PREDICTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

DILLI PRASAD REDDY [Reg No: RA1811003010662]
SHAIK FIRDOSE [Reg No: RA1811003010640]

Under the guidance of

Dr. R. Rajkamal
(Associate Professor, Department of Computing Technologies)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that 18CSP109L project report titled "**PREDICTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP LEARNING TECHNIQUES** " is the bonafide work of " **DILLI PRASAD REDDY [Reg No: RA1811003010662]** and **SHAIK FIRDOSE [Reg No: RA1811003010640]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

GUIDE
Dr. R. Rajkamal
Associate Professor
Dept. of Computing Technologies

PANEL HEAD
Dr. K. Vijaya
Associate Professor
Dept. of Computing Technologies

HEAD OF THE DEPARTMENT
Dr. M. Pushpalatha
Professor
Dept. of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

SRM Institute of Science & Technology
School of Computing
Department of Computing Technology

Own Work Declaration Form

This sheet must be filled in and signed with date along with the student registration number, work will not be marked unless this is done.

To be completed by the student:

Degree/Course : BTECH / CSE

Student Name : Kota Reddy Dilli Prasad Reddy , Shaik Firdose

Registration Number : RA1811003010662 , RA1811003010640

Title of Work : Prediction of Covid-19 From Chest X-Ray Images Using Deep learning Techniques.

I / We here by certify that this work compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**. as listed in the University Website, Regulations, and the Education Committee guideline.

I/ We Confirm that all the work contained in this project is my / our own except where indicated, and that I /We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted test (from books, web, etc.)
- Given the sources of all pictures, data etc. that are no: my own
- Not made any use of the reports) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statistician, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University a policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.
If you are Working in group, please write your registration numbers and sign with the date for every student in your group.

Signature of the Student(s)

Sk. Firdose

ACKNOWLEDGEMENTS

I express my humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, and **Dr. T. V. Gopal**, Dean, CET, SRM Institute of Science and Technology for the facilities extended for the project work and his continued support. We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her invaluable support. We wish to thank **Dr. M. Pushpalatha**, Professor & Head, Department of Computing Technologies, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the Project work.

We are extremely grateful to our Panel Head, **Dr. K. Vijaya**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her inputs during the project reviews. We register our immeasurable thanks to our Faculty Advisor, **Ms. D. Vanusha**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, **Dr. R. Rajkamal**, Associate Professor, Department of Computing Technologies, for providing us an opportunity to pursue our project under his mentorship. He provided me the freedom and support to explore the research topics of my interest.

We sincerely thank staff and students of the Computing Technologies Department, SRM Institute of Science and Technology, for their help during my project work. Finally, we would like to thank my parents, our family members and our friends for their unconditional love, constant support and encouragement.

Dilli Prasad Reddy

Shaik Firdose

ABSTRACT

The coronavirus, commonly referred to as Covid-19, is the most severe infectious virus that has spread over the world. The human lungs are affected and damaged by certain infections, which might result in the patient's death. Early detection of virus-infected patients can aid in avoiding the patient from spreading the infection to others. Deep learning algorithms can aid doctors in determining whether a chest x-ray image is Covid positive, Normal, or Pneumonia. The CNN model aids in early viral identification by utilizing chest X-ray pictures, as it is one of the quickest and most cost-effective techniques to determine whether or not a person has covid/pneumonia. We employed six distinct pre-trained CNN models for classification, which were trained using a dataset of 15000 chest x-ray pictures. On the basis of a dataset that contained chest X-ray pictures, the model was trained in three-class classifications on Covid Positive, Pneumonia, and Normal cases. In terms of accuracy 96.5%, the CNN model beat all other models, with precision, recall, and F1-score of 95%, 95%, and 95% for Covid Positive, 98%, 97%, and 97% for Normal, 98%, 94%, and 91% for Pneumonia, and AUC score of 99.5%.

• Keywords—Chest X-Ray, COVID, Pneumonia, Convolutional neural networks (CNNs), ResNet50.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

CNN	Convolutional Neural Network
PIL	Python Imaging Library
RAM	Random Access Memory
CPU	Central Processing Unit
DL	Deep Learning
TPR	True Positive Rate
FPR	False Positive Rate
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

CHAPTER 1

INTRODUCTION

Today the globe is fighting a new disease called Coronavirus disease which is called COVID-19 has identified in December 2019 coronavirus was the foremost infectious disease. it's created a pandemic across the world by showing its worse spread of infection from person to person. This rapid spread of covid-19 has resulted in the new SARS COV2 virus. which has caused a plethora of death cases all across the globe. it's also affected the growth of nations and gradually decreased the economic growth of many countries.It is separated into three stages.

In the initial stage, people feel they are suffering from cold, cough, fever, fatigue, body pains, and rashes on the body. In the second phase, the coronavirus reached its peak performance, by increasing the death rate around the globe. The symptoms people commonly felt in this phase were body pains, loss of appetite, rashes on the skin, and mainly difficulty in breathing with gradually decreasing oxygen levels. People experienced moderate symptoms such as a cold, cough, bodily pains, and fever during the third period.

There were no proper medical tests/kits to check the presence of coronavirus within the patient. within the beginning, there have been some blood tests to test the presence of coronavirus within the patient sample, but it took 2-3 days to get the results which were also not accurate.

Later many companies tried out solutions for this and they came up with a test called the Rapid Antigen test which supplies results within 5 minutes by just collecting the sample from the patient's nose and mouth, although this test isn't accurate and can't determine the stage of the virus within the patient which might help doctors provide the amount of treatment required for various patients.

And later many companies come up with another test called RT-PCR, which takes samples from a patient nose and mouth and mixes them in a solution for knowing results, this test took around 24-48 hours to grasp the result, but it provides accurate and exact results. although this test cannot determine the extent or stage of this virus within the patient body.

Then treatment centres and doctors suggest taking the Chest x-ray of the patient, which could detect the presence of covid within the patient lungs in so that they will start injecting antibiotics accordingly into the patient's body, but these chest X-ray images may give results for other diseases like Pneumonia, Asthma, etc., which may mislead the doctor by knowing the precise disease.

Later, this Covid-19 infection has been divided into three stages mild, moderate and severe. The percentage of coronavirus infection may be known only through CT scans. If the CT-scan score is between 0-7 it comes under the mild stage, and for 7-14 it comes under moderate, and within the case of 14-25, it is a severe stage. many of us constitute mild to moderate within the first wave, moderate to severe within the second wave, and mild within the third wave.

The target of this virus was mainly old people and young people who have low immunity power to fight the infection. This virus hasn't stopped here and it continues by generating more variants. because the number of covid cases increases day by day and new variants of virus too, We'd prefer a fast and accurate way to identify this sickness, and artificial intelligence is the only viable option.

In artificial intelligence, we'll give it a set of photos, such as chest x-rays or CT scans, that contain both covid positive, normal, and other diseases, and it will be able to forecast and offer us with accurate findings based on the input images. The only thing we need to do now is create a dataset of photos from which the model can learn and produce correct results.

Figure 1.1, shows a quick description of the CNN model procedure. The layers of CNN receive the input pictures and predict the outcome.

Many individuals all over the world built machine learning deep learning algorithms to forecast this disease, and many of them performed well in terms of accuracy and other metrics. However, the goal is to create a CNN model that can distinguish between Covid positive, Normal, and Pneumonia patients while also delivering high accuracy on fewer

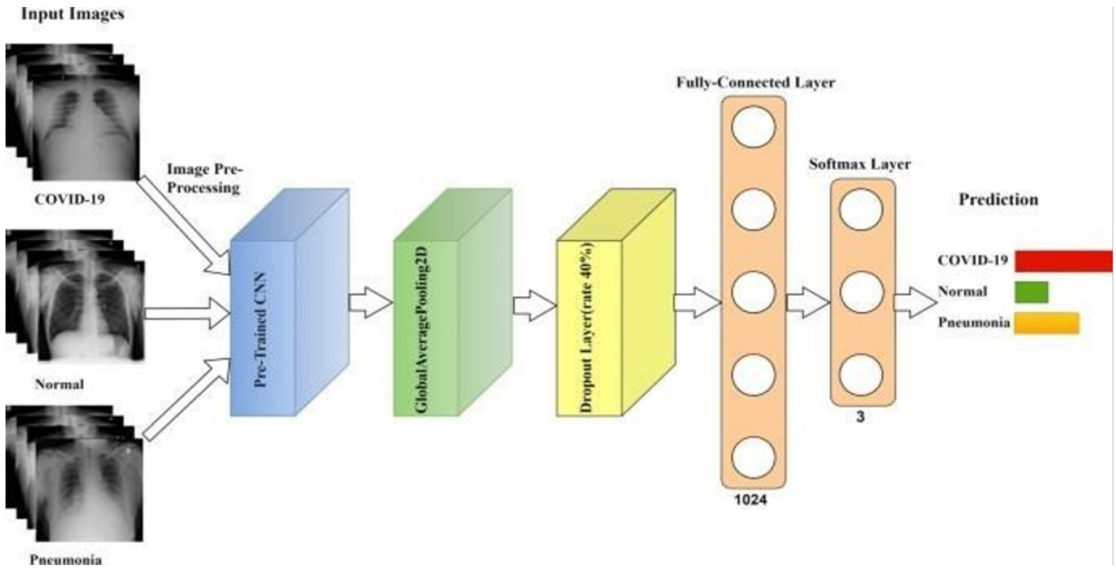


Figure 1.1: CNN model process

datasets.

Because chest x-ray pictures are extremely difficult to induce We will be able to identify between Covid positive, Normal, and Pneumonia with the use of this model, and we may get good accuracy even on smaller datasets. Convolutional Neural Network (CNN) is the best way to do classification and predict the disease. But, the only drawback of CNN is it requires a bigger number of datasets for the purpose of training.

1.1 MOTIVATION

- Anxiety/Fear in a person who has symptoms may lead that person to know the test results whether positive or negative.
- Most of the people go for RT-PCR covid test which can take around 24-48 hours to get results, by this time there is a chance of spread of infection to others.
- By using this project, one can know their test results within short period of time. As a result, controlling the spread of infection.

1.2 OBJECTIVE

• Development of a system for recognising covid patterns/objects on chest X-ray images using well-known Deep Learning techniques.

CHAPTER 2

LITERATURE REVIEW

The authors [1] have employed CNN models for coronavirus(covid-19) identification by employing Chest X-ray pictures. They used 538 COVID +'ve patients' chest x-ray pictures and 468 COVID –'ve patients' chest x-ray images. They [2] employed the weighted average ensemble approach and models DenseNet201, Resnet50, and Inceptionv3. Classification accuracy of 91.62%, Covid '+'Sensitivity of 96.49%, Covid "Sensitivity of 91.67%, Covid '+' F1-score of 92.44%, and Covid '-' F1-score of 92.17% are the metrics used.

In [3, 4] used chest xray images to try to find the one who was suffering from covid or pneumonia. He employed a dataset of 1583 healthy, pneumonia of 4292, and 225 COVID-19 radiograph pictures that have been validated. For improving performance, he employed eightfold cross-validation (a resampling process). The median sensitivity of 93.84%, 99.18% of specificity, and 98.50% of accuracy, and an average ROC AUC of 96.51% is achieved.

