Comparative Study of X-ray and CT Scan Images for the Detection of COVID-19 using Deep Learning

by

Ahashan Habib Niloy
17301004
Shammi Akhter Shiba
18201124
S.M. Farah Al Fahim
17201151
Faizun Nahar Faria
17201003
Md. Jamilur Rahman
17101291

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
August 2015

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Ahashan Habib Niloy

Ahashan Habib Niloy
17301004

[illegible]

Shammi Akhter Shiba
18201124

Fahim

S.M. Farah Al Fahim
17201151

Faizun

Faizun Nahar Faria
17201003

[illegible]

Md. Jamilur Rahman
17101291

Approval

The thesis titled “Comparative Study of X-ray and CT Scan Images for the Detection of COVID-19 using Deep Learning” submitted by

1. Ahashan Habib Niloy (17301004)
2. Shammi Akhter Shiba (18201124)
3. S.M. Farah Al Fahim (17201151)
4. Faizun Nahar Faria (17201003)
5. Md. Jamilur Rahman (17101291)

of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on June 6, 2021.

Examining Committee:

Supervisor:
(Member)

Zavid Parvez
Digitally signed by Zavid Parvez
DN: cn=Zavid Parvez, o=Brac University, ou=CSE, email=zavid.parvez@bracu.ac.bd, c=US
Date: 2021.06.10 14:53:22 +10'00'

Mohammad Zavid Parvez, PhD
Assistant Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

[illegible]

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Coronavirus 2019 (in short, COVID-19), originated in the Wuhan province of China in December 2019, has been declared a global pandemic by WHO in March 2020. Since its inception, it's rapid spread among nations had initially collapsed the world economy and the increasing death-pool created a strong fear among people as the virus spread through human contact. Initially doctors struggled to diagnose the increasing number of patients as there was less availability of testing kits and failed to treat people efficiently which ultimately led to the collapse of the health sector of several countries. To help doctors primarily diagnose the virus, researchers around the world have come up with some radiology imaging techniques using the Convolutional Neural Network (CNN). While some of them worked on x-ray images and some others on CT scan images, none worked on both the image types. Thus there's no way to know which image works better for a particular model. This, therefore, insisted us to perform a comparison between x-ray and CT scan images. Thus we came up with a novel CNN model named CoroPy which works for both the image types and shows that in 2 classes (normal and covid), CT scan images show a better accuracy and it is 99.17% whereas it is 95.73% for x-ray images. However, in the case of 3 classes (normal, covid and viral pneumonia), x-ray images show a better accuracy and it is 92.45% whereas it is 68.81% for CT scan images.

Keywords: COVID-19; Machine learning; Deep learning; Convolutional Neural Network; X-ray; CT scan; Pneumonia; Confusion matrix

Acknowledgement

First of all, all praises to Almighty Allah who gave us a chance to work on this thesis and gave us enough strength to complete it successfully. We also want to express our gratitude to our supervisor Mohammad Zavid Parvez for his effective guidance and support. Without his proper guidance, it would not have been possible for us to complete this thesis. In every step of our research, he encouraged us to achieve our desired goal. Furthermore, we are grateful to Emtiaz Hussain who is an experienced person in our thesis field and helped us in every steps in our thesis. We are also thankful to our family, friends and loved ones who supported us in our research. Last but not least, we appreciate BRAC University for giving us the opportunity to perform this research with all the necessary resources required for this research.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN	Convolutional Neural Network
FN	False Negative
FP	False Positive
ML	Machine Learning
$ReLU$	Rectified Linear Unit
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group

Chapter 1

Introduction

1.1 Motivation

Coronavirus 2019 (in short, covid-19) has originated from the Wuhan city of China back in December 2019 [1] and has been declared as a global pandemic by WHO in march 2020 [2]. Since its inception, it has completely changed the way our life used to be. As this virus spread through human contact, the world faced several lockdowns which created drastic effects on the world economy [3]. Within a short span of only one and half years, it has affected 170 Million people and taken the life of 3.54 Million people [4] (as of 29th May 2021).

From the beginning of this pandemic, testing has been considered as one of the major issues. Several problems related to detection of Covid-19 such as less availability of testing kits, less testing centers, more detection time, less number of tests compared to the total population primarily worsened the situation. With the passage of time, though the availability of testing kits has increased and the test duration got reduced, the availability of test centers and the number of test cases is still not enough. This low number of testing is, in fact, hiding the actual number of patients and thus facilitating the rapid spread of COVID-19 [5].

Though inadequate testing is a global issue, it is even horrible in developing and least-developed countries [6] where the health sector has almost collapsed. In a developing country like Bangladesh, the number of test cases is very low with a total of only 5.88 Million test cases(as of 27th May 2021) [7] whereas the total population is 166.3 Million [8]. Both less testing and testing delays are hiding the actual number and worsening the situation gradually in the country. Sometimes it takes a week to get the test results and sometimes it doesn't even come [9] and people are dying before getting their delayed test results. In fact, this problem is not country specific. This is the problem of this complicated testing system. Developed countries like the UK are facing some miserable conditions due to delays in testing. They are sending people to various distant test centers as the nearest test centers cannot finish processing their testing [10] and delivering test results.

Therefore, this delay in detection of COVID-19 and this complicated testing method needs to be addressed. There should be a detection system which will meet all of these problems, be fast to detect with accuracy and be available anywhere regardless of the testing centers. That's why we are working on x-ray and CT scan images which will solve the mentioned problems and also help doctors initially diagnose patients quickly. As less availability of testing kits is one of the major problems, if we can detect covid-19 using x-ray or CT scan images, this problem will be solved as we will not need individual x-ray or CT scan machines for each test and in fact, the x-ray, CT scan machines are there across the globe. Another major issue was the low number of COVID-19 testing centers. This problem will also be addressed using x-ray or CT-scan because it is available in almost all medical centers across the globe. Another major issue regarding covid-19 testing was delays in getting test results. However, this issue will also be addressed using x-ray and CT scan machines and doctors can get x-ray, CT scan reports within minutes [11]. This faster and easier detection method will eventually result in a huge number of testing within the shortest period of time.

Furthermore, we have seen several studies where researchers tried to solve this issue by using radiology imaging techniques (x-ray, CT scan images) using CNN. However, while some research is on x-ray images and some others are on CT scan images, we didn't find any research which worked on both x-ray, CT scan images. Thus there remains a gap which has urged us to perform a comparative study between the performance of x-ray, CT scan images in detecting coronavirus using the same model. That's why we came up with a novel CNN model named CoroPy which works for both x-ray, CT scan images for 2 class (normal and covid) and 3 class (normal, covid and viral pneumonia). Thus we are trying to solve the problems like delays of testing, less availability of both testing centers and testing kits and help doctors primarily diagnose COVID-19 and treat patients efficiently at an early stage.

1.2 Aims and Objectives

Our main objective is to develop a novel CNN model which can work for both x-ray, CT scan images and help us perform a comparative analysis between the accuracy of x-ray and CT scan images. Our other major goals include:

- Developing our model to acquire great accuracy for both 2 class (normal and covid) and 3 class (normal, covid and viral pneumonia) for x-ray, CT scan images
- Applying basic CNN and other architectures of CNN like InceptionV3, VGG16, VGG19, MobileNetV2, Xception to detect covid-19
- Analyzing the accuracy of both x-ray, CT scan images using our model to find which type of image works well for both the 2 class and 3 class classification
- Analyzing the results of various models to find which models works best in diagnosing covid-19 for 2 class and 3 class classification and also performing k-fold to find maximum accuracy

1.3 Thesis Overview

Chapter 1 is the introduction of our thesis where it contains the motivation behind our thesis and our aims and objectives of the thesis.

Chapter 2 contains the literature reviews of the existing works and also contains related works and methodologies.

Chapter 3 contains the workflow of our thesis and the details of our dataset.

Chapter 4 contains the model analysis and implementation. This chapter discusses the details of the methodologies used in building the model and also how the model is implemented in our thesis that works for both x-ray, CT scan images.

Chapter 5 contains the result analysis section of our thesis and also the discussion section. This chapter analyzes the results obtained by implementing our model and analyzes the results between both x-ray, CT scan images and also discusses which classification works better for which type of image.

Chapter 6 is the conclusion of our thesis. This chapter also contains our future plans to take our thesis to the next level.

Chapter 2

Literature Review and Related Works

In this chapter, we have provided the background study which includes literature review and the detailed explanation of the technological terms we have used throughout the research.

2.1 Literature Review

Kumar et al. [12] reviewed the prevailing technologies for tackling covid-19 pandemic and they conducted a gradual search of the modern technology database in their research. They have analyzed different modern technologies both from the patient's perspective and from computational biology and medicines perspective. They have gathered the database of different works for tackline the covid-19 pandemic analyzing their dataset, implementation model and accuracy. However, they haven't proposed any model of their own or implemented anything new in their research work.

Pereira et al. [13] presented a categorization technique based solely on X-ray images to differentiate pneumonia caused by covid-19 in other species from healthy lungs. They have developed their own model and claimed their work to have the best accuracy among other published research works. However, X-ray images are not considered as gold standard like that of CT-scan images. Therefore, it would be really great work if they could tell something about CT-scan image related detection systems.

Wang et al. [14] suggested a customized deep convolutional neural network architecture for the detection of covid-19 instances in chest X-ray pictures and also claimed that their work was the first of its type. They classified the X-ray images into three categories: normal cases, pneumonia, and covid-19. However, their research works were only confined to X-ray images and lacked any information related to CT-scan images.

Narin et al. [15] offered three distinct models for coronavirus pneumonia diagnosis using CXR pictures. They have analyzed their models and found 98% accuracy for the ResNet50 model. However, they too were confined only to X-ray images creating room for working with CT-scan images.

Cohen et al. [16] suggested a model based on CXRs that predicted the seriousness of pneumonia, COVID-19. Both covid and non-covid datasets have been preprocessed with pictures (recompressing, centering, rescaling), then processed using the extraction caps and network task prediction layer.

Mahmud et al. [17] suggested a deep neural network architecture capable of detecting COVID-19 and other forms of pneumonia in the chest. By merging features from diverse receptive areas, efficient depth convergence is used to analyze abnormalities in x-rays from many perspectives. In addition, an additional performance stacking algorithm was used.

Ozturk et al. [18] proposed a model which can detect covid-19 from CXR pictures. They designed their own model called DarkCovidNet inspired by the DarkNet-19 model with fewer layers and filters. There are 17 convolution layers in the proposed model and performance is assessed using a 5-fold cross-validation process. With an accuracy of 98.08 and 87.02 percent respectively, their developed system can perform binary class- and multi-class tasks. However, they were just confined only to x-rays images. They did not use any CT-images for evaluating them.

Wenhui et al. [19] proposed the use of the X-ray data and Inception V3 model based on DCNN in order to detect COVID-19 automatically. They implemented their work with the Inception V3 model and TensorFlow. They increased the training set to 4000 and obtained a 96 percent accuracy on the final test. They did, however, use transfer learning techniques to compensate for the lack of data and training time. Moreover, they claim that their proposed model Inception V3 architecture performs better than most recent architecture.

Panwar et al. [20] conducted a study in which they employed their own built CNN-based model called COVent, which comprises 24 layers. These layers are trained on the ImageNet dataset. They analyzed their model using both the conventional VGG16 model, which achieved 92.7 percent accuracy on ImageNet, and a transfer learning model. Additionally, they validated the model against Kaggle's X-Ray scans of the chest. As a result, they achieve up to 97.62 percent training accuracy. Finally, they assert that their suggested approach is capable of detecting patients who test positive for COVID-19 in less than five seconds.

Khan et al. [21] offer a technique using deep CNN to detect covid-19 in CXR pictures in order to diagnose three distinct forms of pneumonia utilizing picture data from an open source GitHub repository. CoroNet is their suggested model, which is built on Xception CNN architecture with 71 layers. They employed fourfold cross-validation in conjunction with Keras' CoroNet implementation. CoroNet averaged 89.6 percent accuracy. Lastly, in their paper they were just talking about x-ray images.

Zheng et al. [22] created a software method for detecting COVID-related disorders that is based on weakly supervised deep learning and employs 3D CT volumes. Between December 13, 2019 and January 23, 2020, training data were collected, and testing data were collected between January 24, 2020 and February 6, 2020. The model used in this case, DeCoVNet, consisted of three stages. However, data augmentation was used in this case to avoid overfitting issues due to the limited number of training CT volumes. Testing a CT volume took only 1.93 seconds after training with 499 CT volumes.

Sethy et al. [24] developed a system based on deep CNNs for the identification of COVID-19 as a classification task. They used the dataset of X-Ray images which are available on repositories of GitHub, Kaggle and Open-i. They used multiple architectures for the detection of COVID-19. They used SVM classifiers with features which are obtained from the CNN models. However, ResNet50 and SVM give better classification results for detection of COVID-19. Moreover, they claim that ResNet50 along with SVM achieved 95.38% accuracy which is statistically much better compared to other models.

Hasan et al. [1] proposed the use of a CNN model called CoroNet. They used a 22-layer CNN model built with Keras and TensorFlow 2.0, as well as the ImageNet dataset for pre-training. They used different layered CNN models to determine which CNN model performs the best in terms of accuracy. However, they use their model to determine two-class, three-class, and four-class classification. Additionally, their model is capable of working with both CXR and CT images.

Tuncer et al. [25] propose a model that generates features using the Residual Exemplar Local Binary Pattern (ResExLBP). In the classification phase, different types of methods are used. The method for automatically detecting Covid-19 consists of three stages. At first it processes the data. Then, it extracts the features from the data. After that it selects from the extracted features. For training and testing, tenfold cross-validation was used.

2.2 Convolutional Neural Networks

CNNs consist of neurons which have the weight and bias. The inputs and weight of every neuron are transferred, the activation function passed on and the output reacted by each neuron. Initially CNN extracts the functionality and maps the extracted functions to the final output. The CNN consists of three layers of convolution, pooling, and fully connected layers [26]. They are convertible layers. The CNN layers are arranged in three dimensions: width, height and depth (for example, the third activation volume dimensions) [27].

2.2.1 Convolution Layer

The first layer after receiving input is the Convolution layer. This layer extracts features, typically consisting of a combination of linear and nonlinear operations (e.g. convergence and activation functions).Convolution process extracts functions in a narrow number range called a kernel, a tensor called number array throughout the input. Convolution is a linear process. At each location of the tensor a dot product is calculated between each kernel element and the input tensor, and the result is combined in the position of the outgoing tensor, called the feature map in Figure 2.1. Applying multiple kernels to an arbitrary set of feature maps representing the various properties of input tensors, multiple kernels can be considered as various feature extractors [26].

