DEEP ANALYSIS AND DIAGNOSIS OF COVID-19 USING MACHINE LEARNING

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Divya Sharma (181254)

Kritika Shelly (181251)

Under the supervision of

Dr. Ekta Gandotra

(Assistant Professor (SG))

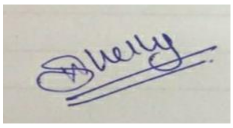
To

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CANDIDATE'S DECLARATION

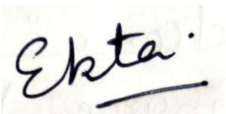
I hereby declare that the work presented in this report entitled **"Deep Analysis and Diagnosis of Covid-19 using Machine learning "** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2021 to May 2022 under the supervision of **Dr. Ekta Gandotra** (Assistant Professor, Senior Grade, Computer science and engineering/Information Technology department). The matter embodied in the report has not been submitted for the award of any other degree or diploma.



Kritika Shelly,181251.

Divya Sharma,181254.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



Dr. Ekta Gandotra
Assistant Professor, Senior Grade
Computer science and engineering/Information Technology department
Dated: 14th May, 2022

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing which makes us possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Ekta Gandotra, Assistant Professor (Senior Grade),** Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of **“Machine Learning & Deep Learning”** helps us to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Ekta Gandotra**, Department of CSE, for her kind help to finish me this project.

I would also generously welcome each one of those individuals who have helped us straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

TABLE OF CONTENTS

CHAPTER 4: PERFORMANCE ANALYSIS

CHAPTER 5: CONCLUSIONS

LIST OF ABBREVIATIONS

DL	Deep Learning
GPU	Graphics Processing Unit
AI	Artificial Intelligence
WHO	World Health Organization
ML	Machine Learning
NN	Neural Networks
RT-PCR	Reverse transcription-polymerase chain
CT	Computed tomography
TL	Transfer Learning
CNN	Convolutional Neural Networks
ROC	Reciever Operating Characterstic Curve

LIST OF FIGURES

DESCRIPTION **PAGE NO.**

LIST OF GRAPHS

DESCRIPTION **PAGE NO.**

LIST OF TABLES

DESCRIPTION **PAGE NO.**

ABSTRACT

The goal of this research project is to examine and illustrate India's whole Covid-19 pandemic scenario. We are using machine learning and deep learning to analyse the number of confirmed cases, fatalities, and recovered cases in each state throughout the country in this research. Following that, we examined the amount of hospital beds available in each state as well as the vaccination supply. Following that, we used matplotlib, seaborn, and Plotly to visualize the pandemic scenario in India and conduct comparative research across each state. Then, based on the CT scans or chest Xray given by the user, we developed a model based on deep learning that can forecast the presence of covid-19 in that respective patient. Various transfer learning techniques and neural networks were used to train eight deep learning models on chest X-rays and CT scans. The transfer learning techniques used while implementing 10 DL models are VGG16, InceptionV3, Xception, Convolutional Neural Networks (CNN) and ResNet50. Eighty percent of the photos were utilized to train the models, while the remaining twenty percent were used to assess their accuracy. On Google Colab GPU, the models were trained for 100 epochs on a total of 1694 pictures of chest X-rays and CT scans. At last, we created a user-friendly interface of the models trained using flask which allows users to upload their chest X-rays or CT scans and receive a report on the likelihood of COVID infection basically, it will predict the presence of Sars cov-2 in the respective patient based on his/her CT scan or Chest Xray. Thus, machine learning and deep learning helps in the prevention of lives by forecasting the risk of infection at an early stage. It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale. This project when implemented on large scale also assist radiologists as these automated programs help them to identify the diseases on a much closer and accurate way. It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale.