In this paper authors [5, 6] used Deep Learning models (VGG19 and U-Net) to train on a chest x-ray image dataset.[7, 8] That applied the approaches like Grayscale, normalization, and standardisation on the dataset to enhance the accuracy of the model. They had a 99% accuracy rate as well as a 98.5% F1-Score. This method demonstrates how traditional technologies can be useful for a variety of duties, as well as how image noise can cause bias in models. Applying the threshold results in a 98% increase in model accuracy.

In this research, [4, 9] the learning and identity map with local weights system to determine if it was covid, pneumonia, or normal. They employed a healthcare chest x-ray dataset as well as weakly-supervised classification and localization standards for prevalent thorax disorders. [10, 11]To balance the dataset, they use the Oversampling approach. They got a 0.9613 correlation coefficient, 0.3121 MAE, 0.3957 RMSE, 62.3354 % relative absolute error, and 79.0202% root relative squared error. Covid, neither any, and pneumonia cases; pneumonitis and also no cases; Covid and pneumonitis cases involving; and Covid and also no cases were enhanced by the SOM-LWL model of correlation from 0.9613 to 0.9788, 0.6113 to 1 0.8783 to 0.9999, 0.8894 to 1.

The writers of [12, 13] employed the CNN Algorithm in this article. The dataset used in this work is split into 2 categories, one COVID and the other Normal, each containing 1,140 photos. After the Convo layer, the activation function was employed to increase the non-linearity in the outcome, and the Grey Scaling approach was used to make the scale of the images equal. They achieved the highest effective accuracy, despite the fact that a huge quantity of pictures is required, with 95% train accuracy and 98% validation accuracy.

[14, 15, 16] employed Deep CNNs models VGG-19, ResNet50, and COVID-net in this study. The dataset included 13,975 CXR pictures from 13,870 patient cases, and the researchers employed a human-machine collaborative design technique. They achieved a sensitivity of 98% and an accuracy of 83% for VGG-19. They acquired an accuracy of 90.6% and a sensitivity of 83% for ResNet50. Covid-Net received a 91% sensitivity and a 93.3% accuracy rating. The Covid –Net model is superior the others, with 93.3% accuracy, 93 covid findings out of 100 covid cases, and 91% sensitivity.

[17, 18, 19] The authors used Deep learning (DLH)-CNN models such as (CNN) and Heat map generation in this paper. This model was using ResNet-101, 101 layers deep as a spine Architecture, the dataset used in fully automated image processing for identifying and classifying irregularities, such as airway obstruction, centralization, incursion, hemothorax, pulmonary edema, respiratory problems, necrosis, fluid overload, and pneumonitis. The suggested approach yields good potential sensitivity, specificity, and accuracy of 97%, 98%, and 98%, respectively, using the convolutional layers of ResNet-101 and U-Net for lung delineation.

In this publication [20, 21, 22] have been using x-Ray images and Learning Techniques with the Cnn model. They used DenseNet201, KNN, Bayes, Random Forest, and Multi-layer perceptron (MLP). They randomly selected 194 chest X-ray pictures from individuals diagnosed with COVID-19 from two datasets A and B. They utilized the Transfer Learning concept to extract characteristics from X-ray pictures.

From such a kernel function, the SVM classifier delivers the desired outcomes in the Mobile Net architecture, with an F1 score of 98.5% and accuracy. The much more successful pair for the other dataset is DenseNet201 with MLP, which obtains accuracy and an F1 score of 95.6%.

In this article, [23, 24, 25] used CNN models such as (CNN), histogram oriented gradient (HOG), river basin fragmentation, and dataset pictures were labeled as of covid positive or non-covid as a benchmark against which the efficiency of the efficient system may be measured. This used convolution layer to use the deep learning method convinced a reasonable success in terms of recognizing covid, especially in contrast to the instant, and 99.49% accuracy, 95.7% specifity, and sensitivity of 93.65% with similar works.

CNNs and some other DL models such as google Net, VGGNet, ResNet, EXCEPTION, Senet, Dense Net, Mobile Net, Shuffle Net, and Alex Net were explored by [26, 27, 28] in this work. A total of eight distinct datasets have been gathered from diverse sources such as GitHub add Kaggle. They employed SMOTE to address the inequity of categories and Data Augmentation to increase the picture to improve quality of the model. They scored 99.46% accuracy, 99.46% precision, 99.73% specificity, 100% sensitivity, 99% AUC Score, and 99.46% F1-Score.

[3, 29]Throughout this study, the most prevalent trained and tested Fully connected layers were discussed. The preprint and published findings for the detection of COVID-19 through CXR-images employing CNN as well as other deep learning architectures are reviewed and critically appraised in this study.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

Our project gathers a patient's chest x-ray picture, saves it, and then analyses it. The picture data is pre-processed and given to the model, which analyses the data and predicts whether the patient has Positive for covid or pneumonia or negative for covid.

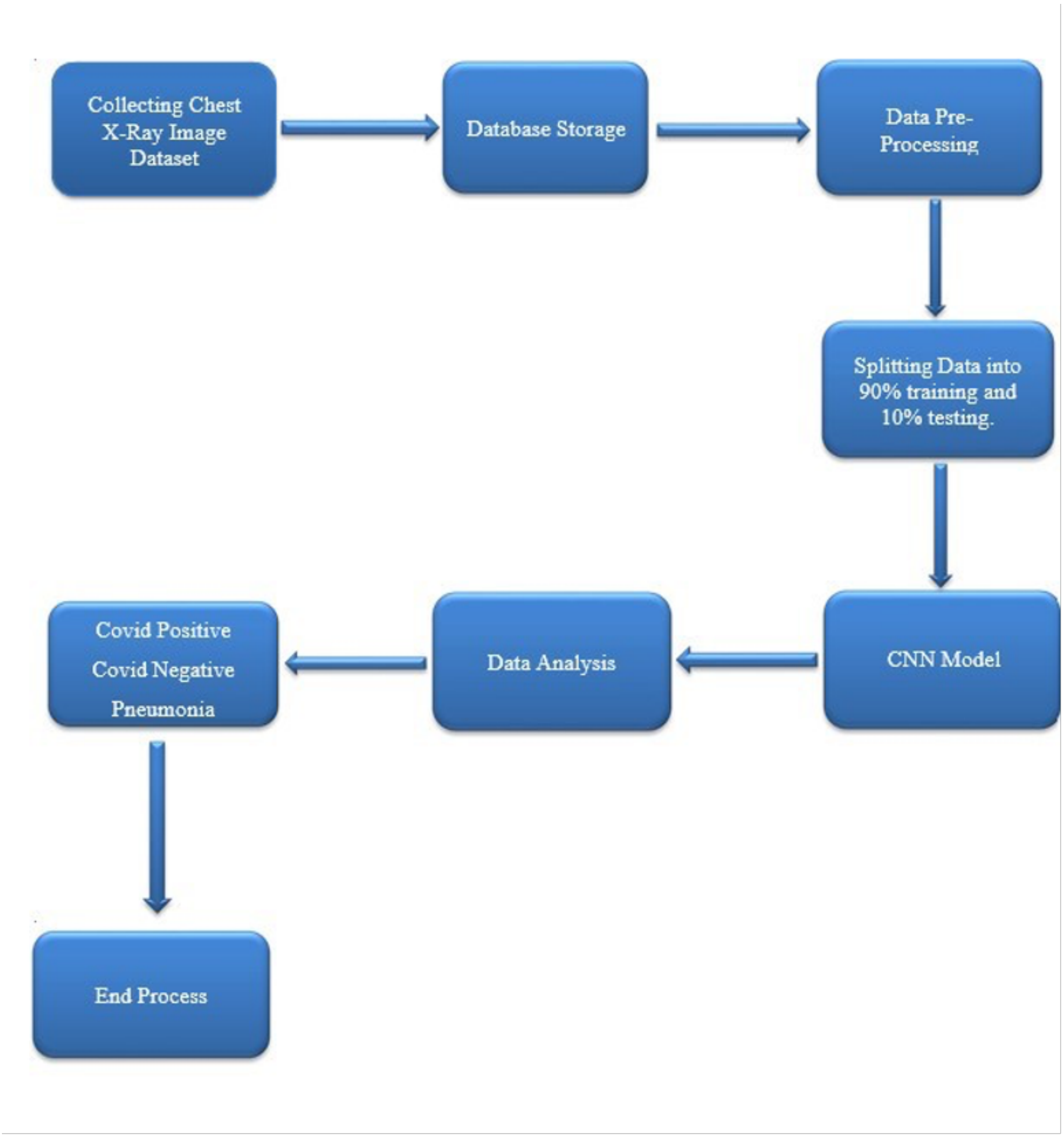


Figure 3.1: Flow chart of CNN model

In figure 3.1, it displays the flow chart diagram that depicts how the project functions. To begin, it gathers CXR images and stores them in a local device. Now we're doing data pre-processing on the model to create our model to operate more smoothly and quickly and make it ready for training. Data pre-processing may be accomplished using a variety of approaches such as data augmentation, grey scaling, and data scaling, this improves our data operationally consistent and aids the CNN model in making judgements. Our data is now ready for training.

To minimize data leaking, we next train, test, and partition the data. At this stage, 90% of the data is used for learning and 10% for validating the system and random state as zero. Now we'll build our CNN model, which will have many convo layers, max-pooling layers, dense layers, dropout layers, loss functions, and activation functions. Following that, data analysis is carried out. It will examine the input image with all the images that have been learned and correlate it with the one that has comparable attributes. Our system will then analyze all of the images in the data to determine whether they are covid positive, pneumonia, or negative. This is the final stage of the procedure.

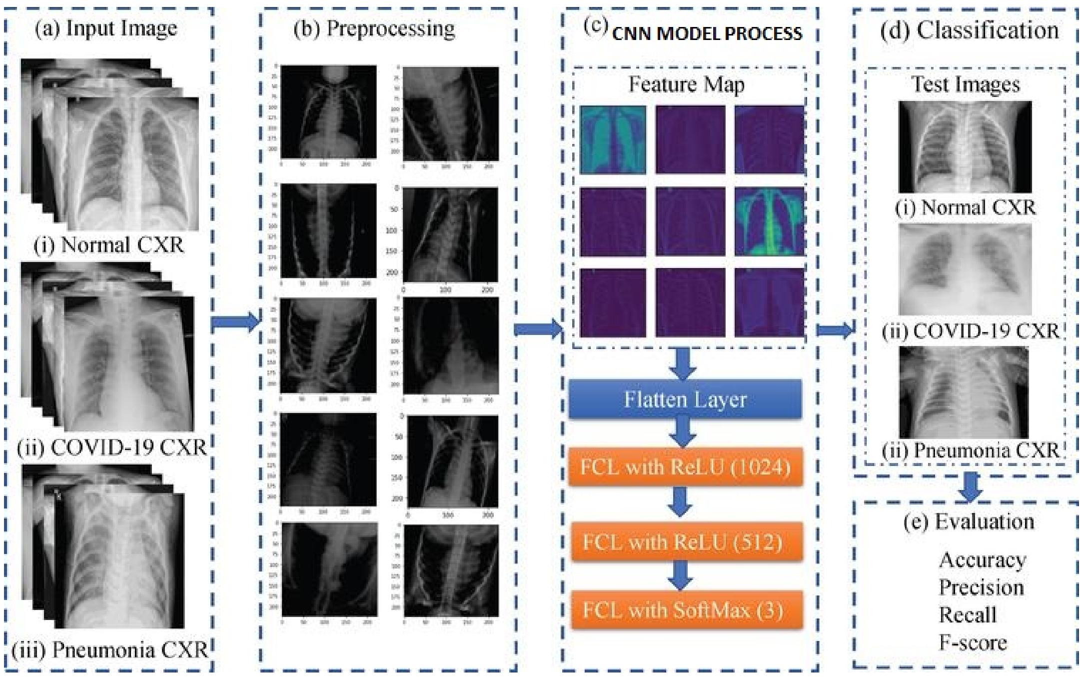


Figure 3.2: Complete working of our project

In figure 3.2, working of our project is demonstrated to be fully functional. Second, data is pre-processed before being submitted to our CNN model, CNN does feature extraction, which is then trained, and at last, the picture is tested before our prediction is made.