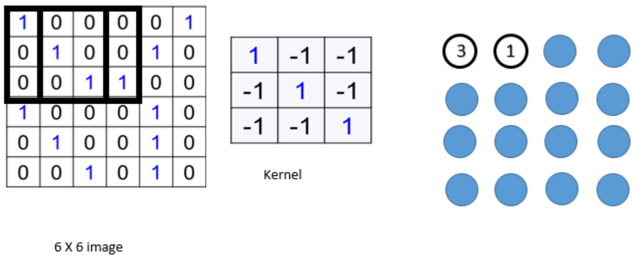


Figure 2.1: 6X6 image to 4X4 image using 3X3 kernel

2.2.2 Pooling Layer

The main purpose of this layer is to lower the size of the curved map so that the number of parameters and calculation in the network can be reduced. The pooling layer works on each input depth slice individually and spatially resizes it [27]. Note that in any of the pooling layers, no parameter can be read, whereas in pooling operations the filter size, stride and padding are hyperparameters similar to convolution operations [26] [27]. Pools are the most popular form of pooling, whereby patches from the input characters are extracted, the maximum value is given in every patch and all other values are discarded in Figure 2.2. In practice a max pooling with a filter size 2 x 2 is usually used. This shows the in-plan size of the maps by a factor of 2. The dimension of depth of the functional maps remains unchanged in contrast to height and width [26].

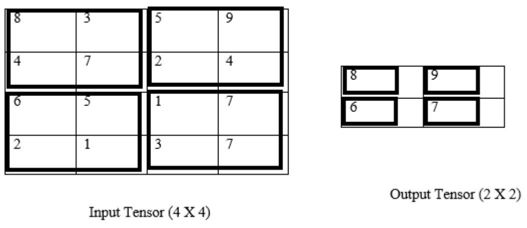


Figure 2.2: Input and Output Tensor

2.2.3 Fully Connected Layer

The fully connected (FC) layer consists of neurons' weights and biases and is used to connect neurons between two layers. These layers generally are placed in front of the output layer and are the final CNN architecture layers. Typically flattened, i.e., converted into a single-dimension (1D) array of values (or the Vector), the output feature map of the final turnover or pooling layer is connected to one or more fully connected strata, also known as dense levels, in which each input can be connected to each output with a learning weight. After functions are created using convolutional layers and sampled using pooling layers, these functions are mapped to the network's end outputs using a subset of completely connected layers, including the probability in image classification in each class. Usually there are the same number of output nodes as class numbers on the final fully connected layer. A nonlinear function, as described above, is followed by every fully connected layer [26]. Then begins the process of classification.

Chapter 3

Workflow and Dataset Description

Before building our own model, we have created a workflow which we have followed for our whole process starting from dataset generation to the performance evaluation. This step by step process has guided us properly to reach our ultimate goal. The steps included generating dataset, data labeling, designing our proposed model, training our proposed model and lastly performance evaluation. And this workflow or blueprint has been shown in Figure 3.1.

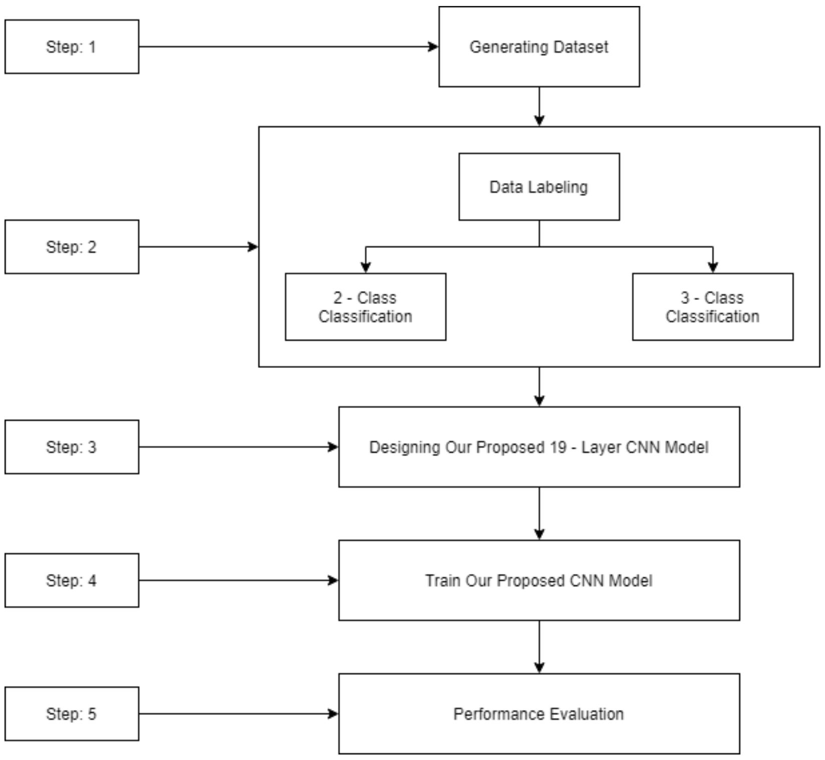


Figure 3.1: Workflow for our model

Then we have worked on each step to build our model properly with a higher accuracy rate. These steps have been discussed briefly.

3.1 Generating Dataset

Generating our dataset was the first step to build our model. And, we have generated one of the largest datasets having both chest X-ray and CT-scan images of Covid-19 affected persons. Our created dataset, CovRecker consists of 2541 X-ray images divided in 2 classes (NORMAL and COVID) and 3841 x-ray images divided in 3 classes (NORMAL, COVID, PNEUMONIA_VIRAL). And 360 CT-scan images divided in 2 classes (COVID and NORMAL) and 560 CT-scan images divided into 3 classes (COVID, NORMAL and PNEUMONIA_VIRAL). So, the total dataset comprises 4401 images. Out of which 45 (15 from each class) of X-ray images and 30 (10 from each class) of CT scan images have been reserved for testing the CoroPy model. A brief introduction to the dataset can be found in Table 3.1, Table 3.2, Table 3.3, Table 3.4 and Table 3.5. Figure 3.2 shows some of the sample dataset of X-ray images of Covid, normal and viral pneumonia. And Figure 3.3 shows it for CT scan images.

Class	Number of images
COVID	1185
NORMAL	1326
PNEUMONIA_VIRAL	1330

Table 3.1: Class wise image count

Labels	Number of Images
NORMAL	1341
COVID	1200
	Total = 2541

Table 3.2: Image count for X-ray images (2 Class)

Labels	Number of Images
NORMAL	200
COVID	160
	Total = 360

Table 3.3: Image count for CT scan images (2 Class)

However, our CovRecker is generated from six data sources, which are Covid-chestxray-datase [28], COVID-CT [29], SARS-COV-2 CT-scan dataset [30], COVID-19 x-ray dataset (train test sets) with COVID-19CNN pneumonia detector [31], COVID-19 Radiography database [32] and COVID-19 and common pneumonia chest CT dataset [33].

These three sample X-ray images from the CovRecker dataset are shown side by side in Figure 3.2. Placing the images side by side we can clearly see the differences among these. X-ray of normal or non-Covid patients show no greyish parts. The lungs image is clean and clear. On the other hand, X-ray image of covid-19 patient and pneumonia patient we can see signs of hyperinflation. We can see shades

Labels	Number of Images
NORMAL	1326
COVID	1185
PNEUMONIA_VIRAL	1330
	Total = 3841

Table 3.4: Image count for X-ray images (3 Class)

Labels	Number of Images
NORMAL	160
COVID	200
PNEUMONIA_VIRAL	200
	Total = 560

Table 3.5: Image count for CT scan images (3 Class)

of white or gray opacities for both cases of pneumonia and Covid-19 patient x-ray images which are caused by bacterial co-infection. Though covid-19 patients do not develop pneumonia, their infection areas and symptoms could be similar. If we compare the x-ray image of Covid-19 and pneumonia with the normal patient's x-ray image, we can see that the chest wall is inflated for the Covid-19 and pneumonia patients. The position of heart, trachea and mediastinum is also different. In the case of Covid-19 persons, the x-ray image shows that the opacity of the chest wall and all other vital organs are more obscure compared to pneumonia patients. Our task with this research is to differentiate these dissimilarities using CNN models.

These three sample CT scan images from the CovRecker dataset are shown side by side in Figure 3.3. From the images we can clearly identify the difference between covid patient from an non-covid patient. The affected or white areas are more pronounced and larger in covid patients compared to non-covid and normal pneumonia patients. Which is also seen for x-ray images. But these areas are easier to distinguish for x-ray images.

3.2 Data Labeling

Data Labeling was the second step for building our model. After generating our dataset, we have labeled our dataset, CovRecker. And it has been divided into 2 subsets for X-ray images and 2 subsets for CT scan images. 2 subsets for x-ray and CT scan images are classified into 2 labels and 3 labels . The images inside the subsets are then renamed accordingly. For 2 class we have labeled our data as COVID and Normal for both X-ray and CT scan images. And for 3 class, we have labeled our data as COVID, Normal and Pneumonia_Viral for both X-ray and CT scan images. This is shown in Table 3.6.

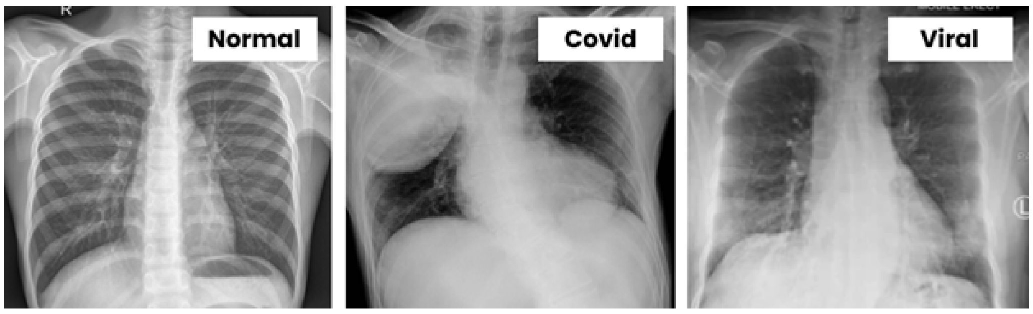


Figure 3.2: Sample dataset of X-ray images

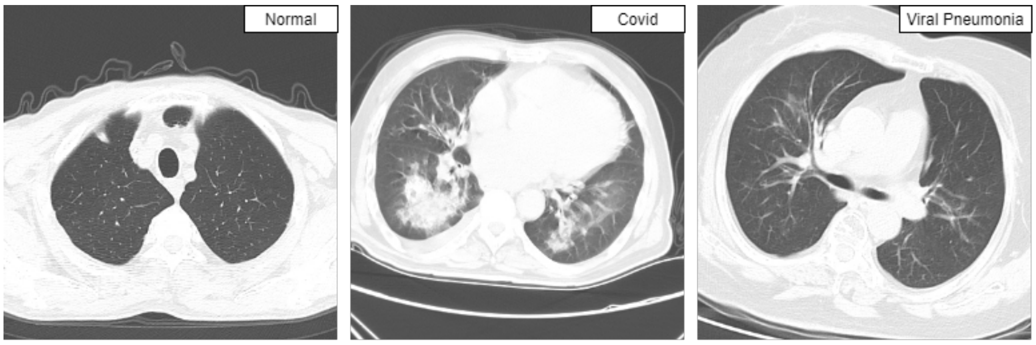


Figure 3.3: Sample dataset of CT scan images

3.3 Designing Our Proposed 19-layer CNN Model

Our proposed model consists of 19 layers consisting of five different types of layers which are - convolutional layer, pooling layer, dense layer, flatten layer, dropout layer. The numbers of layers we have used in each layer are shown in Table 3.7.

The most challenging for us was to configure each layer so that it gives the best results. Building the model for 2 class classification was quite easy but optimizing it for 3 class classification was quite difficult. We had to do a lot of testing and rearrangement of the layers to come up with the 19-layer CNN model.

The model at first contained only 3 conv2d layers and 2 maxpooling2d layers which performed with 92% validation accuracy for 2 class classification and 46.32% validation accuracy for 3 class classification. We kept on adding more conv2d layers and rearranging pooling layers on top of that to increase validation accuracy till we hit 92.45% for 3 class classification for X-ray images. We also had to configure the activation function for conv2d and dense layers to increase the accuracy. Finally, we got to this 19-layer CNN model which is both unique and gets the result required. We have discussed our model elaborately in Chapter 4.

Class	Labels
2 Class	NORMAL, COVID
3 Class	NORMAL, COVID, PNEUMONIA_VIRAL

Table 3.6: Class wise labels

Name of layers	Number of layers
Convolutional layer	10
Pooling layer	5
Dense Layer	2
Flatten Layer	1
Dropout Layer	1

Table 3.7: Number of layers

3.4 Model Training

We have used 3841 X-ray images and 560 CT-scan images to train our model. Then we have used Keras and Tensorflow to implement our proposed model. Throughout the whole experiment, we used Kaggle notebook to run our code. Our model was trained using our prepared data and Adam optimizer. Adam optimizer is used to optimize training of our dataset. We have implemented the EarlyStopping (only for X-ray images) which will stop training that particular training set when there are no fluctuations in consecutive 7 epochs. We have also implemented the ReduceLROn-Plateau method which decreases the learning rate if there are no changes detected for 3 epochs. We have set the value of the learning rate as 0.001. Batch size is 32 and each epoch has the value of 250. While training the dataset, we have shuffled the data before each epoch. We have used a 5-fold cross-validation approach for 3 label and 2 label classification. We have divided the training set into 5 parts on a random basis and each of these subsets has the same number of values. Among the 5 parts, 4 subsets were considered for training CoroPy whereas the other one was used for validation.

3.5 Performance Evaluation

We have calculated Specificity, Sensitivity, Precision, F1 score, accuracy and confusion matrix for each fold to analyze the performance of our model. Firstly, we have measured these values for 3 class 5 folds. Later on, we repeated the process for 2 class classification. We have shown performance for each fold. CoroPy has given a good result for all the folds. We have evaluated our result and compared it with other models in Chapter 5.

Chapter 4

Model Analysis and Implementation

4.1 Model Analysis

This research aims to find which method of covid-19 detections is more suitable, detection. For detecting covid-19 patients using their chest x-ray and ct-scan image dataset, we are proposing a 19 layer CNN model, CoroPy. The proposed model is composed of 10 conv2d layers, 5 pooling layers, 2 dense or fully connected layers and 1 dropout layer. The whole process has been implemented using Tensorflow by google. Details about the proposed model are provided in the sections below.

4.1.1 2D Convolutional Layers

2D Convolutional layer or Conv2D (in TensorFlow) is a function that provides the feature set of a convolutional neural network. The working process of a convolutional layer is quite simple. This layer takes input as an array of numbers where the input data's shape should be specified. In our case, we were working with image data so we needed to convert the image to an array of numbers. While converting the image to a 2D number array each cell of the array will contain the colour value of the image pixels. After converting images to an array we need all images to be of similar height and width. This will ensure that the convolutional layer won't get any inconsistent data. To simplify, the shape of the image array and the input shape of the convolutional layer should be equal.