CHAPTER 1

INTRODUCTION

1.1 Introduction:

The coronavirus illness that is COVID-19 is a global pandemic that was first identified in December 2019 in Wuhan, the capital of Hubei province in mainland China. This Coronavirus is known as SARS-CoV-2 that is Severe Acute Respiratory Syndrome Coronavirus 2. This global pandemic was declared a Public Health Emergency of International Concern by the WHO in January 2020 over its massive exponential increment in number of deaths globally. Coronavirus is a virus family that often causes respiratory tract infections and can be deadly. COVID-19 spread mostly by droplets that are produced when an infected person coughs, sneezes, or exhales. You can become infected by breathing in COVID-19 or by touching a contaminated surface and then contacting your eyes, nose, or mouth if you are in immediate contact to someone who has it.The majority of persons infected with the COVID-19 virus will have mild to moderate respiratory symptoms and will recover without needing any intensive medical supervision but people over the age of 65, as well as those with some underlying medical conditions such as cardiovascular disease, diabetes, chronic respiratory disease, and cancer, are at a higher risk of developing serious illness. Covid is such a dangerous disease that even if you recover from it, you may have post-covid symptoms or diseases such as black fungus, persistent joint aches, loss of taste and smell, extended weariness, and other complications. According to recent research, once the coronavirus pandemic begins, the healthcare system would be overwhelmed in less than four weeks and when a hospital's capacity is exceeded, the death rate rises and hence make this pandemic a never-ending disease. Hence, controlling this deadly pandemic is necessary but not impossible so for controlling these pandemic vaccines are made available after their satisfactory clinical trials in 2021.Vaccines are accessible, however owing to the large population, it takes a long time to vaccinate each and every person. Hence, in order to control the spread of this disease, it is important that we continue to take precautions like social distancing, double masking, by proper washing your hands or using an alcohol-based sanitizer frequently along with vaccination. In this global health disaster, the health sector is avidly seeking new technology and strategies to detect and manage the spread of the coronavirus outbreak. Artificial Intelligence is currently one of the most essential aspects of global technology, since it can track and monitor the rate at which the Corona virus develops as well as determine the danger and severity of Corona virus patients. By thoroughly analysing past patient data, AI may also predict the risk of death and severity in that respective patient. Artificial intelligence can help us fight the virus by providing individual testing, medical aid, data and information, and disease control suggestions. AI is a big umbrella that encompasses several sub-areas in order to address difficult challenges in our life. Learning, preparing, thinking, information representation, and seeking are some of the sub-areas. ML and DL are subfields of AI that combine numerous methods to create intelligent models for identifying or categorizing certain jobs. Machine Learning is widely used in the healthcare industries nowadays. It is used for diagnosis, detection, analysis and prediction of different diseases that can infect humans. Machine learning plays a crucial role to detect the hidden patterns and thereby analyse the respective data statistically. DL, on the other hand, is a subset of machine learning that focuses on creating deep structural NN models that learn from data using feedforward and backpropagation techniques. Following ML, the DL arose and exceeded it in various tasks during the previous two decades. Nonetheless, understanding and analysing a large volume of data is required. Deep learning, also known as hierarchical learning or deep structured learning, is a machine learning subfield that analyses data using a layered algorithmic framework. Data is filtered via a cascade of many layers in deep learning models, with each subsequent layer using the output from the preceding one to determine its outcomes. As more data is processed, deep learning models grow more accurate, effectively learning from prior results to improve their capability to make correlations and connections. Deep learning is based on how biological neurons in animals' brains link with one another to process information. Each succeeding layer of nodes is triggered when it receives impulses from its surrounding neurons, similar to how electrical signals propagate between the cells of biological creatures. Each layer of artificial neural networks, which provide the foundation for deep learning models, may be assigned a specific component of a transformation job, and input may pass through the layers numerous times to refine and optimise the final output.

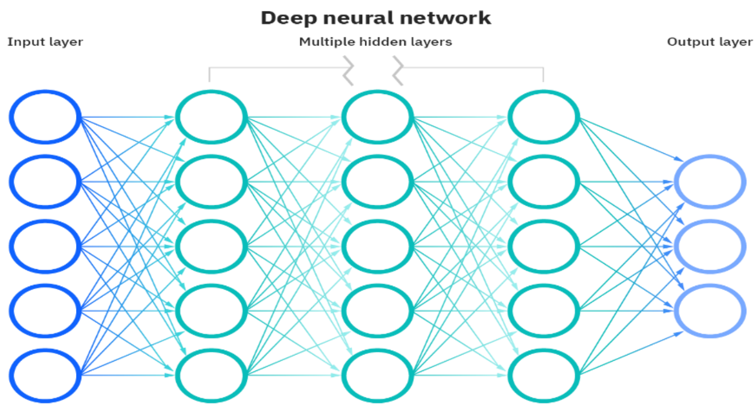


Figure 1.1: Overview of Deep Learning Neural Network

The mathematical translation operations that convert raw input into understandable output are performed by these hidden layers.