3.1 PROGRAMMING ENVIRONMENT

3.1.1 Programming Language(s) used

• **Python 3.8** - Python is an open-source powerful language. Libraries in python helps to make code easier and work smart. Reduces complexity of the code and makes code readable.

3.1.2 Python Packages Used:

1. Numpy
2. Pandas
3. Matplotlib
4. Tensorflow
5. Seaborn

3.1.3 Python Packages Description:

1. **Numpy :-** NumPy library helps in the computation of arrays and large matrices. It contains more number of mathematical Functions which help to make code easier.
2. **Pandas :-** Pandas library is used for Analysing the data, exploring the data, cleaning the data, and manipulating the data. Pandas make it possible to evaluate massive quantities of data and provide conclusions involving the use of statistical theory.
3. **Matplotlib :-** Matplotlib is a library for Python used for creating different types of data visualizations. With help of this we can easily summarize our data.
4. **Tensorflow :-** Tensorflow is a library in Python which provides a collection of workflows to develop and train models using Python.
5. **Seaborn :-** Seaborn library is the advanced functioning of matplotlib. It helps in doing EDA(Exploratory Data Analysis and) and Visualizing the data. Seaborn may also be used to investigate outliers in a collection.

3.1.4 Computer Specifications:

- **Operating System:** Windows 11 x64
- **CPU**: AMD Ryzen 5
- **RAM**: 16GB
- **Graphics Card**: Nvidia GeForce GTX 1660Ti 6GB VRAM
- **Storage**: 512GB SSD
- **GPU**: 8GB

CHAPTER 4

METHODOLOGY

4.1 DATASET

This research endeavour necessitates many Covid Positive, Penumonia and Covid Negative CXR images to predict the coronavirus. Our algorithm was trained on covid, normal, and Pneumonia chest x-ray pictures so that it can predict all three conditions differently. This study makes use of a benchmark dataset containing CXR pictures of covid, Normal, and Pneumonia that was obtained from health care centres and made available on Kaggle.

The following is a list of chest x-ray images used:

• 3616 positive X-ray photos from Covid

• 10192 pictures of normal X-rays

• 1345 x-ray images of viral pneumonia

All images of these classes are in PNG only.

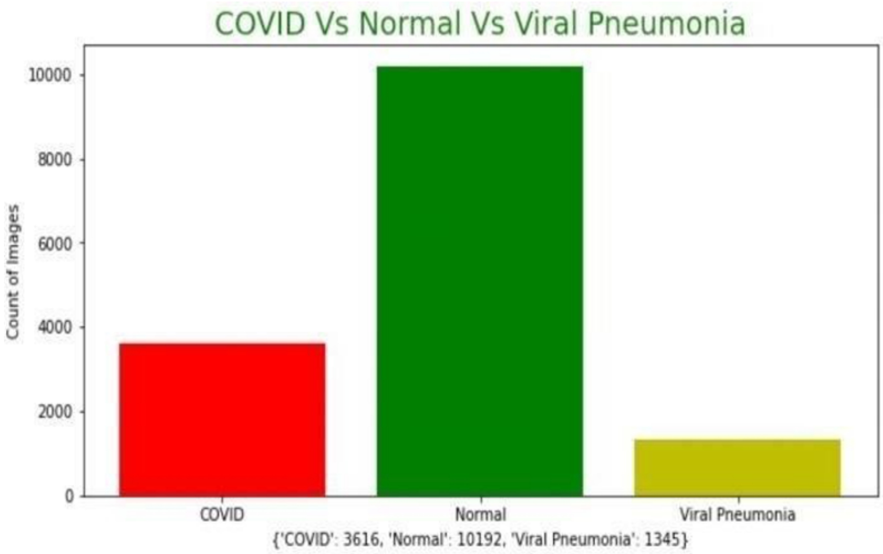


Figure 4.1: Count of chest x-ray images

In figure 4.1, the number of images in our dataset can be seen. In comparison to the other two classes, the number of Normal photos is extremely high. At this stage, it is evident that the dataset is imbalanced. We won't rely on accuracy in this scenario; instead, we'll look at alternative measures that might be used to evaluate an imbalanced dataset.

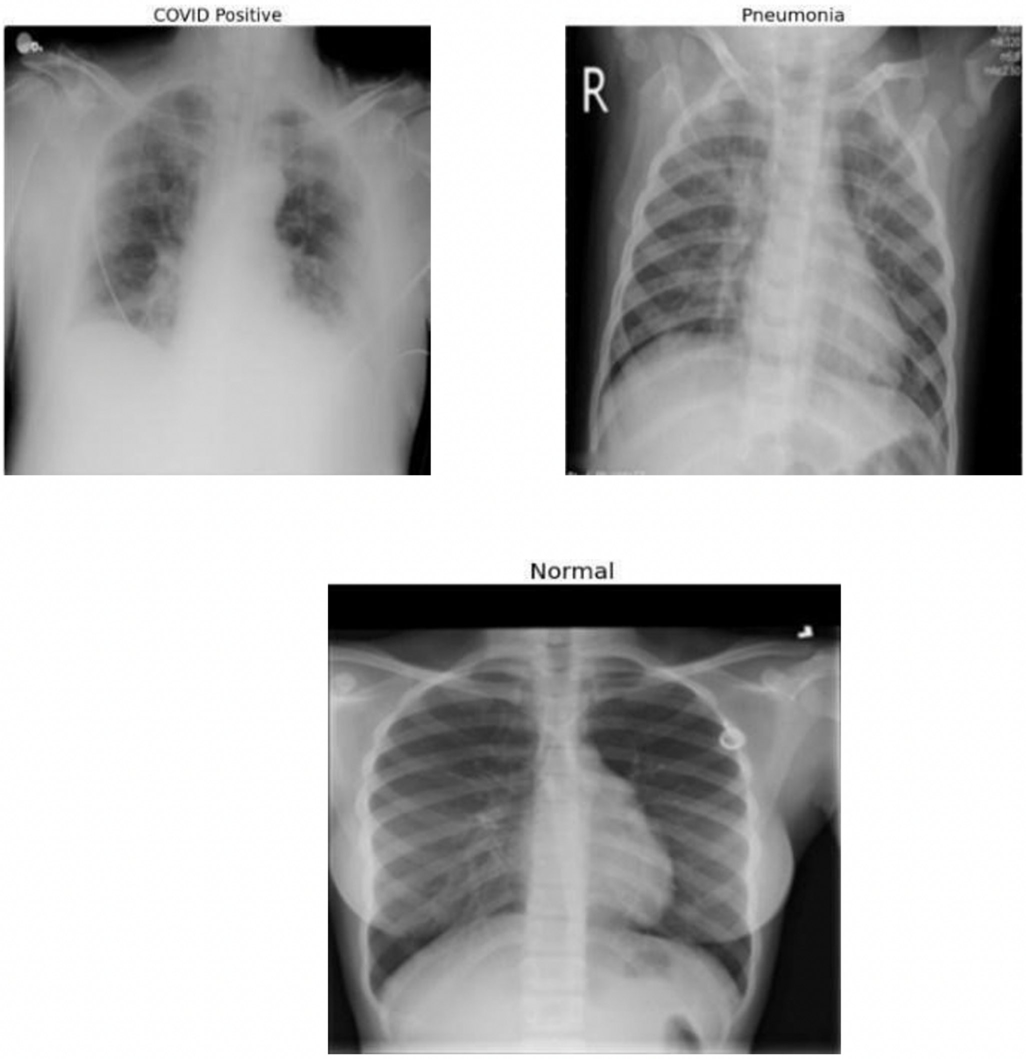


Figure 4.2: Covid Positive, Normal, and Viral Pneumonia Sample Images

In figure 4.2, Sample images of our dataset is shown. We will extract features from these images that will allow us to determine which category the image belongs to.To extract the features, feature extraction is utilized.

4.2 DATA PRE-PROCESSING

• In data preprocessing, "Image Data Generator" helps in increasing the efficiency of the dataset. In a picture Data generator, our images are augmented easily in terms of picture brightness, range of zoom, horizontal flip,angle shifting, shear range, etc.

