of each disease in the data are displayed below.
- But, For the purpose of our project, we have considered only COVID and Non-COVID results.
- There are different views of Chest X-Rays.

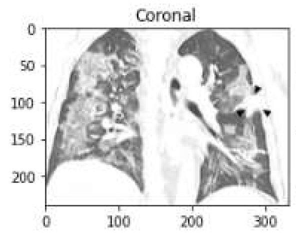


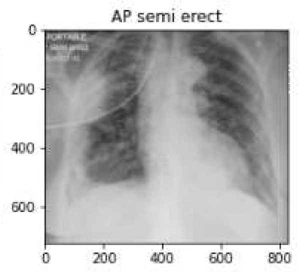


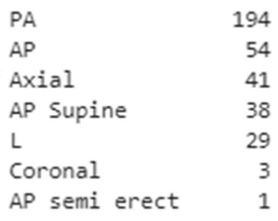












```
PA              194
AP               54
Axial            41
AP Supine        38
L                29
Coronal           3
AP semi erect     1
```

- The counts of each type of X-Rays in the dataset are displayed above.
- As there are a greater number of images of PA view, it would be suitable to select PA view for training.

PRE-PROCESSING

- I have planned to use a transfer learning model which requires a 3-channel image with dimensions 224* 224*3
- But in the dataset, there are few grayscale images (single channelled images).
- This will cause a dimensional mis-match.
- To prevent this, I have converted each image into a 3-channel image.
- Then I have resized the image into 224*224*3.
- For the transfer learning model, images should be normalized with specific values.
    - Mean values for each channel are given in the list. [0.485, 0.456, 0.406]
    - Standard Deviation Values for each channel are given in the list [0.229, 0.224, 0.225]

Balancing the Dataset

- In the dataset given the majority of the images are of COVID. This will cause the model to overfit the images by classifying all the images into COVID images.
- To overcome this, I have defined the training set to contain equal number of COVID and Non-COVID images.
- I have kept all the remaining images in the test set.
- This enables the model to learn both kinds of cases.

Transfer Learning

- Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.
- Basically, in transfer learning, we will use a model (deep convolutional network) trained on certain images (they may be of completely different domains).
- We will update the classifier of the model to suit our problem.
- Even if the images are of different domain, the model works well.
- For our problem, we have selected GoogleNet model.

GOOGLENET

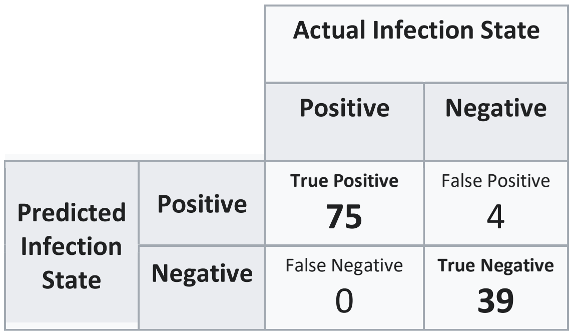
- It is a 22 layered deep convolutional network.
- It was first trained on ImageNet Dataset.
- The ImageNet dataset is a large visual database designed for use in visual object recognition software research.
- ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry", consisting of several hundred images.
- This model was designed to classify the images into 1000 distinct categories.
- For every deep convolutional network, a classifier should be defined at the end of the network to fit the desired number of categories.
- In the original model, the classifier was defined to classify into 1000 categories.
- The convolutional layer in the networks find different features in the images.
- So, a pretrained deep convolutional network also finds the features of the image even if they are of different domains.
- In our case, we need to classify each image into two different categories. So, I have modified the classifier to categorize into two classes (COVID Positive and negative)
- This model needs each image to have dimensions 224*224*3
- And each image should be normalized with the above-mentioned values.

Training the Model

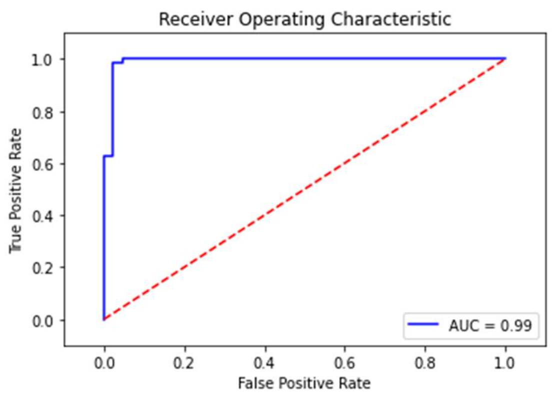
- I have trained the model using Binary Cross Entropy Loss as criterion and Adam Optimiser
- I have already trained the model and saved the best model we have achieved.

Predicting the Output

- I have used only 50% of the data for training purposes. Using remaining data, we have predicted the output of the model.
- This is the confusion matrix and ROC curve we have obtained