After processing the images it is passed on to the convolutional layer. Now each convolutional layer contains a kernel. Where the kernel is a matrix consisting of weights that slides over the images. In most common use cases the kernel size is kept to (3 X 3) but it may also vary. In our case, we kept kernel size to (3 X 3).

The purpose of this kernel is to slide over the image array from the top left corner ending at the bottom right corner. The kernel moves pixel by pixel till all the image array is covered. While moving over the image array, the kernel performs multiplication with each element of the input part it is currently on and then sums up the result into a single output pixel. Quite similar to the Figure 4.1.

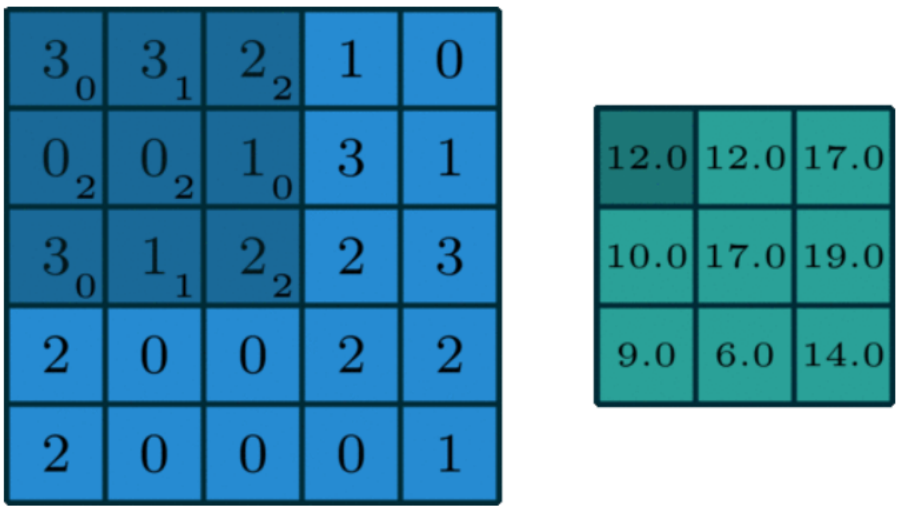


Figure 4.1: Convolutional layer

This multiplication and summing up process continues till the whole array is parsed. This process converts the input matrix of features to output matrix features. The 2D output matrix holds the weighted sums of the input features. These can be called neurons. They are the main part of a CNN model. Whenever a pixel from the input images matches any of the weights a neuron is triggered and works like a chain reaction till it is able to make a decision.

All convolutional layers require an activation function which is ReLU in our case. It is linear and will compare the input. If it is positive then it will take the input directly otherwise output will be zero. The process is quite simple. The function just needs to return the max value between 0 and the input value. To solve the border effect issue, we kept the padding of the convolutional layers to 'same'.

4.1.2 Downsampling Input Data with Pooling Layer

The pooling layer down samples the input layer. The main reason for using this layer is to downsample the input params, which reduces its dimensions. Usually the pooling layer is used after any convolutional layer. After getting input, the pooling layer then down-samples the input layer and outputs a downsampled layer. In order to generate a downsampled layer, there are different types of pooling layers. For example, MaxPooling2D, AvgPooling2D, SumPooling2D and the same for 3D input layers.

Each type of pooling layer does as its name suggests. The working process of a pooling layer is quite simple. For example, if 350 X 350 by image data is passed to a max-pooling layer. It will move the filter over the image data similar within a convolutional layer. For each square matrix data within that captured filter matrix, it will only keep the max value and create a matrix like in Figure 4.2. For an average pooling layer, it will calculate the average value within the captured filter matrix and for the sum pooling layer, it will calculate the sum of the values within the captured filter matrix. After calculation and storing the value to the new matrix the filter will then move to the next position. This process continues until the whole input layer is traversed.

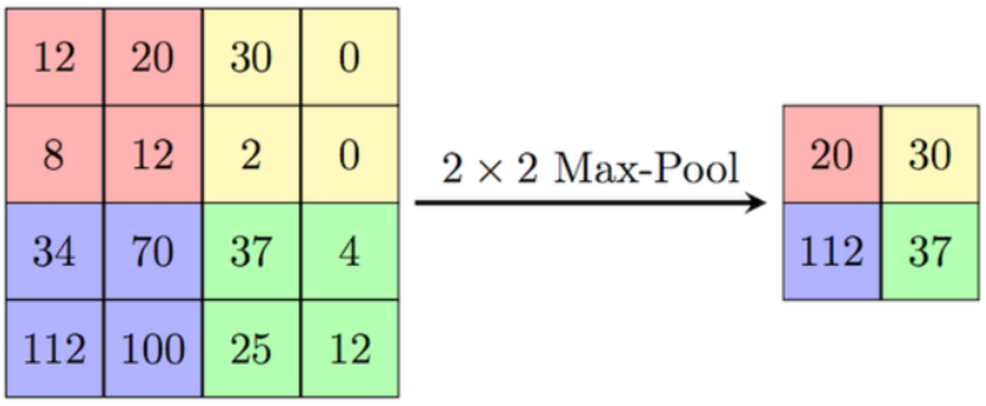


Figure 4.2: Max Pooling Matrix Representation

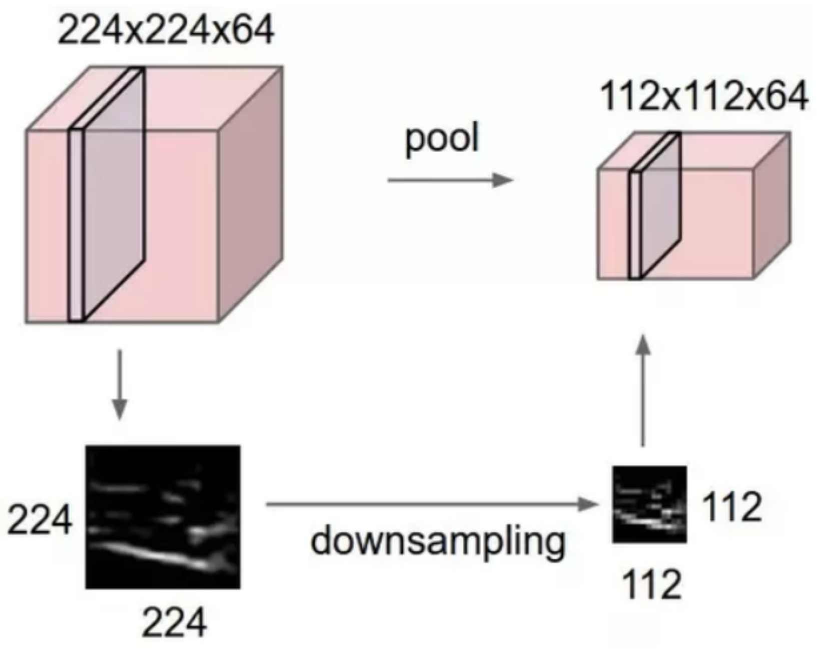


Figure 4.3: Max pooling down sampling example

In our case, we used MaxPooling2D and AvgPooling2D provided by TensorFlow API. In total, we used 3 max-pooling and 2 average pooling layers as this combination of pooling layers provided the best results. Each pooling layer had (2 X 2) filter size. By downsampling like in Figure 4.3 the layer we will be able to get more accurate results.

4.1.3 Flatten Layer

Our 19 layer CNN model or CoroPy model consists of one Flatten layer. It is used to collapse the spatial dimensions of the input into the channel dimension. The job of the flatten layer is to convert the 2 dimensional matrix into a single dimension and pass it down for further processing. Usually, each CNN model has one Flatten layer. Flattening layer removes all dimensions of the input layer and converts it to a single column matrix like in Figure 4.4.

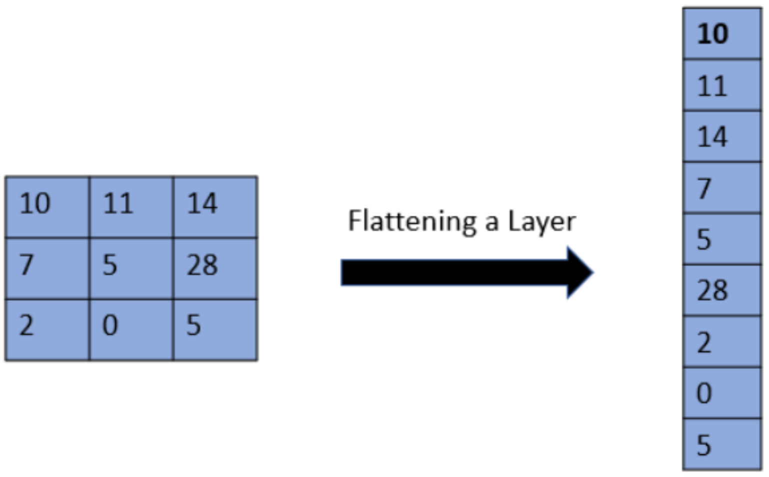


Figure 4.4: Flatter layer

4.1.4 Making Decision with Dense Layer

Every CNN model must have at least one dense layer. A dense layer or fully connected layer is responsible for deciding which features match motley with a specific class like in Figure 4.5. The dense layer works with features with a specific weight which can help to determine the end class. That is why a dense layer is the end process of a CNN or DNN model. This layer makes the end connection of the network which outputs the possibility of an image matching the class.

Our model contains 2 Dense layers. The last dense layer is kept dynamic to match the number of classes. This layer outputs the possibility of an image being of Covid class or Normal class or viral pneumonia for 2 class and 3 class. We then need to calculate the max value among the possibilities. The dense layer also requires an activation function. In our case, it is Sigmoid..

A sigmoid activation function generates values between 0 to 1. As we are predicting the possibility of an image being one of the classes, a value between 0 to 1 is quite helpful to us. This also helps to find the max value of the possibility. For this reason, sigmoid is selected as an activation function for the last dense layer.

4.1.5 Dropout Layer

CNN models can overfit if there is less data available for training. In order to fix this problem, we use the dropout layer. What this layer does is that it randomly

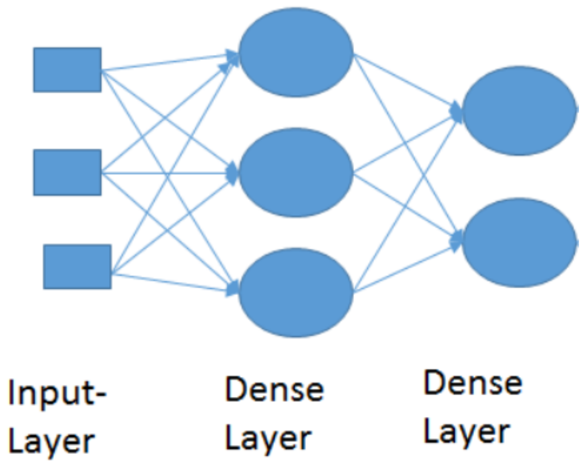


Figure 4.5: Dense layer

drops some layers of the output like in Figure 4.6. This simulates the effect of the new layer being a newly computed layer and the model treats it as such and retrains with the new parameter. This layer also makes the training process noisy. Which tells the model that training is not complete yet and retrain with new params. Our model contains one dropout layer to make sure no overfitting occurs and the model is trained with all possible variations.

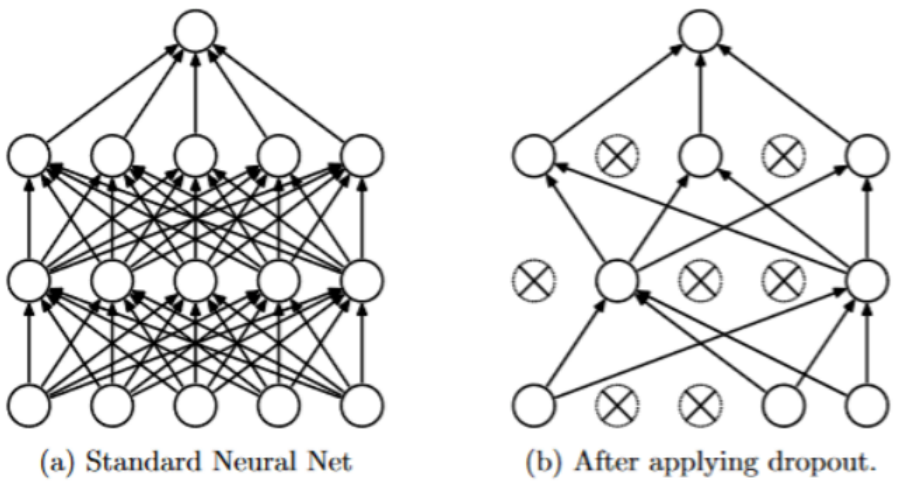


Figure 4.6: Dropout layers

4.2 Model Implementation

For making a comparison on which type of data (x-ray or ct-scan) is best for detecting covid patients we built a 19 layer CNN model. Which is composed of 10 conv2D, 3 Maxpooling2D layers, 2 AvgPooling2D, 2 dense layers and 1 dropout layer and a flatten layer. The working process of our developed system and model implementation is described in the sections below.

4.2.1 Data Processing

For our model to analyze image data first we need to process it. The convolutional layer and all the layers that come after takes input as an array with a specified shape. So we need to convert our image to an array so that the layers can easily read them.

To convert the images to TensorFlow data types we used a method provided by Tensorflow called “flow_from_dataframe”. This method takes a CSV file and image directory as input. Then it parses the image label and path from the CSV file and collects the image from the image directory. After that, it converts them to a tensor of specified shape which is 224 X 224 in our case.

We also introduced image augmentation to generate some random data and make sure the model is introduced to all possible variations of images. We flipped the images horizontally and vertically, zoomed in to the images, sheared the images, made a small degree of rotation and also made height shift and width shift for image augmentation. The data generation library also splits the data into test and train sets which were 80% for training and 20% for testing or validation.

After data processing, we get 2 sets of input tensors for 2 class and 3 class classification. These data are then fitted into the model for training.

4.2.2 Implementation

There are many architectures for creating a CNN model. Our 19 layer CNN model, CoroPy is a sequential model. This means that the input data is passed from the top to bottom pattern. The first layer takes the input data, after processing it passes it to the next layer and so on. Architecture of our model can be seen in the following image.

From the Figure 4.7, we can see that the model starts with a conv2D layer which has a input shape of (224, 224, 64). This layer then passes its output to the max-pooling layer. The MaxPooling2D layer downsamples the input data and passes it on to the convolutional layer 2 and so on. When it finally reaches the last dense or fully-connected layer the input data is converted into an array with a possible probability value for each class.