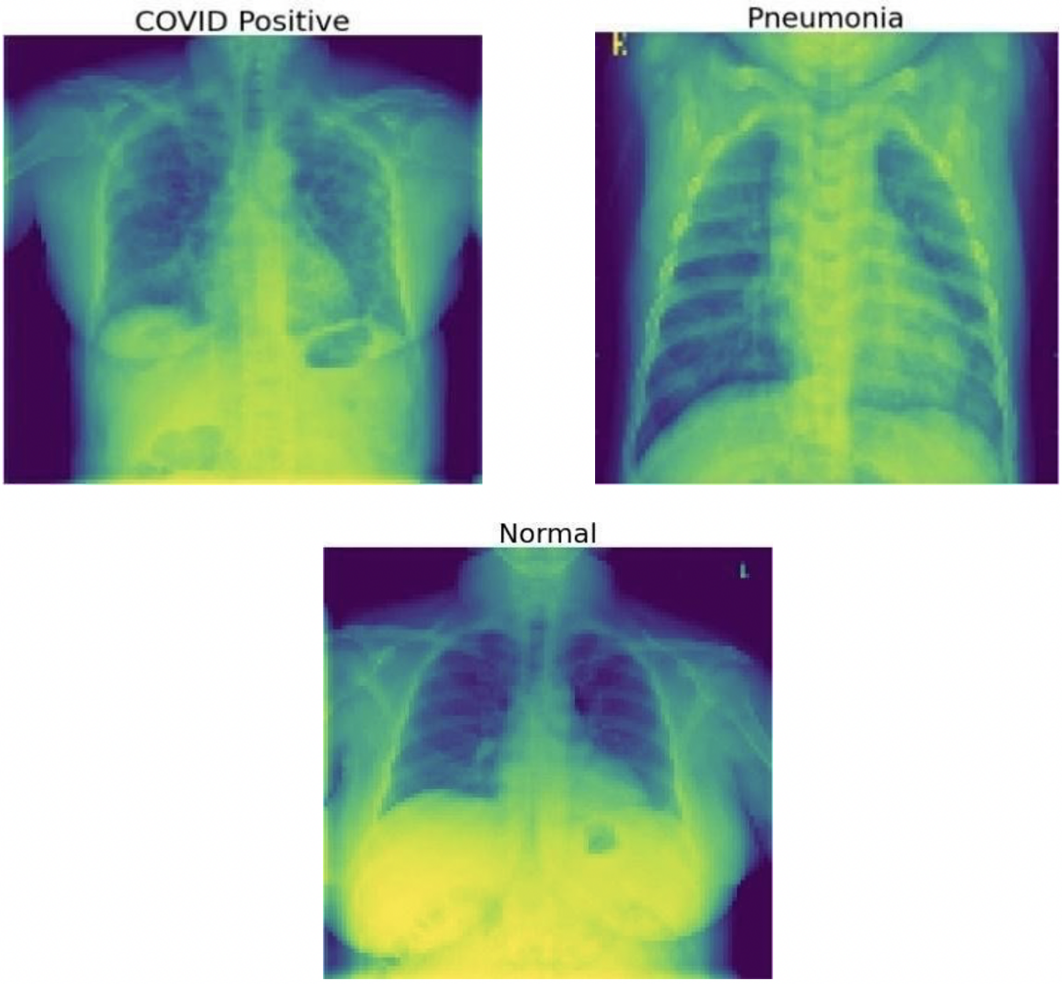


Figure 4.3: Pre-Processed images of Covid Positive, Normal and Pneumonia

• We have used gray scaling because it simplifies the algorithm and reduces computational requirements.

• As the images within the dataset include different sizes. We resized the grayscale to 100*100, so we are going to get the identical set of sizes for all the images within the dataset. Since absolute feature extraction is feasible in gray scaling.

• The formula for Gray Scaling pictures is as follows:

• GrayScale = (R x 0.3) + (G x 0.59) + (B x 0.11)

• R denotes red, G is green, and B denotes blue.

Scaling the Data: Before modeling it's a good advantage/benefit to prepare image pixel. Scaling all pixel values to 0 to 1 by just divding pixel with 255. So that our model works very fast.

In figure 4.3, it displays pre-processed pictures from our dataset's multiple classifications. The photos are brightened, grayscaled, and downscaled to the range 0-1 so that our model can work quickly.

4.3 CONVOLUTIONAL NEURAL NETWORK(CNN)

CNN's are neural networks that are designed to handle statistic data and multidimensional data such as picture. Feature extraction and calculation of weights in the process of training are simple examples of this. A convolution operator is used to determine the identification of such networks, which is useful for tackling complicated jobs.CNN is an effective image enhancement. These techniques are presently the strongest researchers have towards mechanized image analysis. Several organizations implement these techniques to do tasks such as detecting items in pictures.

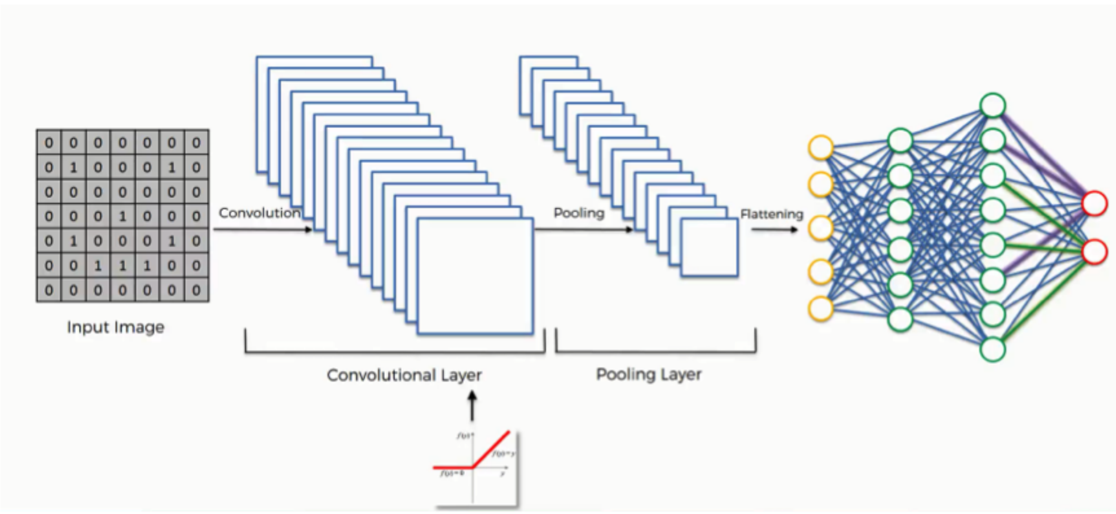


Figure 4.4: CNN work

In figure 4.4: it explains how the CNN model works. The input picture is routed through convolutional layers, pooling layers (max pooling), and flattening layers before being given the prediction.

4.4 FEATURE EXTRACTION

The main feature of Convolutional Neural Networks (CNNs) is the automatic feature extraction function. Throughout most cases, the determined source information is delivered to the Extraction Network function, which then passes the extracted features on to the Classifier Network. Various convolutional and pooling layer pairs are used during the feature extraction procedure. A collection of convolution filters is passed through the input data to perform the convolution process. The threshold is defined by the pooling layer, which acts as a dimensional reduction layer. Some dimensions must be changed during backpropagation, resulting in fewer interactions within the framework of the neural network.

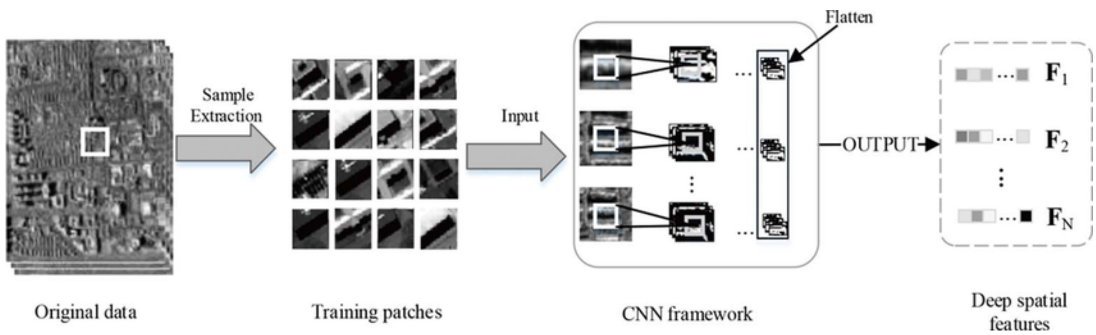


Figure 4.5: Feature extraction work flow in CNN

In figure 4.5, the feature extraction in the CNN model is well demonstrated. The picture is first divided into patches as shown in the training patches section, after which features are extracted using an inuilt procedure in the CNN framework and output as profound spatial objects.

4.5 SUMMARY OF CNN MODEL

To the recommended CNN model, we develop a generative model with a concurrent fully connected layer. To the input layer, a parallel 2-Dimensional convolutional layer is added with different filters of 32, 64, 128 to the different kernels of sizes 3, 5, and 7. We then reduced spatial dimensions using a Max-Pooling layer with (2,2) as the size of the pool. To get the same output size we combined the conv2D layer into a single layer with the same padding. After that, we included convolutional layers with various filters and utilized Relu as an activation function, which returns just '0' when given negative inputs and the same value when given positive inputs. The convolutional layers' output is reduced to a 1D array and passed on to the 3 dense layers in which the first two have activation function as 'relu' and the last one has 'softmax' as activation function so that it classifies the object/image on the basis of the output received from the convolutional layers. To avoid the training data from becoming overfitted, we have included three dropout layers with a 0.5 dropout rate each. we utilized the loss function as 'Categorical Cross Entropy' and 'adam' was used as an optimizer.

The output layer is going to be of three neurons for the Covid positive, Normal, and Pneumonia. When the model completed its training, the one which has higher accuracy and smallest amount of loss is our best model and its saved using save_best_only=True and the file is saved with '.h5' extension. This will be used for further implementation for web application.

In figure 4.6, it explains how our CNN model will function inside. The layers we added to our CNN model are depicted in this figure. And all the layers, activtion fucntion, optimizer, and values used in it.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 model (Functional)          (None, 100, 100, 384)     11008

 conv2d_3 (Conv2D)           (None, 98, 98, 64)        221248

 activation (Activation)     (None, 98, 98, 64)        0

 max_pooling2d (MaxPooling2D  (None, 49, 49, 64)       0
 )

 conv2d_4 (Conv2D)           (None, 47, 47, 32)        18464

 activation_1 (Activation)   (None, 47, 47, 32)        0

 max_pooling2d_1 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 flatten (Flatten)           (None, 16928)             0

 dropout (Dropout)           (None, 16928)             0

 dense (Dense)               (None, 128)               2166912

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 3)                 195

=================================================================
Total params: 2,426,083
Trainable params: 2,426,083
Non-trainable params: 0
```

Figure 4.6: Summary of CNN Model

CHAPTER 5

CODING AND TESTING

We divided our coding part into 7 parts:

1. Loading Data
2. Importing Required Libraries
3. Displaying Dataset Images
4. Data Pre-Processing
5. Model Building
6. Visualizing Graphs
7. Testing Model

```
from google.colab import drive

drive.mount('/content/gdrive')

!unzip /content/gdrive/MyDrive/project/Project.zip
```

Figure 5.1: Loading Data code snippet

In figure 5.1, demonstrates how the data is loaded We saved our dataset in Google Drive and imported it from Google Colab, then unzipped and explored it with !unzip.

```
import os
import cv2
import numpy as np
import pandas as pd
from keras.utils import np_utils
import seaborn as sns
from matplotlib import pyplot as plt
from google.colab.patches import cv2_imshow
import tensorflow as tf
from tensorflow.keras.models import Sequential,Model
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D,Activation,MaxPooling2D
from tensorflow.keras.utils import normalize
from tensorflow.keras.layers import Concatenate, GlobalAveragePooling2D
from tensorflow.keras import Input
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve, auc
```

Figure 5.2: Importing Libraries code snippet

In figure 5.2, explains how to import libraries from various Python packages. Libraries are pre-installed Python features that aid in the training of ML and DL models. Python, Scikit-Learn, and Tensorflow were utilized. We utilized the matplotlib and seaborn libraries from the Python package for visualization. We utilized Sklearn packages for model testing, while Tensorflow is a Python framework that provides a collection of procedures for developing and training models in Python. All of these libraries will simplify our job and make our code more computationally efficient.