In the COVID-19 conflict, research efforts in the industrial, medical, and technical sectors have successfully adopted sophisticated AI-based, ML and DL approaches within a short amount of time following the start of COVID-19 and made significant progress. For example, in medical image analysis, ML and DL aid COVID-19 diagnosis as well as providing non-invasive detection techniques to protect medical employees from infections, and the patient's severity score is also provided for future treatment. ML and DL are utilised in virology investigations to look at SARS-CoV-2 protein-related genetics and forecast new combinations that can be employed in medication development and immunisation. Furthermore, AI intelligent models built on ML and DL learn to build disease transmission models that effectively anticipate outbreaks, transmission course, transmission list, and impacts on a broad scale. In addition to security checks in airports, patient tracking, and pandemic detection, ML and DL are widely utilised in pandemic protection and public surveillance. Medical imaging, X-ray, and CT play a crucial part in the global fight against COVID-19, and recent AI breakthroughs tend to increase the capacity of imaging technologies and make healthcare personnel's lives easier. Medical imaging studies are widely used by clinicians to determine the existence of COVID-19. These clinical imaging tests generally use X-ray and CT scan images of the chest and lungs for determination. In medical imaging testing AI plays a vital role. It has achieved extraordinary precision in image identification, organ recognition, geographic infection categorization, and disease severity. These computer-assisted medical imaging networks also help radiologists make clinical judgments, such as illness detection, monitoring, and prognosis with much effectiveness and accuracy.

Figure 1.2: Significance of AI in Healthcare industry

1.2 Problem Statement:

Because precautionary or experimental vaccine treatment for the COVID-19 infection, which causes extreme acute respiratory syndrome, is not 100 percent effective, Hence, early detection is crucial in allowing infected people to gain rapid protection and reducing the risk of infection for the general population. RT-PCR and gene sequencing of respiratory or blood samples are important diagnostic tools for COVID-19. However, evaluating throat swab samples yields an overall positive RT-PCR average of about 30% to 60%, resulting in undetected COVID-19 infected individuals who might infect a broad and healthy community and thus spread this deadly virus. Long delays in receiving test results, patients with high clinical suspicion testing erroneously negative on first RT-PCR tests, and a plethora of additional laboratory logistical issues are all intrinsic drawbacks of this technology. Sub-optimal clinical sample procedures, fluctuations in viral load, and manufacturer test kit sensitivity may all contribute to low test sensitivity of RT-PCR. Also, managing these RT-PCR negative patients is incredibly difficult in areas where caseloads are at an all-time high. The laboratory's procedure adherence requirements, as well as a variety of testing features, might be ascribed to its limitations. Hence, Laboratories and virology research Centre's are now attempting to overcome the present limits of RT-PCR testing in order to detect the coronavirus with greater accuracy. Chest imaging evaluation is a helpful and an effective way for identifying clinical signs of COVID-19 suspected patients, according to World Health Organization guidelines from October 2020.Given the high frequency of COVID-19 and the lack of qualified radiologists, thus the automatic medical imaging techniques for recognizing COVID-19 infection can improve the diagnostic process and promote high-precision diagnosis at an early stage. AI, ML and DL approaches are powerful technologies that can be applied to create early detection procedures for COVID-19. Using chest X-ray and CT scan datasets, we developed an end-to-end DL architecture to categorize COVID-19 in this research project. Unlike standard AI/ML approaches, which require a two-stage process of manual feature extraction followed by image recognition to categorize the medical pictures, our method uses only one step. We have created this DL-based approaches that can accurately estimate COVID-19 from raw photos without the need for feature extraction. These deep-level learning models, specifically the convolution neural networks, have recently exceeded the traditional AI based models in most computer vision and medical image processing operations, and have been used for a variety of tasks such as image grouping, image segmentation, super-resolution, and image improvement. Deep - learning based diagnostic tools for COVID-19 would provide an automated second review to healthcare professionals from multiple imaging modes such as X-Ray, Ultrasound, and CT, aiding in the diagnosis and criticality assessment of COVID-19 patients, allowing for better decision making in the global fight against the disease. Furthermore, COVID-19 frequently causes pneumonia, making it difficult for radiologists and doctors to distinguish between COVID-19 pneumonia and other viral and bacterial pneumonias based entirely on diagnostic imaging. As a result, our DL-based technique can assist in the accurate diagnosis of COVID-19 and compensate for the scarcity of skilled physicians in the field. This methodology, if deployed on a big scale, might aid clinical practitioners in manual diagnosis.