The model we built consists of 1,178,684 params,1,178,684 trainable params and 0 non-trainable params. Details about each layer can be found in Table 4.1.

This model achieves an accuracy of 95.73% for X-ray 2 class classification, 99.17% for ct scan 2 class classification, 92.45% for x-ray 3 class classification 68.81% for ct scan 3 class classification.

Layer	Output Shape	Param #
conv2d_coro_py	(222, 222, 32)	896
activation_coro_py	(222, 222, 32)	0
max_pooling2d_coro_py	(111, 111, 32)	0
conv2d_1_coro_py	(109, 109, 32)	9248
activation_1_coro_py	(109, 109, 32)	0
max_pooling2d_1_coro_py	(54, 54, 32)	0
conv2d_2_coro_py	(52, 52, 64)	18496
activation_2_coro_py	(52, 52, 64)	0
conv2d_3_coro_py	(50, 50, 64)	36928
activation_3_coro_py	(50, 50, 64)	0
conv2d_4_coro_py	(48, 48, 250)	144250
activation_4_coro_py	(48, 48, 250)	0
conv2d_5_coro_py	(46, 46, 128)	288128
activation_5_coro_py	(46, 46, 128)	0
conv2d_6_coro_py	(44, 44, 128)	147584
activation_6_coro_py	(44, 44, 128)	0
average_pooling2d_coro_py	(22, 22, 128)	0
conv2d_7_coro_py	(20, 20, 64)	73792
activation_7_coro_py	(20, 20, 64)	0
average_pooling2d_1_coro_py	(10, 10, 64)	0
conv2d_8_coro_py	(9, 9, 256)	65792
activation_8_coro_py	(9, 9, 256)	0
conv2d_9_coro_py	(8, 8, 256)	262400
activation_9_coro_py	(8, 8, 256)	0
max_pooling2d_2_coro_py	(4, 4, 256)	0
flatten_coro_py	4096	0
dense_coro_py	32	131104
dropout_coro_py	32	0
dense_1_coro_py	2	66
activation_10_coro_py	2	0
Total params: 1,178,684 Trainable params: 1,178,684 Non-trainable params: 0		

Table 4.1: Summary of our 19 layer model

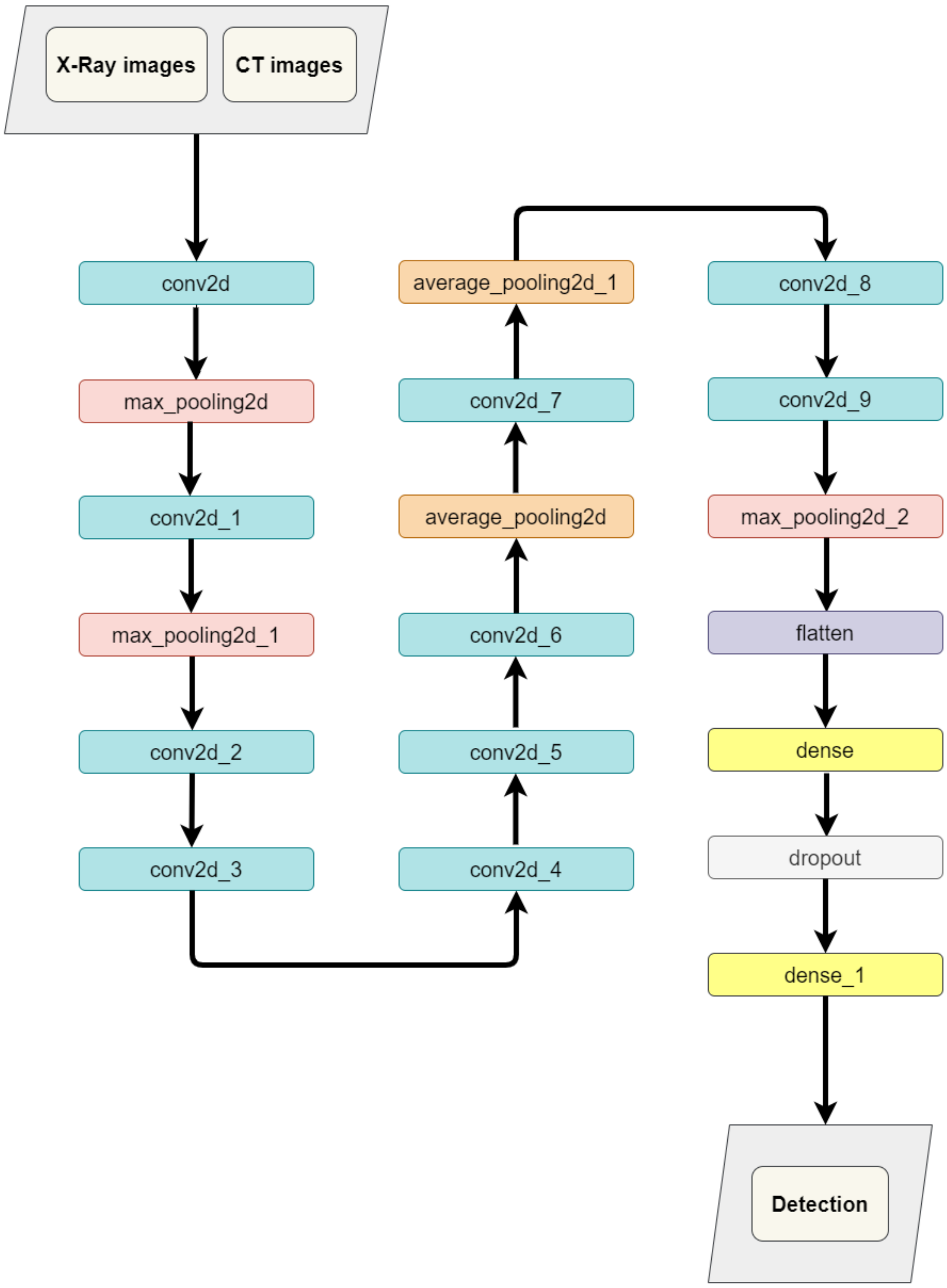


Figure 4.7: Architecture of our model

Chapter 5

Result and Discussion

Our results of detecting Covid-19 using both X-ray images and CT scan images are shown here. The performance evaluation between these images are also displayed here so that we can get a clear idea based on our result analysis. And for that evaluation we have selected a set of parameters like confusion matrix, accuracy, precision etc. We have also shown some of the graphs, charts and tables to have an idea of our model by having an instant look. Then we have compared our model, CoroPy, with some renowned existing models like inception, exception, VGG etc. And lastly, we have given a brief discussion about the performance of our own model, CoroPy.

5.1 Metrics Used to Determine Result

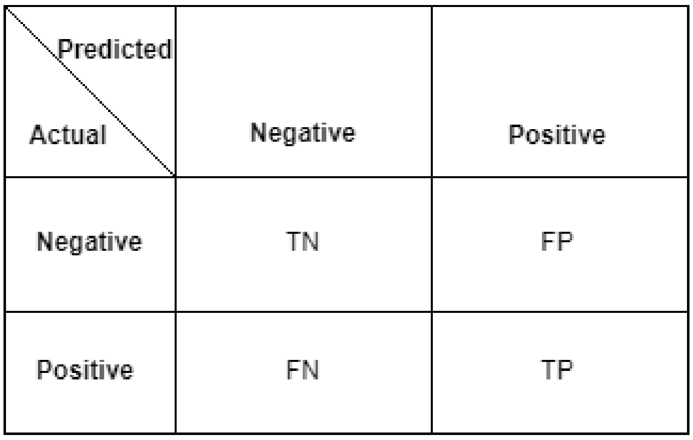
Our model was evaluated based on certain matrices. These are Confusion matrix, Specificity, Sensitivity, Accuracy, Precision, Recall, F1 score etc. Moreover, these matrices are important for measuring most of the models available nowadays. So, before analyzing our proposed model, we have discussed these matrices below to have a better idea about the performance of our model.

5.1.1 Confusion Matrix

It is one of the best processes for understanding the results of a classifier [34]. It shows the short review of classifiers [35]. It gives us the summary in the tabular form which contains the information of actual and the predicted values. However, the size of the confusion matrix is determined by the number of things we want to predict. From Figure 5.1, we can have an idea about the representation of this information in the confusion matrix.

There are some terms associated with this confusion matrix. These are:

- **TN (True Negative):** It represents the actual negative examples those are predicted as negative. That means that predicted values and the actual values match with each other.
- **TP (True Positive):** It represents the actual positive examples those are predicted as positive. That means that predicted values and the actual values match with each other.



Predicted / Actual	Negative	Positive
Negative	TN	FP
Positive	FN	TP

Figure 5.1: Confusion matrix for binary classification

- **FN (False Negative):** It represents the actual positive examples those are predicted as negative. That means the predicted values do not match with the actual values.
- **FP (False Positive):** It represents the actual negative examples those are predicted as positive. That means the predicted values do not match with the actual values.

In summary, confusion matrix tells us what our model did right and what our model did wrong during the prediction.

5.1.2 Specificity

This is used to measure the negative examples those were predicted correctly. It is one of the ways to find out the values of TN for every category [36]. In equation 5.1, we can see the formula for measuring specificity.

$$Specificity = \frac{TN}{TN + FP} \tag{5.1}$$

5.1.3 Sensitivity

This is used to measure the positive examples that were predicted correctly. It is one of the ways to find out the values of TP for every category [36]. This is however known as recall or the true positive rate. In equation 5.2, we can see the formula for measuring sensitivity.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.2}$$

5.1.4 Accuracy

It is one of the ways of understanding the model's performance [37]. It is the ratio between the correct assumptions and all the inputs. In equation 5.3, we can see the formula for measuring accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.3}$$

5.1.5 Precision

In the whole predicted positive result, how many results were actually positives are known as precision. It is also known as positive prediction values. It is the ratio of the true positives to all positives [38]. In equation 5.4, we can see the formula for measuring precision.

$$Precision = \frac{TP}{TP + FP} \tag{5.4}$$

5.1.6 F1 Score

It is the harmonic mean of precision and recall [39]. It ranges from 0 to 1 and tries to make a balance between precision and recall. In equation 5.5, we can see the formula for measuring F1 Score.

$$F1\_Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5.5}$$

5.1.7 Cross Validation

It is a statistical method to understand the performance of various models and also to test the performance of those models. It is used to reduce the problem of overfitting issues when there is a limited number of data [40]. Different types of cross validation methods are used in different cases [23]. Figure 5.2 shows the 5 fold cross-validation methods.

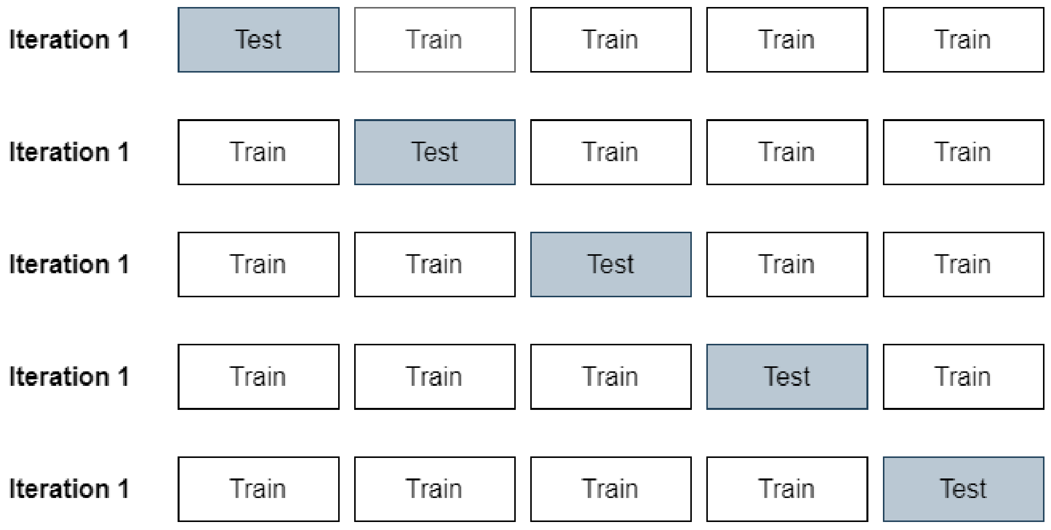


Figure 5.2: K-fold cross-validation

5.2 Performance of Our Proposed Model

We have evaluated our model based on the previously discussed evaluation matrices. Then we have shown the performance of our model on our dataset.

5.2.1 Performances on the X-ray Images

At first, we measured the performance of our model for a 2-class classifier on X-ray images. And the result is given in Table 5.1. Then we measured it for a 3-class classifier. And the result is given in Table 5.2.

Performance metrics	Result
Specificity	0.8
Sensitivity	1.0
Accuracy	0.897
Precision	0.823
F1 Score	0.903

Table 5.1: Performance measurement of X-ray images for 2-class classifier

Performance metrics	Result
Specificity	0.857
Sensitivity	1.0
Accuracy	0.89
Precision	0.778
F1 Score	0.875

Table 5.2: Performance measurement of X-ray images for 3-class classifier

5.2.2 Performances on the CT Scan Images

After measuring the performance on X-ray images, we have measured the performance of the CT scan images. We have measured the performance of our model for a 2-class classifier on CT scan images. And the result is given in Table 5.3. Then we measured it for a 3-class classifier. And the result is given in Table 5.4.

Performance metrics	Result
Specificity	1.0
Sensitivity	0.9
Accuracy	0.957
Precision	1.0
F1 Score	0.947

Table 5.3: Performance measurement of CT scan images for 2-class classifier

Performance metrics	Result
Specificity	1.0
Sensitivity	0.0
Accuracy	0.3
Precision	nan
F1 Score	nan

Table 5.4: Performance measurement of CT scan images for 3-class classifier

5.3 Visualization of Performance

We have also created some graphs and charts to visualize the performance of our model. Those are shown here to have a better understanding for our model evaluation.

5.3.1 Visualization of the Performances on the X-ray Images

At first, we created the confusion matrices for 2-class and 3-class classifiers for X-ray images. These are shown in Figure 5.3 and Figure 5.4 respectively.

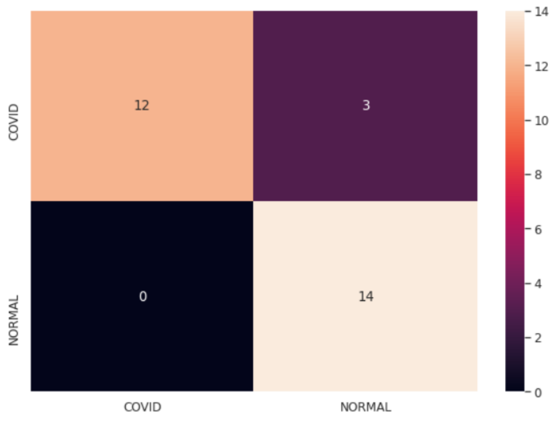


Figure 5.3: Confusion matrix for 2-class (X-ray images)

Then we have created the training and validation accuracy and training and validation loss graphs for X-ray images. And these graphs are shown in Figure 5.5 and in Figure 5.6 for 2-class and for 3-class respectively.