```python
fig = plt.figure(figsize=(30, 21))
rows = 1
columns = 3
Image1 = cv2.imread('/content/Project/dataset/COVID Negative/Normal-1.png')
Image2 = cv2.imread('/content/Project/dataset/COVID Positive/COVID-1.png')
Image3 = cv2.imread('/content/Project/dataset/Pneumonia/Viral Pneumonia-1.png')


# Adds a subplot at the 1st position
fig.add_subplot(rows, columns, 1)

# showing image for covid negative
plt.imshow(Image1)
plt.axis('off')
plt.title("COVID Negative", fontsize = 20)

# Adds a subplot at the 2nd position
fig.add_subplot(rows, columns, 2)

# showing image for covid positive
plt.imshow(Image2)
plt.axis('off')
plt.title("COVID Positive", fontsize=20)

# Adds a subplot at the 3rd position
fig.add_subplot(rows, columns, 3)

# showing image for pneumonia
plt.imshow(Image3)
plt.axis('off')
plt.title("Pneumonia", fontsize=20)
```

Figure 5.3: Displaying Dataset Images code snippet

In figure 5.3, the code snippet for displaying our dataset image is displayed. We read the dataset picture using the OpenCV library and show/display them using the OpenCV library imshow. So that we can view how the CXR pictures in our dataset appear.

In Below figure 5.4, demonstrates a code snippet for our dataset's Data Pre-Processing.Data pre-processing is utilised to speed up our model and make it easier to categorise. We resized our photos and converted our dataset images into grey scale images.

```
img_size=100
data=[]
target=[]

fig = plt.figure(figsize=(30, 21))
rows = 1
columns = 3

for category in categories:
    if(category == "Pneumonia"):
      flag_pn = 1
    if(category == "COVID Positive"):
      flag_po = 1
    if(category == "COVID Negative"):
      flag_ne = 1

    folder_path=os.path.join(data_path,category)
    img_names=os.listdir(folder_path)


    for img_name in img_names:
        img_path=os.path.join(folder_path,img_name)
        img=cv2.imread(img_path)
        try:
            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
            #Coverting the image into gray scale / Gray scaling
            resized=cv2.resize(gray,(img_size,img_size))
```

```
  if(flag_ne):
    fig.add_subplot(rows, columns, 1)
    plt.imshow(resized)
    plt.axis('off')
    plt.title("COVID Negative", fontsize = 30)
    flag_ne=0

  if(flag_po):
    fig.add_subplot(rows, columns, 2)
    plt.imshow(resized)
    plt.axis('off')
    plt.title("COVID Positive", fontsize = 30)
    flag_po=0

  if(flag_pn):
    fig.add_subplot(rows, columns, 3)
    plt.imshow(resized)
    plt.axis('off')
    plt.title("Pneumonia", fontsize = 30)
    flag_pn=0

  data.append(resized)
  target.append(label_dict[category])
```

Figure 5.4: Data Pre-Processing-1 code snippet

```
# Converting pixels(0 to 255) into 0 to 1 (Scaling the data)
data=np.array(data)/255.0
data=np.reshape(data,(data.shape[0],img_size,img_size,1))
target=np.array(target)

new_target=np_utils.to_categorical(target)
```

```
np.save('data',data)
np.save('target',new_target)
```

Figure 5.5: Data Pre-Processing-2 code snippet

In figure 5.5, displays a fragment of Data Scaling Pre-Processing code for our dataset. We reduce the picture size from 255px to 0-1px. So that our model can quickly assess it and decrease computational efforts.

```
for k in range(len(parrallel_kernels)):
    conv = Conv2D(128, parrallel_kernels[k],padding='same',activation='relu',input_shape=input_shape,strides=1)(inp)
    convs.append(conv)

out = Concatenate()(convs)
conv_model = Model(inp,out)

model = Sequential()
model.add(conv_model)

model.add(Conv2D(64,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(32,(3,3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(3,input_dim=128,activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])

model.summary()
```

Figure 5.6: CNN Model code snippet

In figure 5.6, exhibits the CNN model code. 2D convo layers, max-pooling layers, dense layer, dropout, and activation functions such as softmax and relu, and adam optimizer were utilized in this model. In the Methodology section, the model's summary is briefly discussed.

```python
# Training the model
train_data,test_data,train_target,test_target=train_test_split(data,target,test_size=0.1,random_state=0)

checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')
history=model.fit(train_data,train_target,epochs=20,validation_split=0.1)
```

Figure 5.7: Training the model code snippet

In figure 5.7, shows the code for our model's Training. To avoid data leaking, we first separated our data into train and test sections. 90% of the dataset is used to train the model, while 10% is used to evaluate the model. We then maintain several checkpoints in place, such as monitoring validation loss, storing the best model by setting save best only to True and setting the random state to zero. The fit approach is then used to train our model. Our best accuracy given epoch will be saved after this operation is done.

```python
# visualizing the Training and validation accuracy

plt.plot(history.history['loss'],'r',label='training loss')
plt.plot(history.history['val_loss'],label='validation loss')
plt.title("Training Vs Validation Loss", fontsize = 20, color = "blue")
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()

plt.plot(history.history['accuracy'],'r',label='training accuracy')
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.title("Training Vs Validation Accuracy",fontsize = 20, color = "blue")
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

Figure 5.8: Visualizing Training and Validation accuracy code snippet

The above figure 5.8, the code for visualizing training and validation accuracy graphs is demonstrated. We may learn about the training and validation loss, as well as the training and validation accuracy of our model, by utilizing graphing and visualization modules in Python. Then we can figure out which epoch is producing the best results for our model.

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

rounded_predictions = model.predict(test_data, batch_size=128, verbose=0)
rounded_labels = np.argmax(rounded_predictions ,axis =1)
test_target_1 = np.argmax(test_target, axis=1)
print('Confusion Matrix')
confusion_matrix = confusion_matrix(test_target_1, rounded_labels)
print(confusion_matrix)
print('Classification Report')
target_names = ['Pneumonia','Covid', 'Normal']
print(classification_report(test_target_1, rounded_labels, target_names=target_names))

import seaborn as sns
sns.heatmap(confusion_matrix,annot=True,xticklabels=target_names, yticklabels=target_names,cmap='Blues',fmt="d")
```

Figure 5.9: Testing-1 Model code snippet

In figure 5.9, our CNN model's testing code is provided. It is the major section of the code that determines how well our model works and where it falls short. We used the popular Scikit-learn package to load libraries like classification report, accuracy score, and confusion matrix, which helped us evaluate our model.

```
Y_pred_proba = []
for i in range(len(rounded_predictions)):
    v = rounded_predictions[i]/rounded_predictions[i].sum()
    Y_pred_proba.append(v)

auc = roc_auc_score(test_target_1, Y_pred_proba, multi_class='ovr', average='weighted')
print(auc)

# roc curve for classes
fpr = {}
tpr = {}
thresh ={}

n_class = 3
for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(test_target_1, rounded_predictions[:,i], pos_label=i)