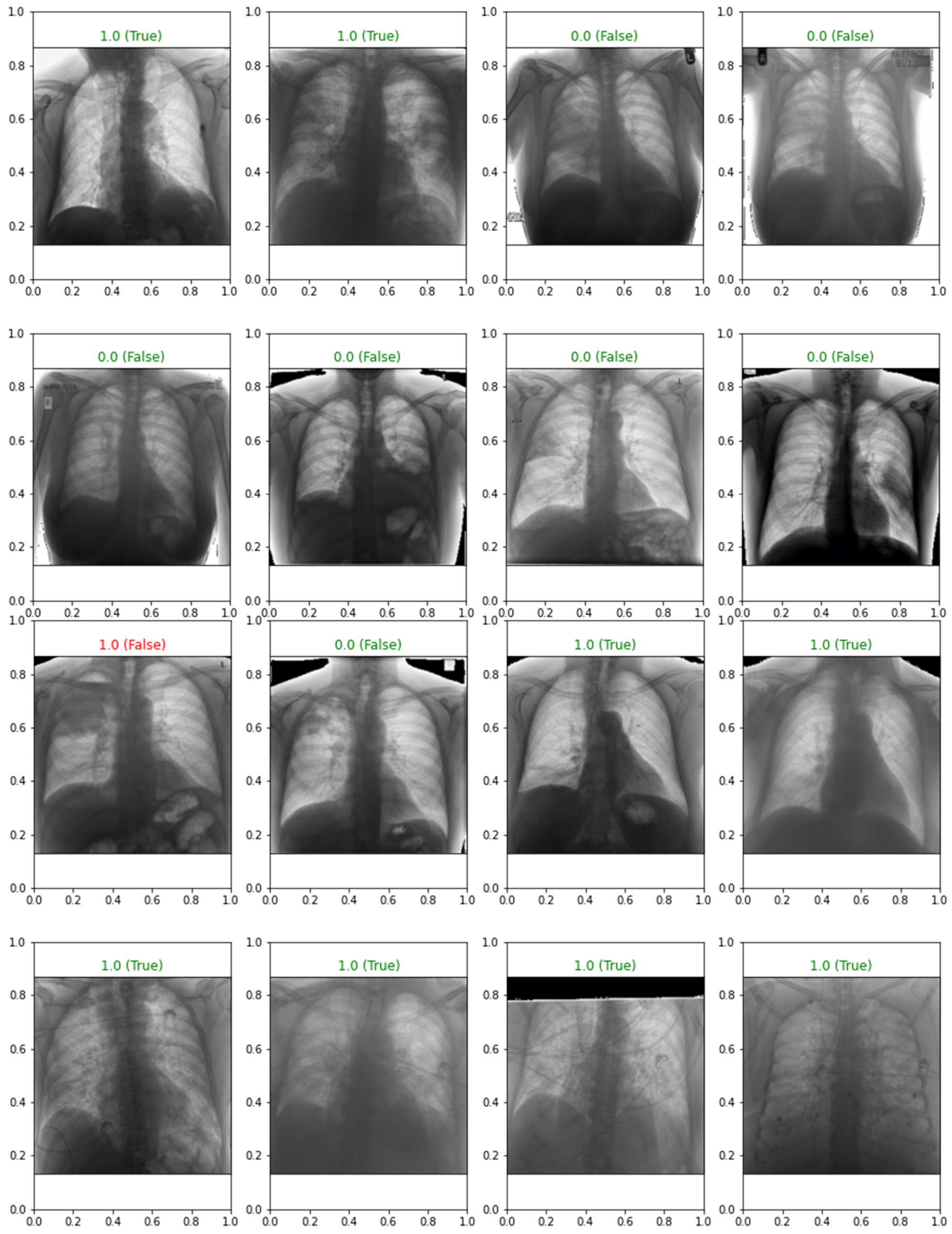
		Actual Infection State	
		Positive	Negative
Predicted Infection State	Positive	True Positive 75	False Positive 4
	Negative	False Negative 0	True Negative 39



- In our prediction, there are no False Negative cases.
- So, we can say that our model is working fine for the dataset.

The Images displayed below are the output of the model on the testing data.
True Label of the image is displayed in Brackets and Predicted label is shown outside the brackets.

True ---- 1.0 ---- COVID Positive
False --- 0.0 ---- COVID Negative



Using this model to predict COVID-19 Patients

The training of the model is saved in **final_covid_training.ipynb** notebook and saved the model with best ROC AUC Score and made it a checkpoint.

To make a new prediction, first, we have to define the same model and load the checkpoint. We have to pre-process the image and make a prediction from the model.

This entire function is written in **final_covid_prediction.ipynb**. For the function we just need to provide image path and model to the function.
Along with these files a checkpoint file **best_model.pt** is attached.

Requirements to run the files

The entire requirements are saved in requirements.txt.
To install all the requirements, we run the following command.

```
pip install -r requirements.txt
```

I suppose **Anaconda** is already installed in the machine. Then run the following command

```
conda install pytorch torchvision -c pytorch
```

Conclusion

Early prediction of COVID-19 patients is vital to prevent the spread of the disease to other people. In this study, we proposed a deep transfer learning-based approach using chest X-ray images obtained from COVID-19 patients and normal to predict COVID-19 patients automatically. Performance results show that the GoogleNet pre-trained model yielded the accuracy of 96.6%. In the light of our findings, it is believed that it will help doctors to make decisions in clinical practice due to the high performance. In order to detect COVID-19 at an early stage, this study gives insight on how deep transfer learning methods can be used. In subsequent studies, the classification performance of different CNN models can be tested by increasing the number of images in the dataset.

References

1. Open dataset of chest X-ray and CT images of patients which are positive or suspected of COVID-19 or other viral and bacterial pneumonias
   https://github.com/ieee8023/covid-chestxray-dataset

2. Data regarding the world wide spread of COVID-19
   https://www.worldometers.info/coronavirus/coronavirus-cases/

3. Information regarding GoogleNet Model
   https://arxiv.org/abs/1409.4842

4. Dermatologist-level classification of skin cancer with deep neural networks
   https://www.nature.com/articles/nature21056.epdf?author_access_token=8oxIcYWf5UNrNpHsUHd2StRgN0jAjWel9jnR3ZoTv0NXpMHRAJy8Qn10ys2O4tuPakXos4UhQAFZ750CsBNMMsISFHIKinKDMKjShCpHIlYPYUHhNzkn6pSnOCt0Ftf6