5.3.2 Visualization of the Performances on the CT Scan Images

Then, we created the confusion matrices for 2-class and 3-class classifiers for CT scan images. These are shown in Figure 5.7 and Figure 5.8 respectively.
Then we have created the training and validation accuracy and training and validation loss graphs for CT scan images. And these graphs are shown in Figure 5.9 and

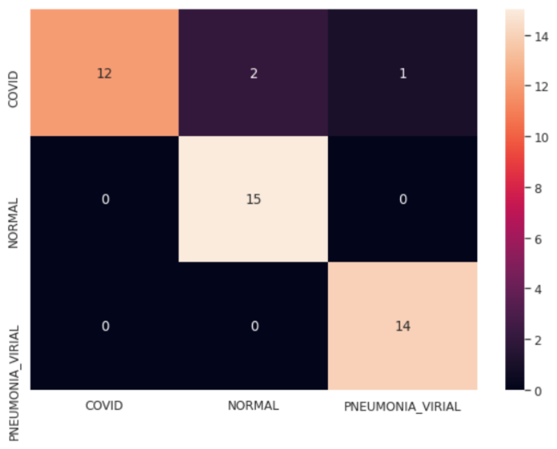


Figure 5.4: Confusion matrix for 3-class (X-ray images)

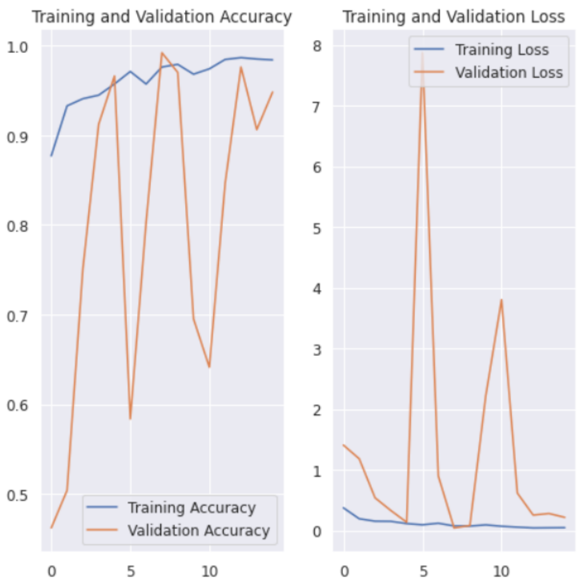


Figure 5.5: Training and validation accuracy and loss for 2-class (X-ray images)

Figure 5.10 for 2-class and for 3-class respectively.

5.4 Result After Cross Validation

We have performed 5 fold cross validation in our model. And the results of X-ray and CT scan images after cross validation are given below.

5.4.1 Cross Validation Results for X-ray Images

The performance evaluation metrics of our model for 5 fold cross validation and average values of those folds of X-ray images are shown in Table 5.5 and Table 5.6 for 2 class and for 3 class respectively.
Then, we created the confusion matrices for 2-class and 3-class classifiers of X-ray images for 5 fold cross validation which are shown in Figure 5.11 and Figure 5.12

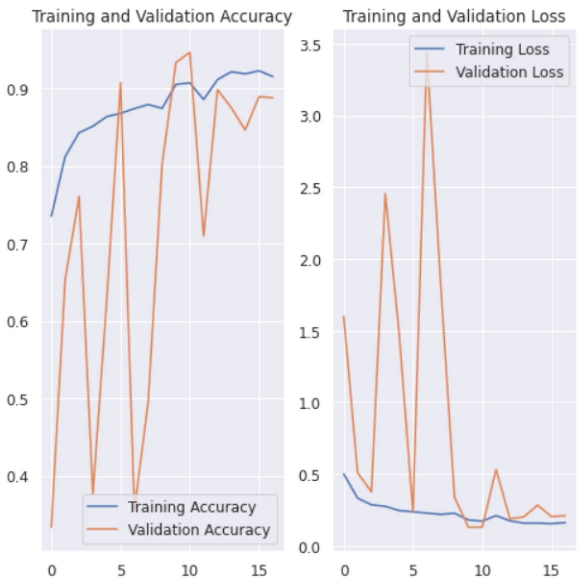


Figure 5.6: Training and validation accuracy and loss for 3-class (X-ray images)

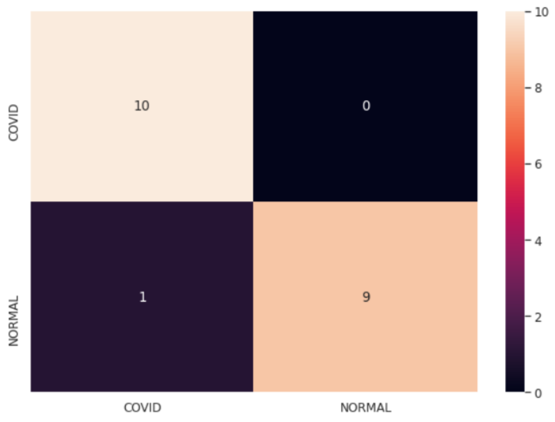


Figure 5.7: Confusion matrix for 2-class (CT scan images)

respectively.

Then we have created the training and validation accuracy and training and validation loss graphs for 2-class and 3-class classifiers of X-ray images for 5 fold cross validation which are shown in Figure 5.13 and Figure 5.14 respectively.

5.4.2 Cross Validation Results for CT Scan Images

The performance evaluation metrics of our model for 5 fold cross validation and average values of those folds of CT scan images are shown in Table 5.7 and Table 5.8 for 2 class and for 3 class respectively.

Then, we created the confusion matrices for 2-class and 3-class classifiers of CT scan images for 5 fold cross validation which are shown in Figure 5.15 and Figure 5.16 respectively.

Folds	Specificity	Sensitivity	Precision	Accuracy	F1 score
Fold 1	0.8	1.0	0.824	0.897	0.903
Fold 2	0.67	1.0	0.737	0.828	0.848
Fold 3	0.67	1.0	0.737	0.828	0.848
Fold 4	0.67	0.929	0.722	0.793	0.813
Fold 5	0.67	1.0	0.737	0.828	0.848
Average	0.696	0.9858	0.7514	0.8348	0.852

Table 5.5: Result of 2-class for 5-fold cross validation (X-ray images)

Folds	Specificity	Sensitivity	Precision	Accuracy	F1 score
Fold 1	0.923	0.0	0.0	0.8	nan
Fold 2	0.857	0.0	0.0	0.8	nan
Fold 3	0.8	nan	0.0	0.8	nan
Fold 4	0.8	1.0	0.25	0.813	0.4
Fold 5	0.8	nan	0.0	0.8	nan
Average	0.836	0.333	0.05	0.8026	0.4

Table 5.6: Result of 3-class for 5-fold cross validation (X-ray images)

Folds	Specificity	Sensitivity	Precision	Accuracy	F1 score
Fold 1	1.0	1.0	1.0	1.0	1.0
Fold 2	1.0	0.0	nan	0.5	nan
Fold 3	1.0	1.0	1.0	1.0	1.0
Fold 4	1.0	0.0	nan	0.5	nan
Fold 5	1.0	1.0	1.0	1.0	1.0
Average	1.0	0.6	1.0	0.8	1.0

Table 5.7: Result of 2-class for 5-fold cross validation (CT scan images)

Folds	Specificity	Sensitivity	Precision	Accuracy	F1 score
Fold 1	1.0	nan	nan	1.0	nan
Fold 2	1.0	nan	nan	1.0	nan
Fold 3	1.0	nan	nan	1.0	nan
Fold 4	1.0	nan	nan	1.0	nan
Fold 5	1.0	nan	nan	1.0	nan
Average	1.0	nan	nan	1.0	nan

Table 5.8: Result of 3-class for 5-fold cross validation (CT scan images)

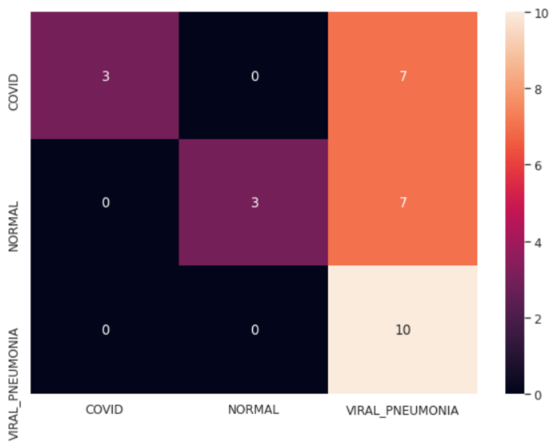


Figure 5.8: Confusion matrix for 3-class (CT scan images)

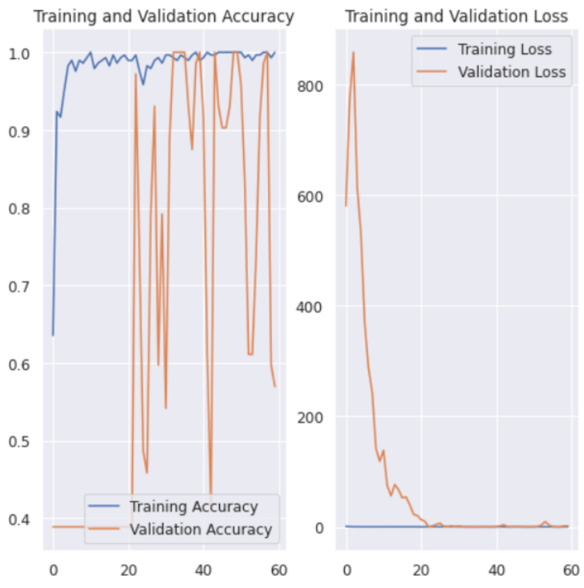


Figure 5.9: Training and validation accuracy and loss for 2-class (CT scan images)

Then we have created the training and validation accuracy and training and validation loss graphs for 2-class and 3-class classifiers of CT scan images for 5 fold cross validation which are shown in Figure 5.17 and Figure 5.18 respectively.

5.5 Comparative Analysis With the Previous Models

We are not the first ones who have made a unique model known as CoroPy. Many one also tried to build models for detecting Covid-19. So, after evaluating our proposed model, we have compared the accuracy of our model with the existing model. And the result is given in Table 5.9. Also we have compared our model with some basic CNN models. And it is shown in Table 5.10 and Table 5.11 for X-ray images

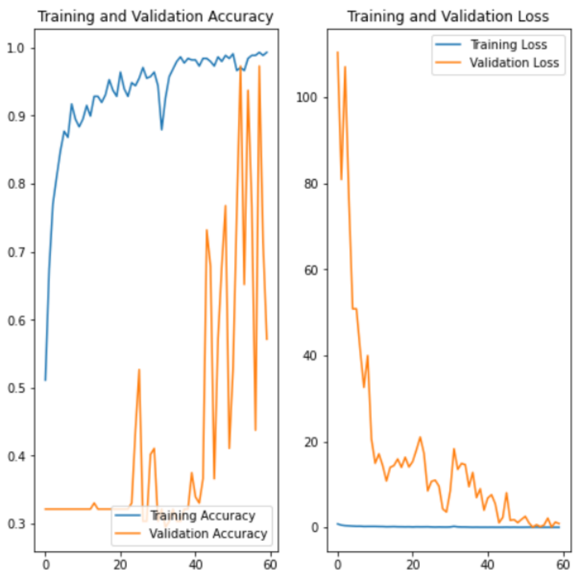


Figure 5.10: Training and validation accuracy and loss for 3-class (CT scan images)

and CT scan images respectively.

Model	X-ray (2 class)	CT scan (2 class)	X-ray (3 class)	CT scan (3 class)
Wang et al.[14]	82.90%	N/A	N/A	N/A
Narin et al.[15]	96.1%	N/A	N/A	N/A
Khan et al.[21]	99%	N/A	N/A	N/A
Zheng et al.[22]	90.08%	N/A	N/A	N/A
Sethy et al.[24]	95.33%	N/A	N/A	N/A
Maghdid et al. [41]	94%	94.1%	N/A	N/A
Shah et al. [42]	N/A	82.1%	N/A	N/A
Panwar et al.[43]	89.47%	95%	N/A	N/A
Hussain et al. [1]	99.1%	N/A	94.2%	N/A
CoroPy	95.73%	99.17%	92.45%	68.81%

Table 5.9: Comparison of accuracy of our model with the existing models

5.6 Discussions

Many researches are done for detection of Covid-19. And much other research is also happening. Some of those include detection of Covid-19 using only X-ray images while some others tried to detect it using CT scan images. But, till now, there are very few papers where a comparative analysis for detecting Covid-19 using both the X-ray images and CT scan images are done. And we have found that our model, CoroPy, can detect Covid-19 from X-ray images more precisely than the CT scan images. The accuracy of detecting Covid-19 and Normal patients is 95.73% and 99.17% for X-ray and CT scan images respectively. And the accuracy of detecting Covid-19, Normal patient and Viral Pneumonia is 92.45% and 68.81% for X-ray and CT scan images respectively. Moreover, we have used our dataset to run on some

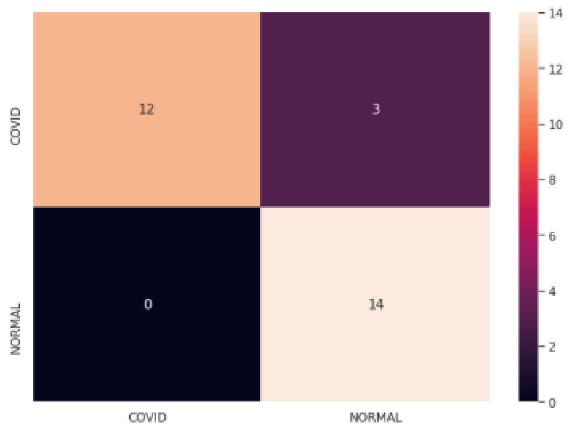
CNN module	Accuracy (2 class)	Accuracy (3 class)
Efficient Net	0.86	0.95
Inception V3	0.79	0.85
MobileNetV2	0.51	1.0
ResNet50	0.48	0.87
VGG16	0.86	0.85
VGG19	0.48	0.0
Xception	0.89	0.76
CoroPy	0.95	0.92

Table 5.10: Comparison with some basic CNN model for X-ray images

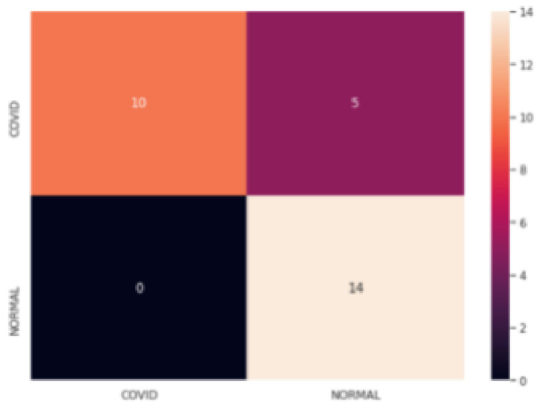
CNN module	Accuracy (2 class)	Accuracy (3 class)
Efficient Net	0.5	0.0
Inception V3	0.5	0.0
MobileNetV2	0.5	0.0
ResNet50	0.5	0.0
VGG16	0.5	1.0
VGG19	0.5	1.0
Xception	0.55	0.0
CoroPy	0.99	0.68

Table 5.11: Comparison with some basic CNN model for CT scan images

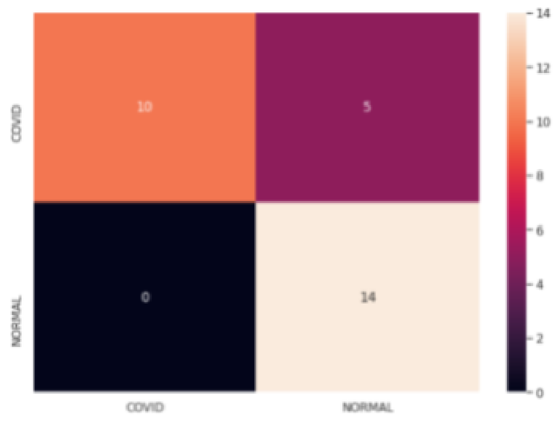
classic and modern deep learning methods. Then we have found that our dataset works best in our own model. So, we can say that our model, CoroPy, is one of the best models for detecting Covid-19.