# plotting
plt.plot(fpr[0], tpr[0], linestyle='--',color='red', label='Covid vs Rest')
plt.plot(fpr[1], tpr[1], linestyle='--',color='green', label='Normal vs Rest')
plt.plot(fpr[2], tpr[2], linestyle='--',color='blue', label='Pneumonia vs Rest')
plt.text(1, 1.2, 'Auc Score = 99.5%', fontsize = 20)
plt.title('Multiclass ROC curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive rate')
plt.legend(loc='best')
plt.savefig('Multiclass ROC',dpi=300);
```

Figure 5.10: Testing-2 Model code snippet

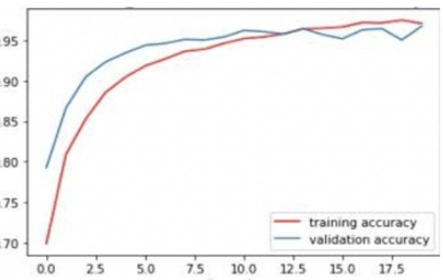
In figure 5.10, illustrates the code for testing part of our model. The roc-auc score module from the Scikit-learn package was utilized here. It aids in the evaluation of our algorithms, particularly for imbalanced datasets and multiclass classification tasks. It indicates how much data is contained inside the roc curve; the sharper the curve, the more accurate the model. As a result, the more data in the roc curve, the more accurate the model is.

CHAPTER 6

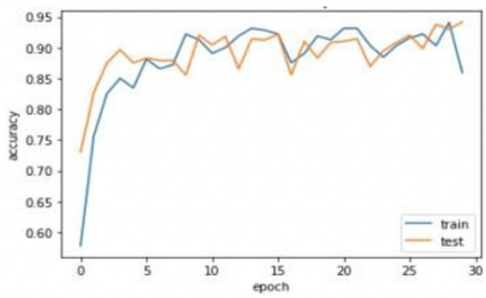
RESULTS AND OBSERVATIONS

Our model's main purpose is to create a powerful DL model that can classify and forecast whether a disease is Covid Positive, Normal, or Pneumonia with more accuracy and better performance metrics. The photos in the collection are divided into three categories: 3616 Covid Positive, 10192 Normal, and 1345 Pneumonia images of a chest x-ray. For multiclassification of these three classes, we used seven different CNN models. Where most of them did well and a handful of them did not. Our model's major goal is to distinguish between Covid positive, Normal, and Pneumonia. On comparing our CNN model with our pre-trained base models, the CNN model performed better.

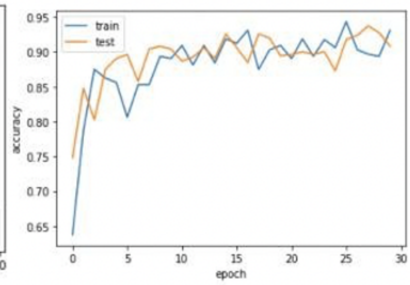
In figure 6.1, With epoch graphs, the photographs demonstrate the correctness of the pre-trained RESNET50, VGG-16, M0BILENET, DENSENET169, XCEPTION, and EFFICIENTB0 models. This clearly reveals that our model CNN outperforms the other six models in terms of accuracy rate.



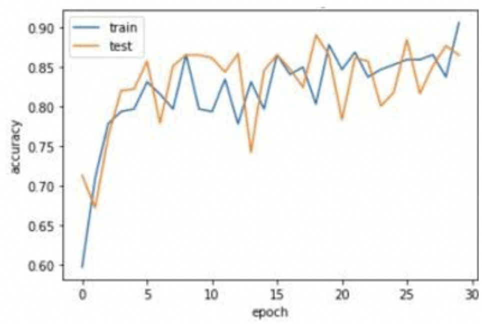
CNN Model



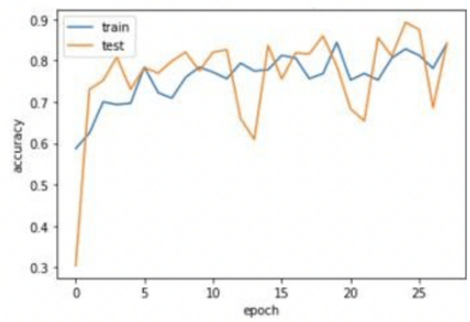
RESNET50



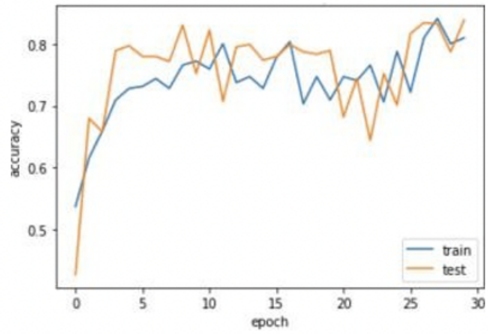
VGG 16



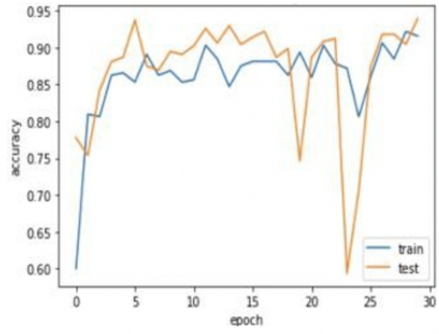
MOBILENET



DENSENET 169



XCEPTION



EFFICIENTNET

Figure 6.1: Accuracy vs Epoch graph of different models

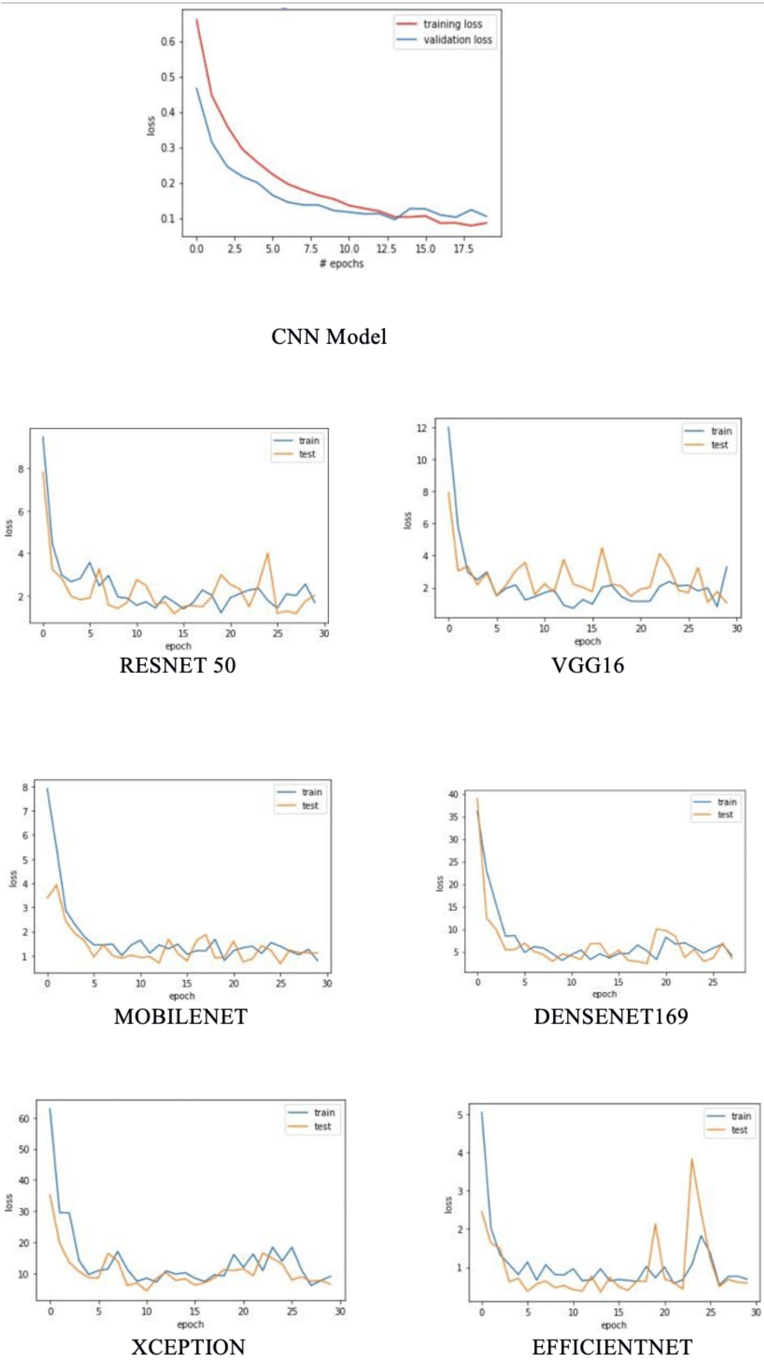


Figure 6.2: Loss vs Epoch graph of different models

In figure 6.2, The photos depict the RESNET50, VGG-16, MobileNet, DENSENET169, XCEPTION, and EFFICIENTB0 models' losses to the epoch graph. The loss graph of our suggested CNN model is shown at the top. This shows that our proposed CNN model has a lesser loss rate than that of the other six models which were pre-trained, which have values of 98%, 97%, and 97% for Normal, and precision, recall, and F1 scores of 91%, 98%, and 94% for Pneumonia, respectively, and an AUC score of 99.5%.

6.1 Classification Reports

- **Precision :-** It's employed in pattern recognition and information retrieval. Precision is the proportion of positive points that are genuinely positive out of all the positive points claimed.
- **Recall :-** Is not about how many points are genuinely positive, but about what proportion is deemed positive.
- **F1-score :-** It is used to determine the correctness of tests. The accuracy and recall are weighted averaged. The best F1 score is 1 and the poorest is 0.
- **AUC-ROC :-** It is one of the key measurements for assessing our demonstration. Since our information is imbalanced information, we ought to not consider exactness alone for assessing our demonstration. ROC is compared between True positive Rate(TPR), and Wrong Positive Rate, the more the bend towards the TPR, the more exact is our demonstration. AUC gives how much information is displayed interior it, more the more information interior the ROC, the more precise is our demonstrate.
- **Confusion Matrix :-** It tells us how our data is categorized correctly or incorrectly.

Table 6.1: Classification report of CNN model

	Precision	Recall	F1-Score
Covid	0.95	0.95	0.95
Normal	0.98	0.97	0.97
Pneumonia	0.91	0.98	0.94
Accuracy	0.967		
AUC	0.995		

In Table 6.1, achievement of model is exceptional. With recall, F1 scores and precision of 95 %, 95 %, and 95 % for Covid Positive, 97 %, 97 % and 98 % for Normal, 94 %, 91 %, and 98 % for Pneumonia, and an AUC score of 99.5 % for correctly categorizing images. And the accuracy of 96.5%

Table 6.2: Classification report of Resnet50 model

	Precision	Recall	F1-Score
Covid	0.79	0.92	0.85
Normal	0.95	0.91	0.93
Pneumonia	0.98	0.82	0.90
Accuracy	0.942		
AUC	0.926		

In Table 6.2, achievement of model is good. With recall, F1 scores, and precision of 92 %, 85 %, and 79 % for Covid Positive the precision is not that good, 91 %, 93 % and 95 % for Normal, 82 %, 90 %, and 98 % for Pneumonia the recall is not that good, and an AUC score of 92.6 % for correctly categorizing images. And the accuracy of 94.2%

Table 6.3: Classification report of VGG 16 model

	Precision	Recall	F1-Score
Covid	0.86	0.85	0.86
Normal	0.92	0.95	0.94
Pneumonia	1.00	0.78	0.88
Accuracy	0.939		
AUC	0.901		

In Table 6.3, achievement of model is great. With recall, F1 scores, and precision of 85 %, 86 %, and 86 % for Covid Positive the metrics were not that good, 95 %, 94 % and 92 % for Normal, 78 %, 88 %, and 100 % for Pneumonia the recall isn't great, but the precision is fantastic., and an AUC score of 90.1 % for accurately identifying images. And the accuracy of 93.9%.

Table 6.4: Classification report of MOBILENET model

	Precision	Recall	F1-Score
Covid	0.92	0.69	0.79
Normal	0.89	0.98	0.93
Pneumonia	0.93	0.81	0.87
Accuracy	0.896		
AUC	0.957		

In Table 6.4, achievement of model is not that amazing. With recall, F1 scores, and precision of 69 %, 79 %, and 92 % for Covid Positive the metrics were not that good, 98 %, 93 % and 89 % for Normal, 81 %, 87 %, and 93 % for Pneumonia the recall is not that good, and an AUC score of 95.7 % for truthfully labelling pictures. And the accuracy of 89.6%.

Table 6.5: Classification report of DENSENET 169 model

	Precision	Recall	F1-Score
Covid	0.56	0.86	0.68
Normal	0.93	0.76	0.84
Pneumonia	0.97	0.88	0.92
Accuracy	0.814		
AUC	0.630		

In Table 6.5, achievement of model is poor. With recall, F1 scores, and precision of 86 %, 68 %, and 56 % for Covid Positive the numbers were terrible, 76 %, 84 % and 93 % for Normal the recall is not good, 88 %, 92 %, and 97 % for Pneumonia the metrics were good, and an AUC score of 63 % indicating that most of the images are falsely labeled. And the accuracy of 81.4%.

Table 6.6: Classification report of XCEPTION model

	Precision	Recall	F1-Score
Covid	0.35	0.91	0.51
Normal	0.75	0.41	0.53
Pneumonia	0.90	0.13	0.23
Accuracy	0.846		
AUC	0.680		

In Table 6.6, achievement of model is disappointing. With recall, F1 scores, and precision of 91 %, 51 %, and 35 % for Covid Positive the numbers were terrible, 41 %, 53 % and 75 % for Normal the values were horrible, 13 %, 23 %, and 90 % for Pneumonia the metrics were too bad, and an AUC score of 68 % % inferring that most of the pictures are incorrectly identified. And the accuracy of 84.6%.

Table 6.7: Classification report of EFFICIENTB0 model

	Precision	Recall	F1-Score
Covid	0.88	0.88	0.88
Normal	0.92	0.96	0.94
Pneumonia	1.00	0.69	0.82
Accuracy	0.917		
AUC	0.968		

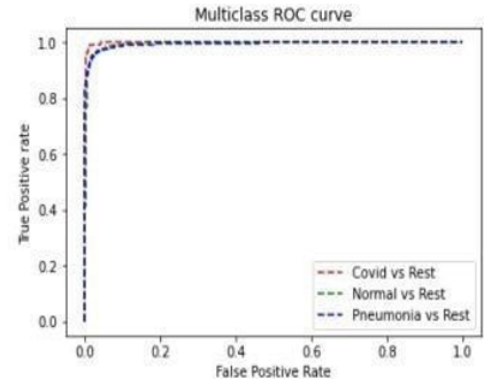