Also, alongside to this diagnosis as we know that Covid-19 is a disease that spreads at an exponential pace, understanding its pattern is critical to controlling this fatal disease. During the second wave, it was also extremely difficult for individuals to locate hospital beds and vaccination centres across India. In order to address this problem, we're employing machine learning and artificial intelligence to analyse and display the number of confirmed cases, deaths, and recovered cases in each state throughout the country.We also intend to examine the amount of hospital beds and icmr labs available in each state as well as the vaccination supply. Also, the user friendly interface of the DL based model trained will predict the severity and presence of Sars cov-2 in the respective patient. It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale.

1.3 Objective:

The goal of this research project is to help combat the COVID-19 pandemic by introducing an automated and quick COVID-19 diagnosis system as a simple alternative diagnosis technique to reduce COVID-19 spread and improve the accuracy of conventional diagnosis systems that is RT-PCR testing. Hence, Laboratories and virology research Centre's are now attempting to overcome the present limits of this conventional system by introducing an accurate DL based COVID-19 diagnosis system. Chest imaging evaluation is a helpful and an effective way for identifying clinical signs of COVID-19 suspected patients, according to WHO guidelines from October 2020.Given the high frequency of COVID-19 and the lack of qualified radiologists, thus the automatic medical imaging techniques for recognizing COVID-19 infection can improve the diagnostic process and promote high-precision diagnosis at an early stage.

As a result, the purpose of this project is to create a COVID-19 diagnosis system that is accurate using cutting-edge NN architectures and TL techniques. Using the CT scans or chest Xrays provided by the user, we created a deep learning model that can predict the existence of covid-19 in that particular patient. On chest X-rays and CT images, eight deep learning models were trained using a variety of transfer learning techniques and neural networks. VGG16, InceptionV3, Xception, Convolutional Neural Networks (CNN) and ResNet50 are the transfer learning techniques utilized to create 10 DL models. The models were trained using 80% of the pictures, while the remaining 20% were used to test their correctness. The models were trained for 100 epochs on a total of 1694 images of chest X-rays and CT scans on a Google Colab GPU. Thus, machine learning and deep learning helps in the prevention of lives by forecasting the risk of infection at an early stage. It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale. This project when implemented on large scale also assist radiologists as these automated programs help them to identify the diseases on a much closer and accurate way.

1.4 Methodology Used:

Step 1: Dataset Acquisition:

In this research project we have taken 7 datasets from Kaggle. The datasets taken are stated below with the information they contain:

1.1 **AgeGroupDetails.csv:** This dataset contains the records of the covid-19 patients age-wise in India during covid-19 second wave. This gives the insight about the rate of infection age-wise.

1.2 **Covid_19_India.csv:** This dataset contains the daily records of all the states across India with their cured, confirmed and deaths during covid-19 second wave.

1.3 **Covid_vaccine_statewise:** This dataset contains the daily records of vaccination process in all states across India with the information about number of doses administered, gender-wise vaccination analysis, Covaxin and Covishield vaccination administered.

1.4 **HospitalBedsIndiaLocations:** This dataset contains the records of all the hospitals across India with the information of about how many hospital beds available across each state during covid-19 second wave.

1.5 **ICMRTestingLabs:** This dataset contains the records of all the testing labs available which are approved by ICMR across different states in India.

1.6 **StatewiseTestingDetails:** This dataset contains the records of all the samples taken across each state in India and also contains the information about the positive and negative samples collected across India during covid-19 second wave.

1.7 **Data:** This folder contains the information of covid-19 positive and negative CT scans and Chest Xray images. This dataset contains approximately 1694 images of healthy and infected lungs in the form of CT scans and chest Xray's.

Step2: Data Preprocessing:

Data preprocessing is the process of cleaning, checking for missing values, organizing the raw data into a form so that it can be analysed accurately.

Step3: Analyse and visualize the dataset:

we used matplotlib, seaborn, and Plotly to visualize the pandemic scenario in India and conduct comparative research across each state.