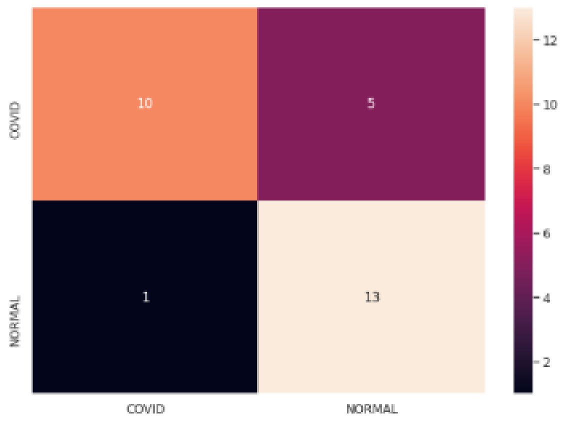
(Fold 1)



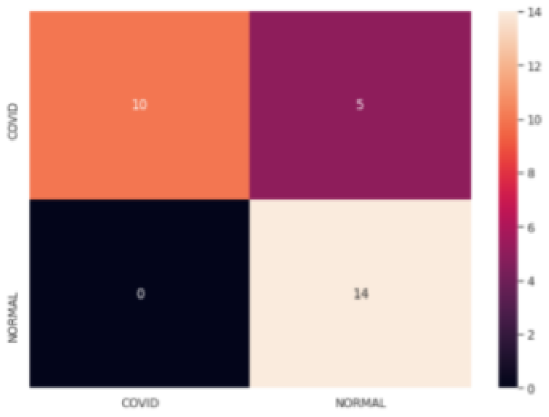
(Fold 2)



(Fold 3)

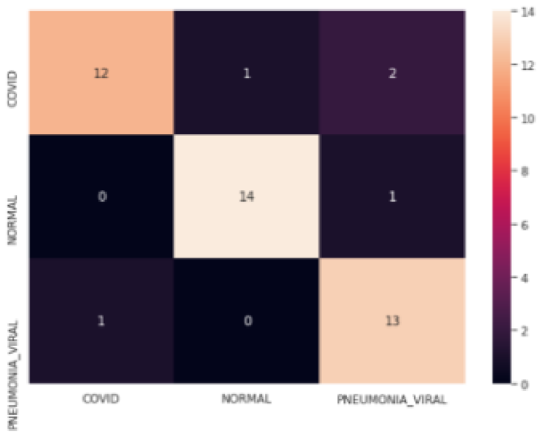


(Fold 4)

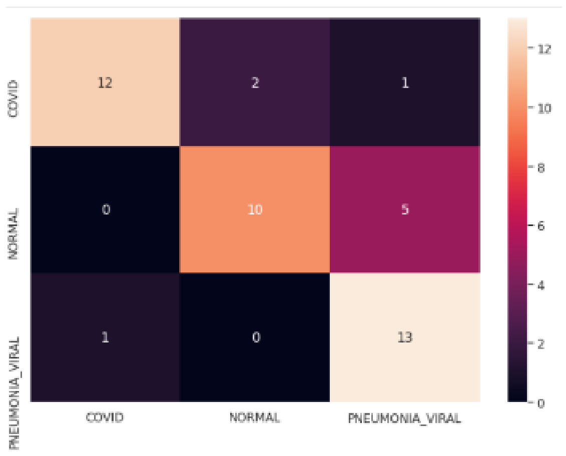


(Fold 5)

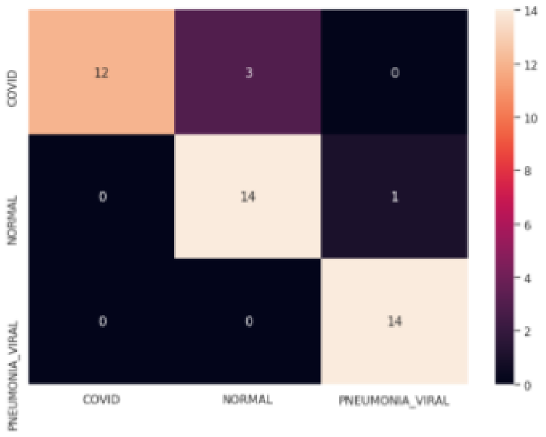
Figure 5.11: Confusion matrix for 5 folds in 2-class (X-ray images)



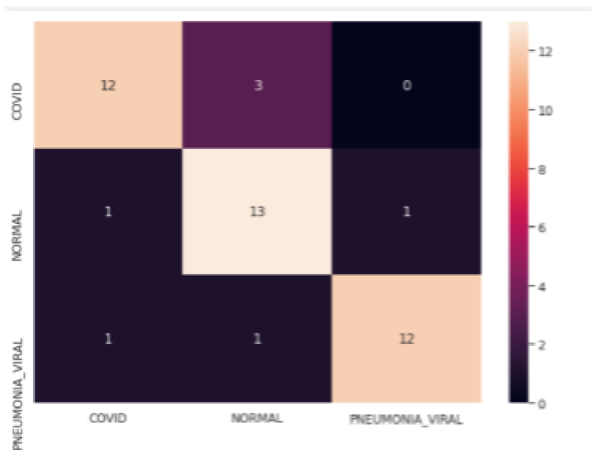
(Fold 1)



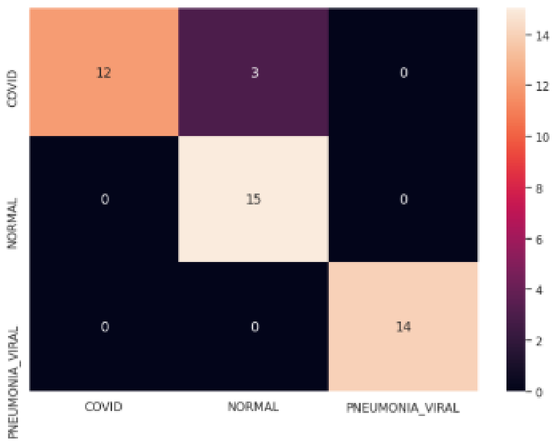
(Fold 2)



(Fold 3)

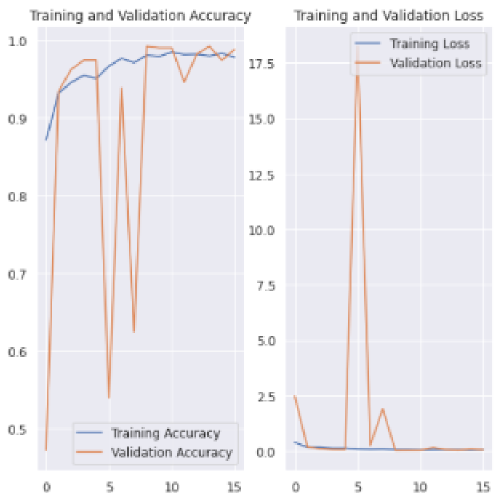


(Fold 4)

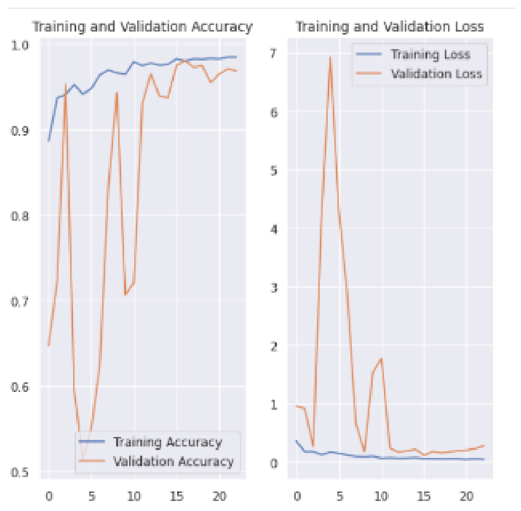


(Fold 5)

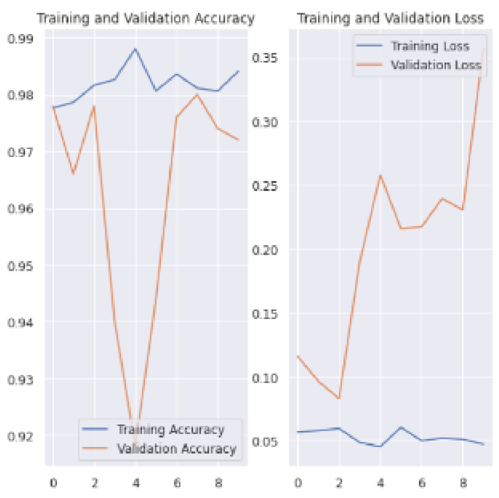
Figure 5.12: Confusion matrix for 5 folds in 3-class (X-ray images)



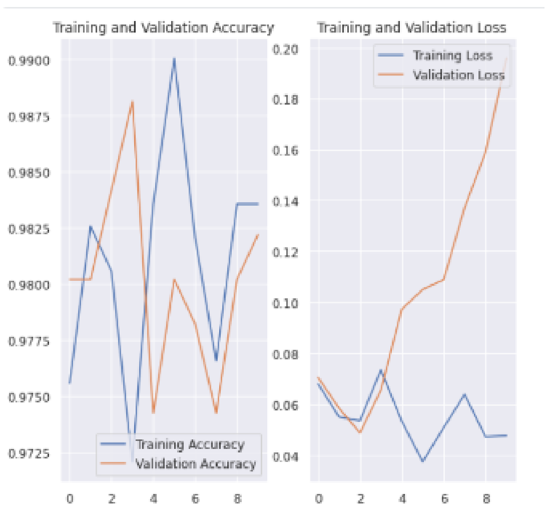
(Fold 1)



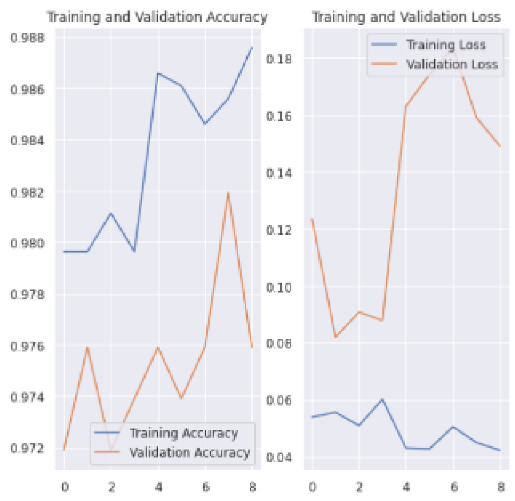
(Fold 2)



(Fold 3)



(Fold 4)



(Fold 5)

Figure 5.13: Training and validation accuracy and loss for 5 folds in 2-class (X-ray images)