In Table 6.7, achievement of model is satisfactory. With recall, F1 scores, and precision of 88 %, 88 %, and 88 % for Covid Positive the numbers were good, 96 %, 94 % and 92 % for Normal the values were excellent, 69 %, 82 %, and 100 % for Pneumonia the precision is more accurate, and an AUC score of 96.8 % % trying to show that many of the images are properly identified. And the accuracy of 91.7%.

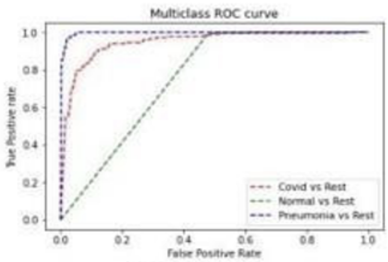
6.2 ROC CURVE

The Receiver Operating Curve (ROC) is used to calculate the performance of a classification model at all threshold levels with the use of two parameters: True Positive Rate (TPR) and False Positive Rate (FPR).

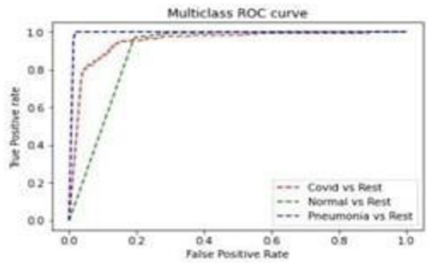
The curve must have sharp edges rather than flat edges, showing that it is approaching TPR and that the model is really good.

In below figure 6.3, The closer the curve is to the TPR in ROC space, the better the model's performance. The closer the curve gets to the FPR of ROC space, the less accurate it is. The AUC is a metric that estimates the total 2 Dimensional area beneath the ROC curve. In our case, the AUC score is more for our proposed CNN model (99.5%).

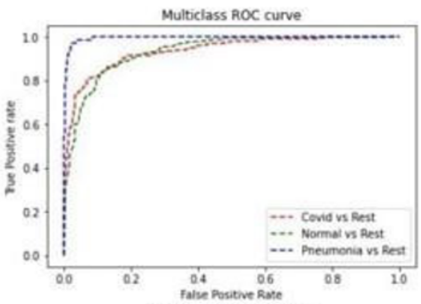




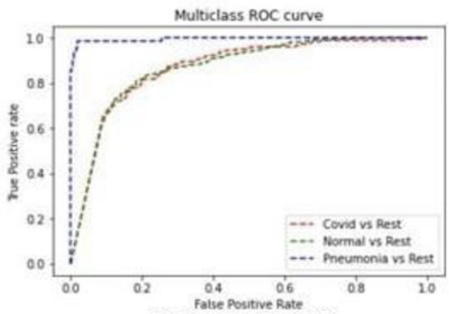
RESNET50



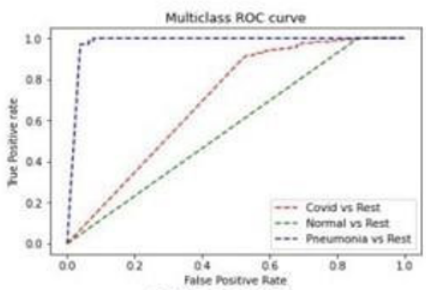
VGG16



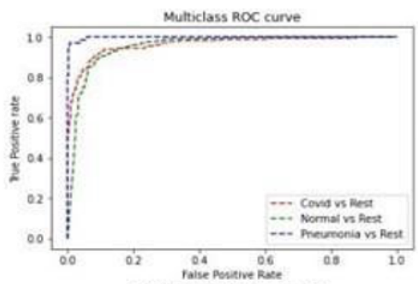
MOBILENET



DENSENET169



XCEPTION



EFFICIENTNET

Figure 6.3: ROC vs Epoch graph of different models

6.3 Confusion matrix

For imbalanced datasets and multiclass classifications, the Confusion Matrix has been the most essential evaluator. We evaluate the model's performance by comparing Actual and Predicted Values.

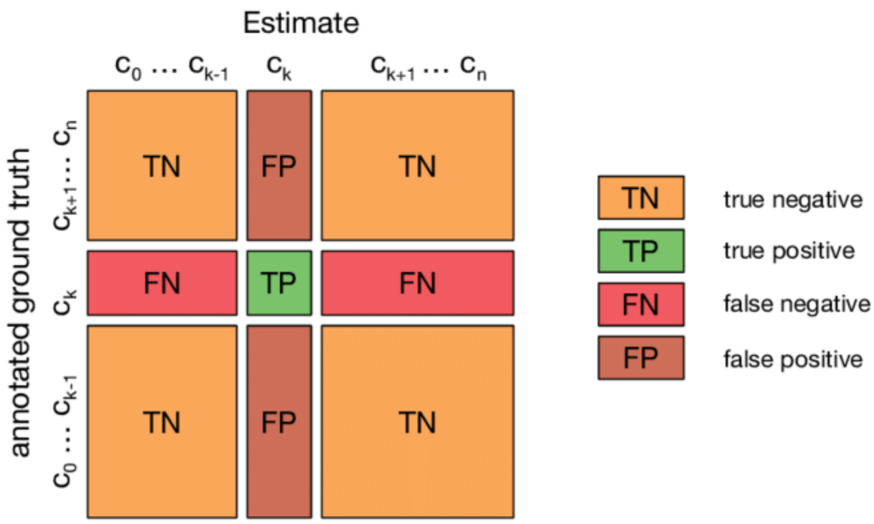


Figure 6.4: Confusion Matrix for mutliclass

In figure 6.4, illustrates how to simplify, and understand a confusion matrix. True Positive (TP), False Positive (FP), False Positive (FP), and False Negative (FN) are used to examine the confusion matrix (FN). False Positive and False Negative are low in a great model. Accuracy, Precision, recall, and the F1-score are all evaluated using this method. The higher the TP and TN numbers, the better our model is. The model is worse when the FP and FN values are higher.

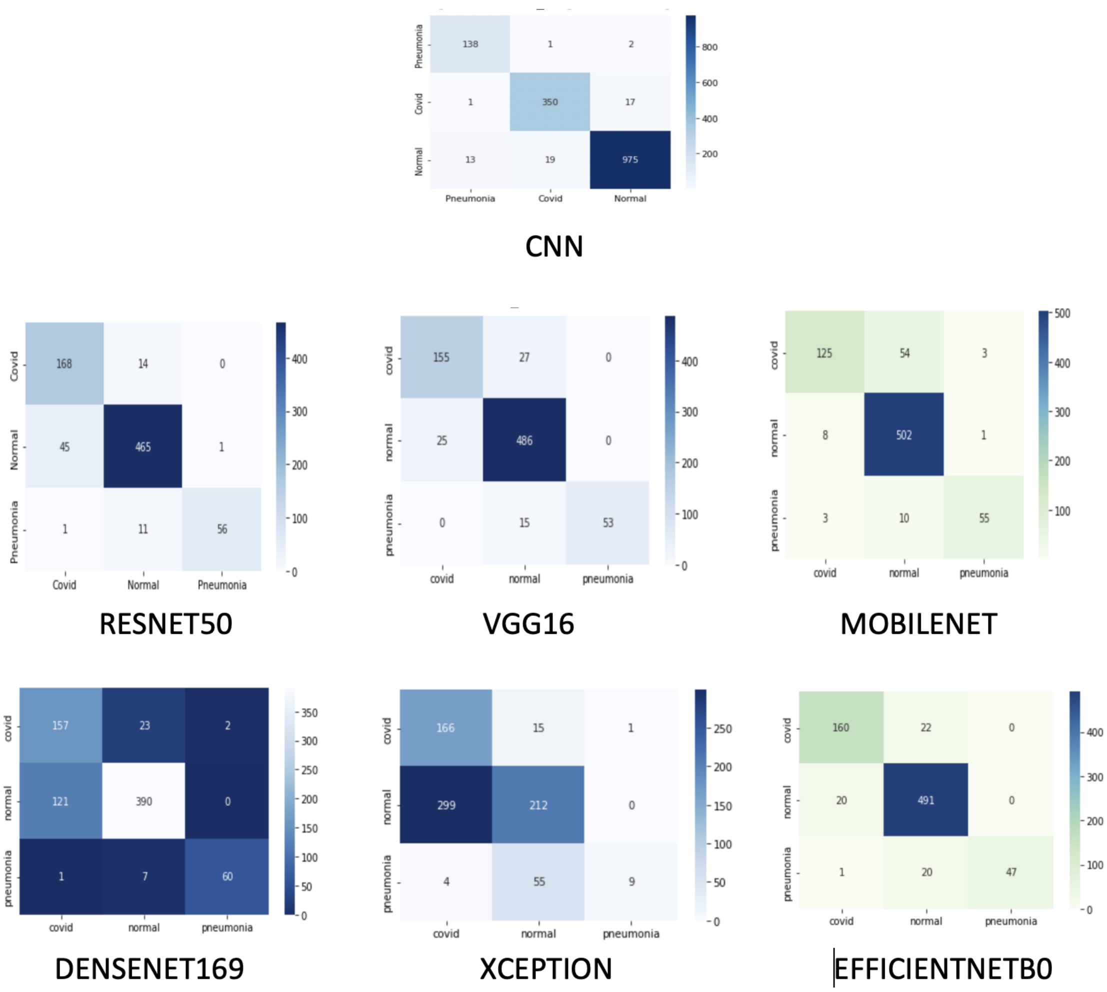


Figure 6.5: Confusion Matrix of different models

In figure 6.5, confusion matrices for several models have been shown. In comparison, all models performed admirably. However, when compared to all other models. The ResNet50 model and CNN model produces fewer number of incorrect predictions, this implies that the model's performance is satisfactory. And higher number of inaccurate predictions were produced by Xception model and Densenet169 model, This signifies that the model's effectiveness is poor.

CHAPTER 7

CONCLUSION

The main problem is the spreading of this infection, this is due to not knowing exactly whether someone is stricken by covid or other diseases whose symptoms are similar to covid like pneumonia, etc. There are many tests for knowing whether an individual has been infected with covid or not with the assistance of the tests like Rapid Antigen Test, antibodies test, and RT-PCR test which are good enough but their chances of this test can give False-Negative report, and also cannot classify between covid and other diseases.

Due to the similar symptoms of Covid positive and Pneumonia might be confused, an accurate diagnosis is difficult to come by.In order to address these obstacles, a Deep Learning model was constructed to forecast and categorise the situation.

We were able to correctly diagnose the condition as either Covid Positive, Pneumonia, or Covid Negative. Covid Positive has a greater accuracy rate and a lower loss rate than Pneumonia. with a short duration In order to diagnose chest x-ray images in a short amount of time and for other purposes. This model is utilised with precision. In rural places, primary health care workers may make a difference. This automatic Covid and pneumonia detection device is being used. This proposed project is improved by adding more Covid X-rays and pneumonia chest x-rays to the dataset and x-rays of the chest for various lung illnesses. It can also be used for IoT-related activities by connecting them to the setup for predicting COVID-19 and pneumonia in real-time.

REFERENCES

[1] Das, A., Ghosh, S., Thunder, S., Dutta, R., Agarwal, S., and Chakrabarti, A., 2021, "Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network," Pattern Analysis and Applications, **24**.

[2] Mporas, I. and Naronglerdrit, P., 2020, "COVID-19 Identification from Chest X-Rays," pp. 69–72, doi:10.1109/BIA50171.2020.9244509.

[3] Sekeroglu, B. and Ozsahin, I., 2020, "Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks," SLAS technology, **25**, p. 2472630320958376.

[4] Karim, M., Dohmen, T., Cochez, M., Beyan, O., Rebholz-Schuhman, D., and Decker, S., 2020, "DeepCOVIDExplainer: Explainable COVID-19 Diagnosis from Chest X-ray Images," pp. 1034–1037, doi:10.1109/BIBM49941.2020.9313304.

[5] Arias, D., Grisales, J., Orozco Arias, S., Arteaga Arteaga, H., Ortíz, M., Rubio, A., Saborit-Torres, J., Serrano, J., Tabares Soto, R., and de la Iglesia Vaya, M., 2021, "COVID-19 detection in X-ray images using convolutional neural networks," Machine Learning with Applications, **6**, p. 100138.

[6] Goswami, D., Law, A., Chakraborty, D., and Dey, A., 2020, "Convolutional neural network for prediction of COVID-19 from chest X-ray images," doi:10.1145/3429210.3429219.

[7] Dasari, H., Praneeth, C., and Pranathi, M., 2020, "Covid Prediction from X-ray Images," pp. 1–5, doi:10.1109/ICCCS49678.2020.9276795.

[8] El-Dahshan, E.-S., Bassiouni, M., Hagag, A., Chakrabortty, R., Loh, H., and Acharya, U. R., 2022, "RESCOVIDTCNnet: A residual neural network-based framework for COVID-19 detection using TCN and EWT with chest X-ray images," Expert Systems with Applications, p. 117410.

[9] Hamza Osman, A., Aljahdali, H., Tarazi, S., and Ahmed, A., 2021, "SOM-LWL method for identification of COVID-19 on chest X-rays," PLOS ONE, **16**, p. e0247176.

[10] Kuamr, R., Arora, R., Bansal, V., Joseph, V., Buckchash, H., Imran, J., Narayanan, N., Pandian, G., and Raman, B., 2022, "Classification of COVID-19 from chest x-ray images using deep features and correlation coefficient," Multimedia Tools and Applications, pp. 1–25.

[11] Erdem, E. and Aydın, T., 2020, "COVID-19 detection in Chest X-ray Images using Deep Learning," doi:10.21203/rs.3.rs-65954/v1.

[12] Pillalamarry, M., Prathyusha, Y., Sahithi, B., and Nagendram, S., 2021, "Covid-19 Detection from Chest X-Ray using Convolution Neural Networks," Journal of Physics: Conference Series, **1804**, p. 012197.

[13] Ozturk, T., Talo, M., Yildirim, A., Baloglu, U., Yildirim, , and Acharya, U. R., 2020, "Automated Detection of COVID-19 Cases Using Deep Neural Networks with X-ray Images," Computers in Biology and Medicine, **121**.

[14] Wang, L., Lin, Z., and 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**.

[15] KARHAN, Z. and Akal, F., 2020, "Covid-19 Classification Using Deep Learning in Chest X-Ray Images," pp. 1–4, doi:10.1109/TIPTEKNO50054.2020.9299315.

[16] Varshney, Y., Krishna, A., Nagrath, P., and Jain, R., 2021, *Detection of COVID-19 by X-rays Using Machine Learning and Deep Learning Models*, pp. 409–421.

[17] Al-Asfoor, M. and Abed, M., 2022, *Deep Learning Approach for COVID-19 Diagnosis Using X-Ray Images*, pp. 161–170.

[18] Taneja, S., Gupta, M., and Jain, R., 2020, "CNN Based Analysis of COVID-19 Using Chest X-Ray Images," International Journal of Innovative Research in Applied Sciences and Engineering, **4**, pp. 644–651.

[19] Kusakunniran, W., Borwarnginn, P., Sutassananon, K., Tongdee, T., Saiviroonporn, P., Karnjanapreechakorn, S., and Siriapisith, T., 2021, "COVID-19 detection and heatmap generation in chest x-ray images," Journal of Medical Imaging, **8**.

[20] Ohata, E. F., Bezerra, G. M., Chagas, J. V. S. d., Lira Neto, A. V., Albuquerque, A. B., Albuquerque, V. H. C. d., and Reboucas Filho, P. P., 2021, "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning," IEEE/CAA Journal of Automatica Sinica, **8**(1), pp. 239–248.

[21] Madaan, V., Roy, A., Gupta, C., Agrawal, P., Sharma, A., Bologa, C., and Prodan, R., 2021, "XCOVNet: Chest X-ray Image Classification for COVID-19 Early Detection Using Convolutional Neural Networks," New Generation Computing, **39**.

[22] Moustafa, H. E.-D. and El-Seddek, M., 2020, "Accurate Diagnosis of COVID-19 Based on Deep Neural Networks and Chest X-Ray Images," Bulletin of the Faculty of Engineering. Mansoura University.

[23] Ahsan, M., Based, M., Haider, J., and Kowalski, M., 2021, "COVID-19 Detection from Chest X-ray Images Using Feature Fusion and Deep Learning," Sensors, **21**.

[24] Sanket, S., Sarobin, M., Anbarasi, L., Thakor, J., Singh, U., and Narayanan, S., 2021, "Detection of novel coronavirus from chest X-rays using deep convolutional neural networks," Multimedia Tools and Applications.

[25] Ahmed, S., Hossain, T., Hoque, O., Sarker, S., Rahman, S., and Shah, F., 2021,

"Automated COVID-19 Detection from Chest X-Ray Images: A High-Resolution Network (HRNet) Approach," SN Computer Science, **2**.

[26] Alghamdi, H., Amoudi, G., Elhag, S., Saeedi, K., and Nasser, J., 2021, "Deep Learning Approaches for Detecting COVID-19 from Chest X-ray Images: A Survey," IEEE Access, **PP**, pp. 1–1.

[27] Akgül, , Kaya, V., Ünver, E., Karavaş, E., Baran, A., and Tuncer, S., 2022, "Covid-19 detection on x-ray images using a deep learning architecture," Journal of Engineering Research.

[28] Asif, S., Yi, W., Jin, H., Tao, Y., and Jinhai, S., 2020, "Classification of COVID-19 from Chest X-ray images using Deep Convolutional Neural Networks," doi: 10.1101/2020.05.01.20088211.

[29] Mohammad, S. and Masud, M., 2020, "On the Detection of COVID-19 from Chest X-Ray Images Using CNN-based Transfer Learning," Computers, Materials Continua, **64**, pp. 1359–1381.

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Office of Controller of Examinations

REPORT FOR PLAGIARISM CHECK ON THE PROJECT REPORTS

1	Name of the Candidate	**SHAIK FIRDOSE** **KOTAREDDY DILLI PRASAD REDDY**
2	Address of the Candidate	11-8-95/11, Lenin Nagar, Busdepo Road, Khammam, Telangana, India -507001. **Mobile Number: 8185826974** 2-746/A, Santhi nagar colony, Kongareddy palli, Chittor,Chittor, AndhraPradesh, India - 517001 **Mobile Number: 8919444057**
3	Registration Number	RA1811003010640 RA1811003010662
4	Date of Birth	13/04/2001 24/06/2001
5	Department	Computer Science and Engineering
6	Faculty	Engineering and Technology
7	Title of the Project Report	PREDICTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP LEARNING TECHNIQUES
8	Name and address of the Guide	Dr. R. Rajkamal **Mail ID:** rajkamar@srmist.edu.in **Mobile Number: 9176689008**
9	Name and address of the Co-Supervisor / Co- Guide (if any)	Nill
10	Software Used	Turnitin
11	Date of Verification	09/05/2022

Date :

12	**Plagiarism Details:**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (including selfcitation)**	**Percentage of similarity index (Excluding selfcitation)**	**% of plagarism after excluding Quotes, Bibliography, etc.,**
1	Introduction	<1%	1%	<1%
2	Literature Review	<1%	<1%	<1%
3	System Architecture and Design	<1%	<1%	<1%
4	Methodology	<1%	<1%	<1%
5	Coding and Testing	<1%	<1%	<1%
6	Results and Observations	<1%	<1%	<1%
7	Conclusion	<1%	<1%	<1%
Report abstract		0%\	0%	0%
Appendices		0%	0%	0%
I declare that the above information has been verified and found true to the best of my knowledge.				
Signature of the Candidate		**Signature of the Guide**		
		Signature of the HOD		

Date :

Prediction of covid 19 from chest X ray images using deep learning techniques

ORIGINALITY REPORT

4%	1%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Birkbeck College Student Paper	1%
2	Elene Firmeza Ohata, Gabriel Maia Bezerra, Joao Victor Souza das Chagas, Aloisio Vieira Lira Neto et al. "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning", IEEE/CAA Journal of Automatica Sinica, 2020 Publication	<1%
3	Submitted to Universidad San Francisco de Quito Student Paper	<1%
4	B M Anakha, Greshma Shaji, S Geetha. "Detecting COVID-19 from Chest X-Ray Images using Deep Learning", 2021 5th International Conference on Information Systems and Computer Networks (ISCON), 2021 Publication	<1%
5	www.mdpi.com Internet Source	<1%

PAPER PUBLICATION

IEEE CONFERENCE PUBLICATION

PREDICTION OF COVID-19 FROM CHEST X-RAY IMAGES USING DEEP LEARNING TECHNIQUES was SUBMITTED with Paper ID-876 to 2022 The International Conference for Intelligent Technologies in Karnataka, India.

Submission Summary

Conference Name
2022 The International Conference for Intelligent Technologies, Karnataka, India.

Paper ID
876

Paper Title
Prediction of Covid-19 from Chest X-Ray Images Using Deep Learning Techniques

Abstract
The coronavirus has been the most dangerous infectious virus that spread across the globe and is popularized as Covid-19. These infections affect and damage the human lungs and which may lead to the death of the patient. Early diagnosis of patients infected by the virus can help in preventing the patient by the spread of infection to others. Deep learning approaches can help doctors in detecting whether it's Covid positive, Normal, or Pneumonia from chest x-ray scans. The CNN model helps in the early diagnosis of the virus using chest X-ray images, as it is one of the fastest and most cost-effective ways to know whether a person is suffering from covid/pneumonia or not. For classification, we have used eight different convolutional neural networks (CNN) models, which were trained using a dataset containing 15000 chest x-ray images. The model has trained three (3) class classifications on Covid- Positive, Pneumonia, and Normal cases based on the dataset that included chest X-ray (CXR) images. It is Observed that the CNN model gives the highest results among all other models used in terms of accuracy of the model is 96.5% and other performance metrics like precision, recall, and F1-score of 95% for Covid Positive, whereas the precision, recall, and F1-score of 98%, 97%, and 97% for Normal, and also the precision, recall, and F1-score of 91%, 98%, and 94% for Pneumonia and AUC (area under the curve) SCORE of 99.5%, respectively.

Created on
12/04/2022, 11:27:19

Last Modified
12/04/2022, 11:27:19

Authors
Shaik Firdose (SRM Institute of Science and Technology) < sf6816@srmist.edu.in>
Kota Reddy Dilli Prasad Reddy (SRM Institute of Science and Technology) < kz4954@srmist.edu.in>
R Rajkamal (SRM Institute of Science and Technology) < rajkamar@srmist.edu.in>

Submission Files
covid paper publish(SRM).pdf (1.1 Mb, 12/04/2022, 11:26:49)

Figure A.1: Paper Publication in conference