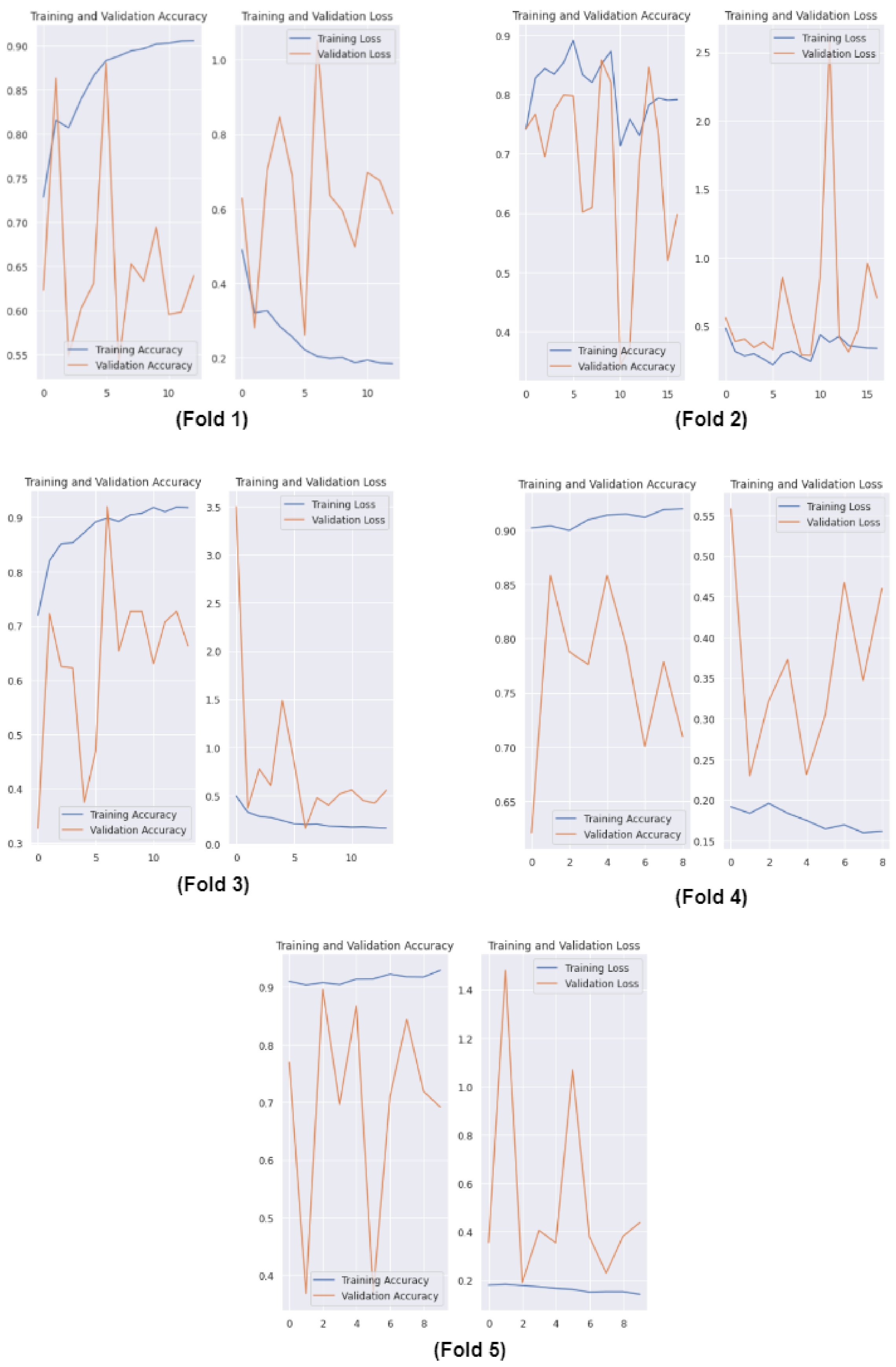
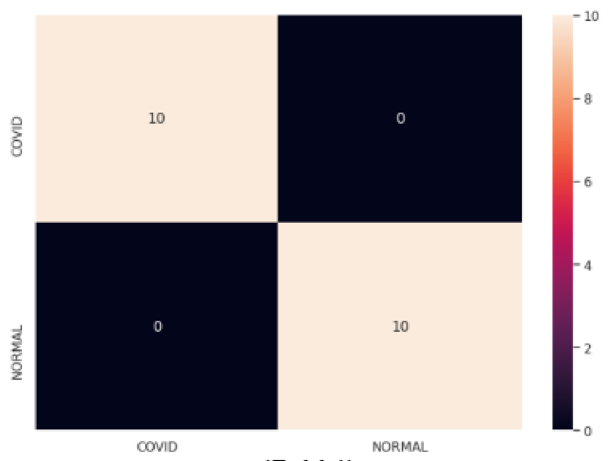
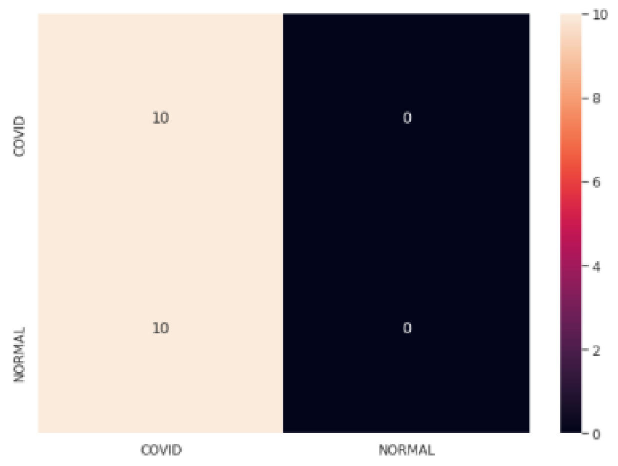


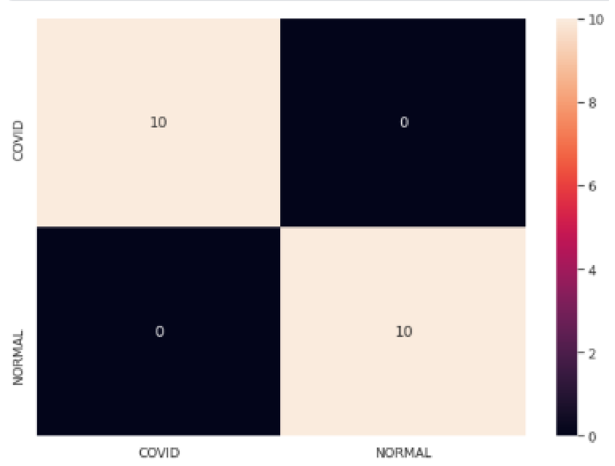
Figure 5.14: Training and validation accuracy and loss for 5 folds in 3-class (X-ray images)



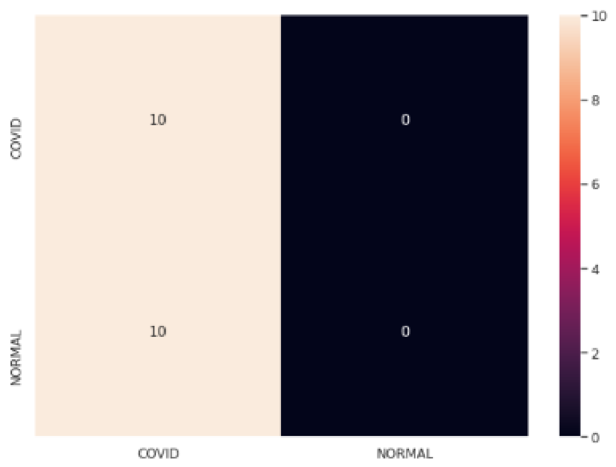
(Fold 1)



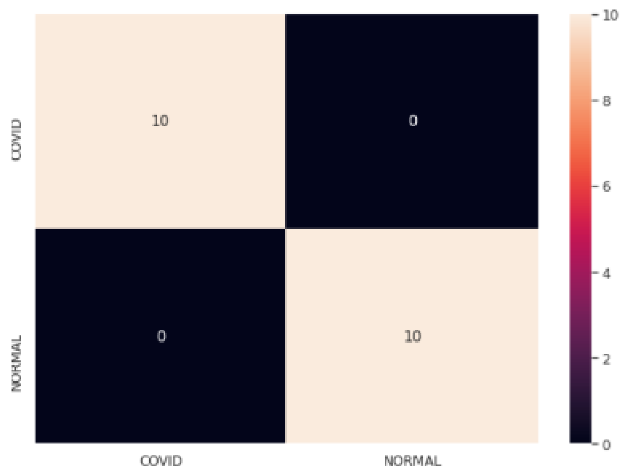
(Fold 2)



(Fold 3)

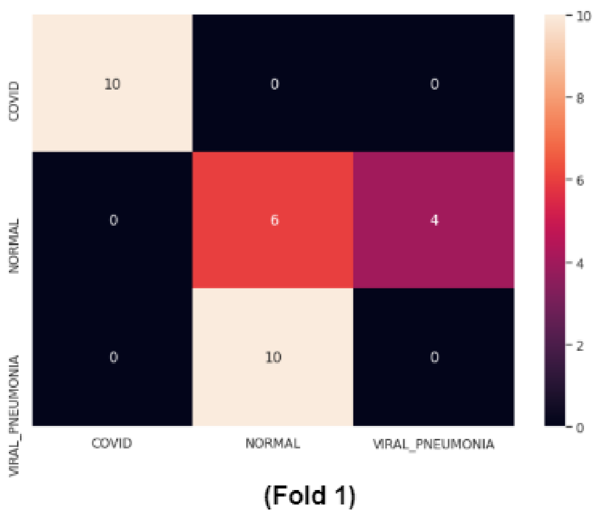


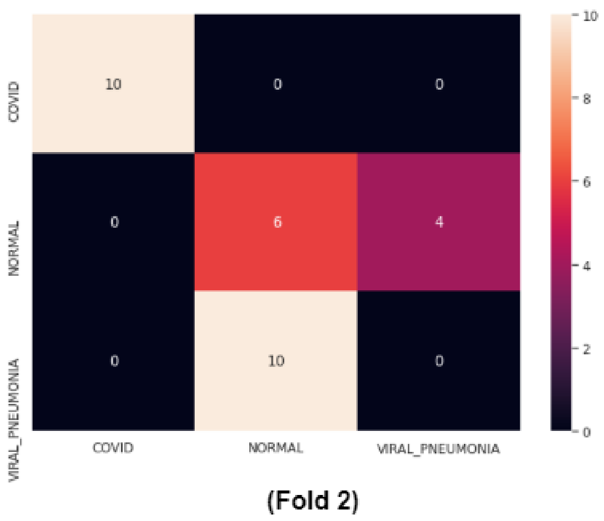
(Fold 4)

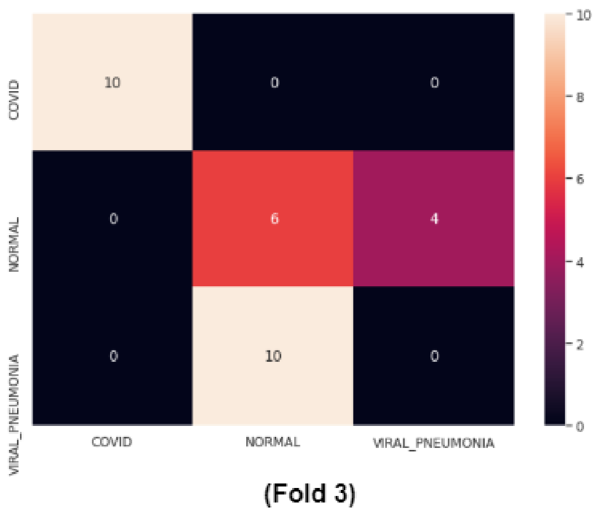


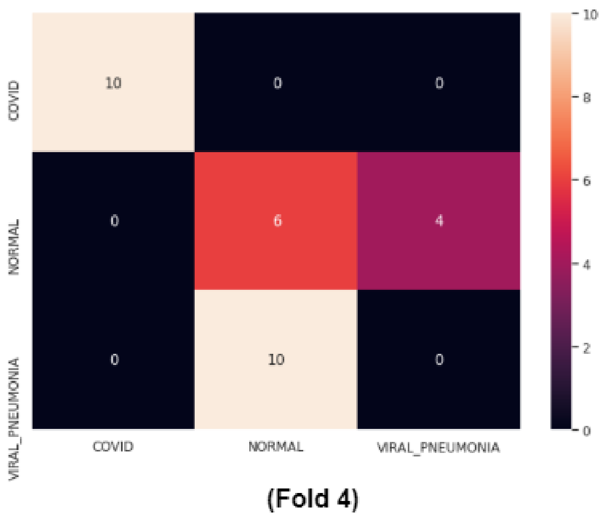
(Fold 5)

Figure 5.15: Confusion matrix for 5 folds in 2-class (CT scan images)









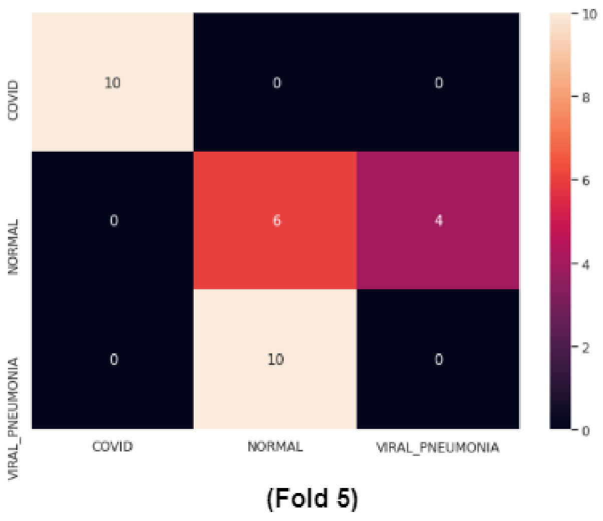
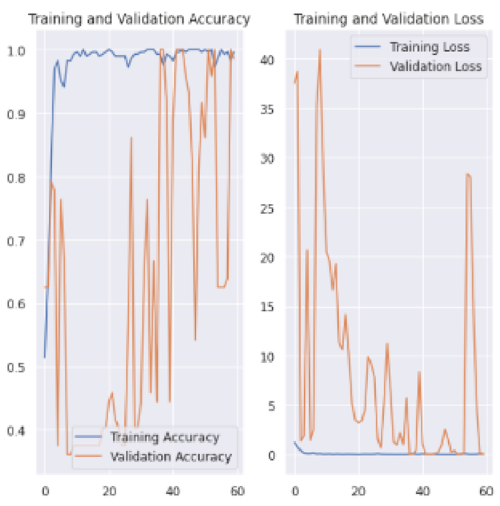
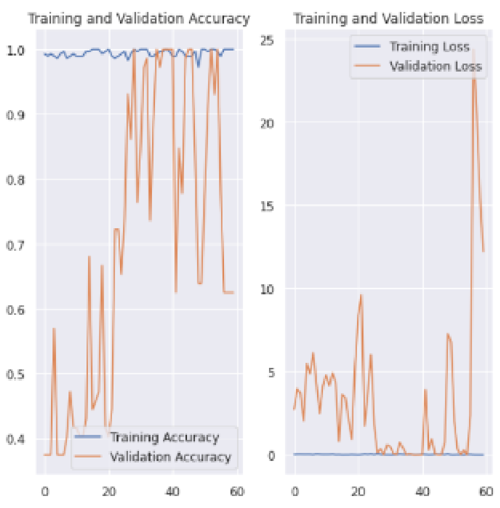


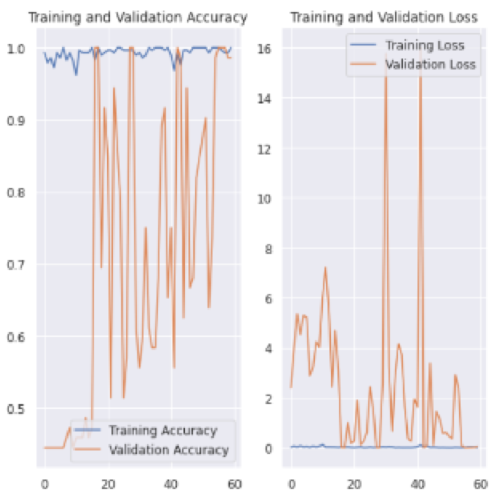
Figure 5.16: Confusion matrix for 5 folds in 3-class (CT scan images)



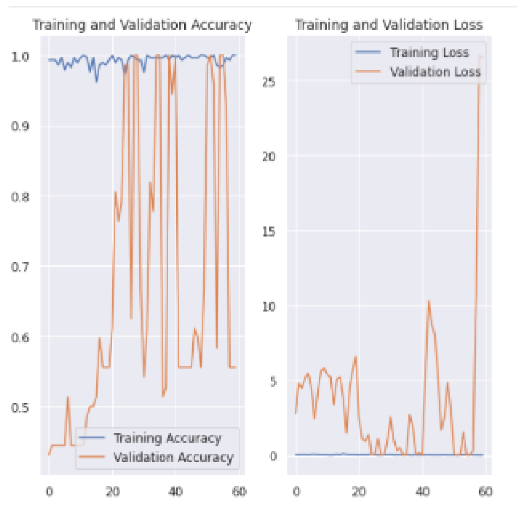
(Fold 1)



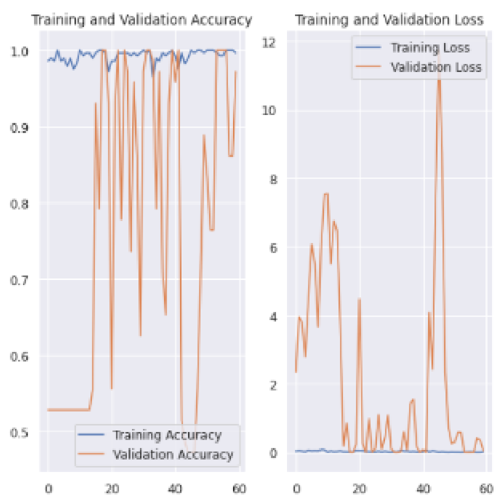
(Fold 2)



(Fold 3)

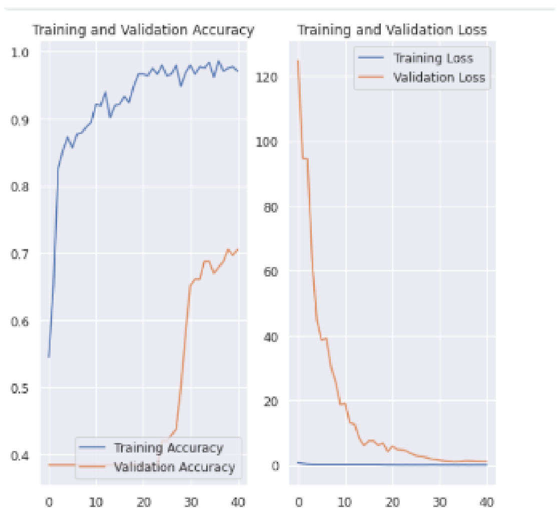


(Fold 4)

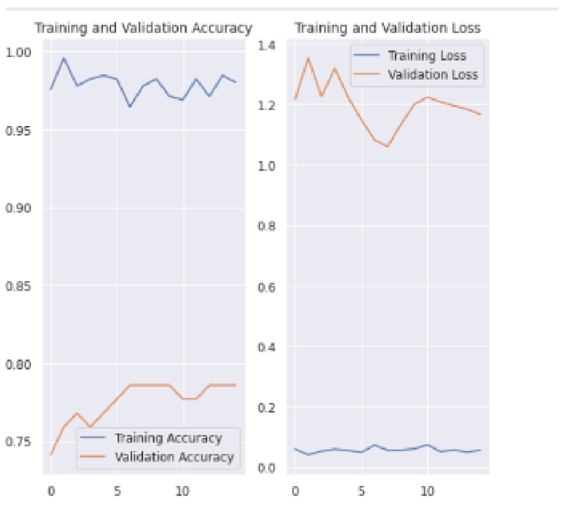


(Fold 5)

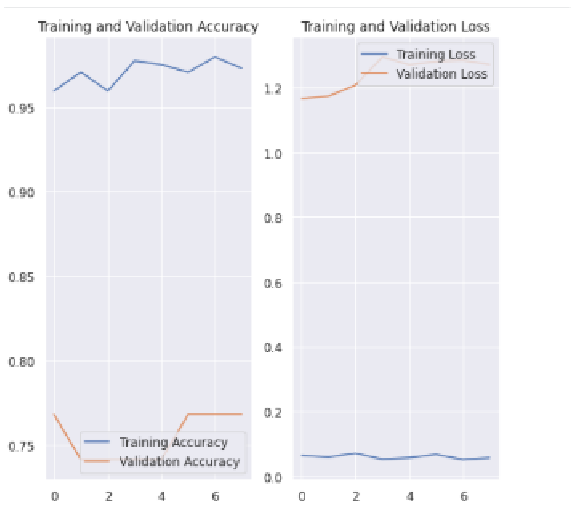
Figure 5.17: Training and validation accuracy and loss for 5 folds in 2-class (CT scan images)



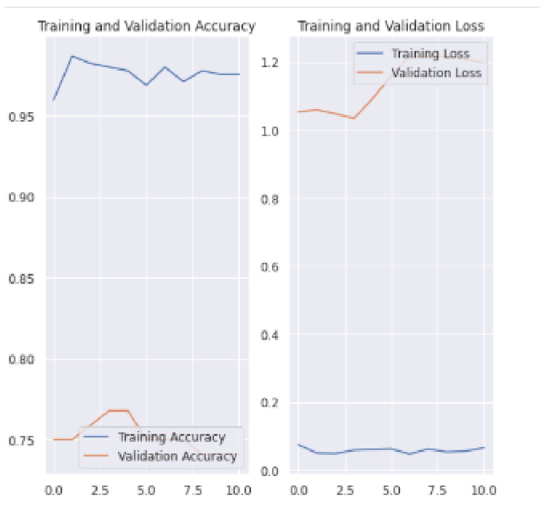
(Fold 1)



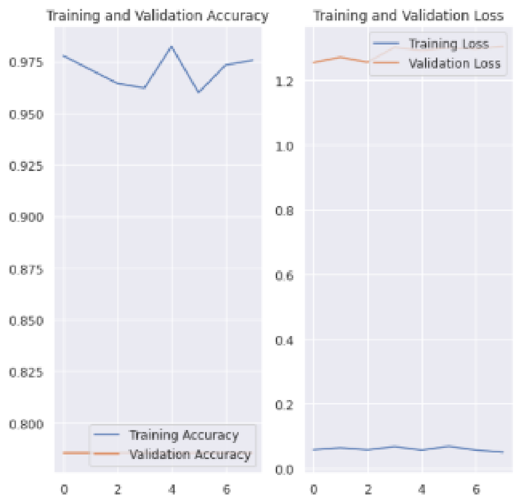
(Fold 2)



(Fold 3)



(Fold 4)



(Fold 5)

Figure 5.18: Training and validation accuracy and loss for 5 folds in 3-class (CT scan images)

Chapter 6

Conclusion

6.1 Conclusion

COVID-19 has caused a devastating effect in our daily life and global economy and also has a dreadful impact in the world health sector. In this thesis, we have done a comparative study of chest x-ray images and CT scan images in detecting covid-19. In order to do that, we have proposed a novel CNN model named CoroPy which works well for both x-ray, CT scan images and provides results for 2 class (normal and covid) and 3 class (normal, covid and viral pneumonia) for both the image types. By analyzing the obtained results, we have seen that in the case of 2 class classification, CT scan images work better for our proposed model where it has an accuracy of 99.17% for CT scan images and 95.73% for x-ray. However, in 3 class, the images of chest x-ray work better for our model where it has an accuracy of 92.45% for chest x-ray and 68.81% for CT scan images.

6.2 Future Plan

In this research, we have worked for 2 class (normal and covid) and 3 class (normal, covid and viral pneumonia), but couldn't work for 4 classes (normal, covid, viral pneumonia, and bacterial pneumonia) due to hardware limitations and also due to limitation of the availability of dataset. Therefore, our future plan is to work on 4 class classification. We are also planning to create an app with our proposed model where people will be able to upload x-ray images or CT scan images and get results within a blink of an eye.

Bibliography

[1] Hussain, E., Hasan, M., Rahman, M. A., Lee, I., Tamanna, T., Parvez, M. Z. (2021). CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images. Chaos, Solitons Fractals, 142, 110495.

[2] Cucinotta, D., Vanelli, M. (2020). WHO declares COVID-19 a pandemic. Acta Bio Medica: Atenei Parmensis, 91(1), 157.

[3] Shrestha, N., Shad, M. Y., Ulvi, O., Khan, M. H., Karamehic-Muratovic, A., Nguyen, U. S. D., ... Haque, U. (2020). The impact of COVID-19 on globalization. One Health, 100180.

[4] Worldometer. (n.d.). COVID-19 CORONAVIRUS PANDEMIC. COVID Live Update. Retrieved from https://www.worldometers.info/coronavirus/

[5] Li, R., Pei, S., Chen, B., Song, Y., Zhang, T., Yang, W., Shaman, J. (2020). Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2). Science, 368(6490), 489-493.

[6] Bruckner, M., Mollerus, R. (2020). COVID-19 and the least developed countries.

[7] Our World in Data. (n.d.). Total COVID-19 tests. Retrieved from https://ourworldindata.org/grapher/full-list-total-tests-for-covid-19?time=2020-02-20..latestcountry=BGD

[8] Macrotrends. (n.d.). Bangladesh Population 1950-2021. Retrieved from https://www.macrotrends.net/countries/BGD/bangladesh/population

[9] Cousins, S. (2020). Bangladesh's COVID-19 testing criticised. The Lancet, 396(10251), 591.

[10] Wise, J. (2020). Covid-19: What's going wrong with testing in the UK?. bmj, 370.

[11] Rosetta Radiology. (2021, April 30). HOW LONG DOES IT TAKE TO GET TEST RESULTS FOR A PROCEDURE? Retrieved from https://rosettaradiology.com/rosetta-radiology-blog/how-long-does-it-take-to-get-test-results-for-a-procedure/

[12] Kumar, A., Gupta, P. K., Srivastava, A. (2020). A review of modern technologies for tackling COVID-19 pandemic. Diabetes Metabolic Syndrome: Clinical Research Reviews.

[13] Pereira, R. M., Bertolini, D., Teixeira, L. O., Silla Jr, C. N., Costa, Y. M. (2020). COVID-19 identification in chest X-ray images on flat and hierarchical classification scenarios. Computer Methods and Programs in Biomedicine, 105532.

[14] Wang, L., Lin, Z. Q., Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. Scientific Reports, 10(1), 1-12.

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

[16] Cohen, J. P., Dao, L., Morrison, P., Roth, K., Bengio, Y., Shen, B., ... Duong, T. Q. (2020). Predicting covid-19 pneumonia severity on chest x-ray with deep learning. arXiv preprint arXiv:2005.11856.

[17] Mahmud, T., Rahman, M. A., Fattah, S. A. (2020). CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization. Computers in biology and medicine, 122, 103869.

[18] Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., Acharya, U. R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine, 103792.

[19] Asif, S., Wenhui, Y., Jin, H., Tao, Y., Jinhai, S. (2020). Classification of covid-19 from chest x-ray images using deep convolutional neural networks. medRxiv.

[20] Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Singh, V. (2020). Application of Deep Learning for Fast Detection of COVID-19 in X-Rays using nCOVnet. Chaos, Solitons Fractals, 109944.

[21] Khan, A. I., Shah, J. L., Bhat, M. M. (2020). Coronet: A deep neural network for detection and diagnosis of COVID-19 from chest x-ray images. Computer Methods and Programs in Biomedicine, 105581.

[22] Zheng, C., Deng, X., Fu, Q., Zhou, Q., Feng, J., Ma, H., ... Wang, X. (2020). Deep learning-based detection for COVID-19 from chest CT using weak label. medRxiv.

[23] JavaTpoint. (n.d.). Cross-Validation in Machine Learning. Retrieved from https://www.javatpoint.com/cross-validation-in-machine-learning

[24] Sethy, P. K., Behera, S. K. (2020). Detection of coronavirus disease (covid-19) based on deep features. Preprints, 2020030300, 2020.

[25] Tuncer, T., Dogan, S., Ozyurt, F. (2020). An automated Residual Exemplar Local Binary Pattern and iterative ReliefF based corona detection method using lung X-ray image. Chemometrics and Intelligent Laboratory Systems, 104054.

[26] Yamashita, R., Nishio, M., Do, R. K. G., Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629.

[27] Stanford University. (n.d.). CS231n: Convolutional Neural Networks for Visual Recognition. Retrieved from https://cs231n.github.io/convolutional-networks/

[28] Cohen, J. P. (2020). covid-chestxray-dataset. Retrieved from https://github.com/ieee8023/covid-chestxray-dataset

[29] UCSD-AI4H. (2020). COVID-CT. Retrieved from https://github.com/UCSD-AI4H/COVID-CT

[30] Eduardo, P. (2020). SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification. Retrieved from https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset

[31] Khoong, W. H. (2020). COVID-19 Xray Dataset (Train Test Sets) with COVID-19 CNN Pneumonia Detector. Retrieved from https://www.kaggle.com/khoongweihao/covid19-xray-dataset-train-test-sets

[32] Rahman, T. (2020). COVID-19 Radiography Database COVID-19 Chest X-ray Database. Retrieved from https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

[33] Yan, T. (2020). COVID-19 and common pneumonia chest CT dataset (416 COVID-19 positive CT scans). Retrieved from https://data.mendeley.com/datasets/3y55vgckg6/2

[34] Nitin1901. (2020). Confusion Matrix in Machine Learning. Retrieved from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

[35] Brownlee, J. (2020). What is a Confusion Matrix in Machine Learning. Retrieved from https://machinelearningmastery.com/confusion-matrix-machine-learning/

[36] Mitrani, A. (2019). Evaluating Categorical Models II: Sensitivity and Specificity. Retrieved from https://towardsdatascience.com/evaluating-categorical-models-ii-sensitivity-and-specificity-e181e573cff8

[37] Gad, A. F. (2020). Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall. Retrieved from https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/

[38] Huilgol, P. (2020). Precision vs. Recall – An Intuitive Guide for Every Machine Learning Person. Retrieved from https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

[39] Mishra, A. (2018). Metrics to Evaluate your Machine Learning Algorithm. Retrieved from https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[40] Mujtaba, H. (2020). What is Cross Validation in Machine learning? Types of Cross Validation. Retrieved from https://www.mygreatlearning.com/blog/cross-validation/

[41] Maghdid, H. S., Asaad, A. T., Ghafoor, K. Z., Sadiq, A. S., Mirjalili, S., Khan, M. K. (2021, April). Diagnosing COVID-19 pneumonia from X-ray and CT images using deep learning and transfer learning algorithms. In Multimodal Image Exploitation and Learning 2021 (Vol. 11734, p. 117340E). International Society for Optics and Photonics.

[42] Shah, V., Keniya, R., Shridharani, A. et al. Diagnosis of COVID-19 using CT scan images and deep learning techniques. Emerg Radiol 28, 497–505 (2021).

[43] Panwar, H., Gupta, P. K., Siddiqui, M. K., Morales-Menendez, R., Bhardwaj, P., Singh, V. (2020). A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. Chaos, Solitons Fractals, 140, 110190.