Step 4: Implement Different TL techniques:

Eight deep learning models were trained using a variety of transfer learning techniques and neural networks. VGG16, InceptionV3, Xception, Convolutional Neural Networks(CNN) and ResNet50 are the transfer learning techniques utilized to create 10 DL models

Step5: Evaluate the performance of the techniques applied:

Training dataset is used to train the model and testing dataset is used to test and analyse the performance of the trained model. These ten DL based models were analyzed and their performance is calculated on the basis of accuracy score, precision score, recall score and F-score. The model with highest accuracy score will be selected for deployment in the model when implemented on large scale.

Step6: Develop the interface of the models trained for diagnosing the presence and severity of Covid-19:

At last, we created a user-friendly interface of the models trained using flask which allows users to upload their chest X-rays or CT scans and receive a report on the likelihood of COVID infection basically, it will predict the presence of Sars cov-2 in the respective patient based on his/her CT scan or Chest Xray.

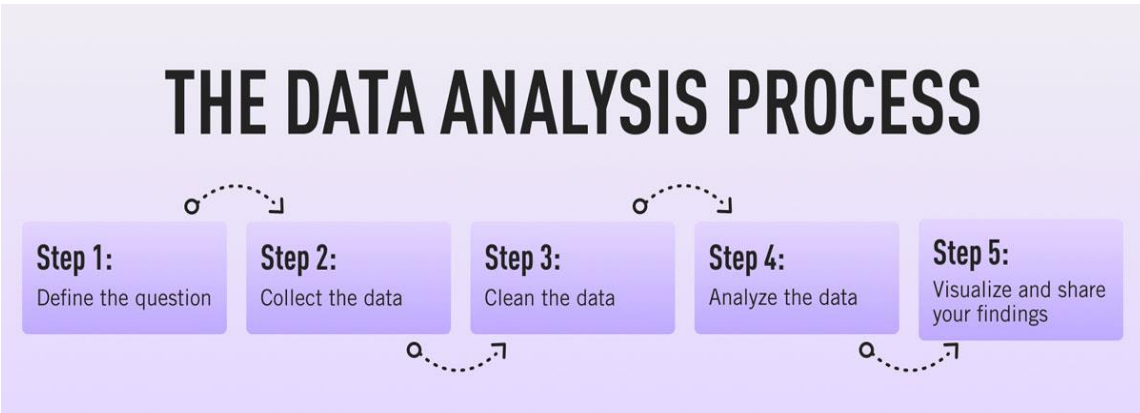


Figure 1.3: Data Analysis and Visualization process used

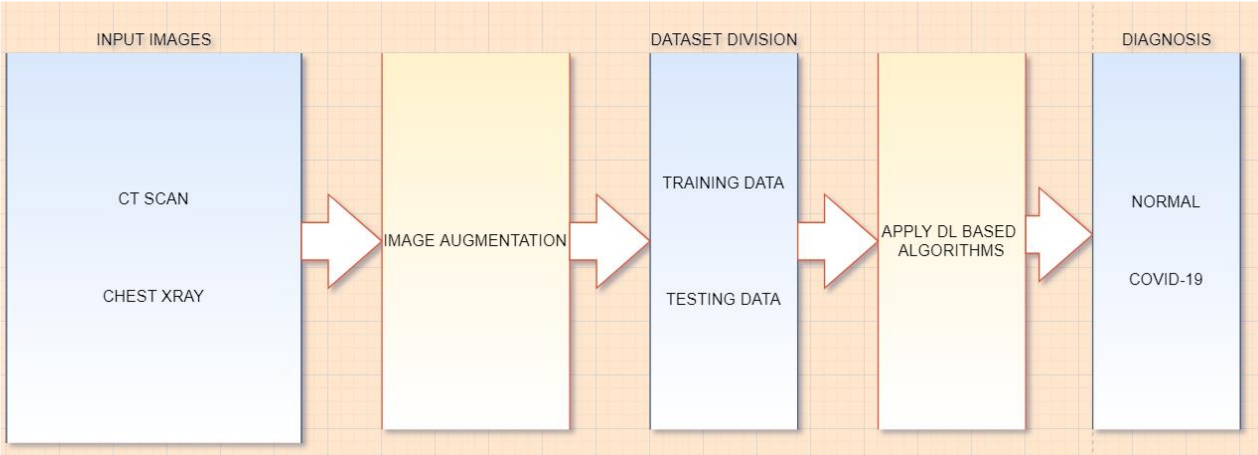


Figure 1.4: Proposed DL based Covid-19 diagnosis system

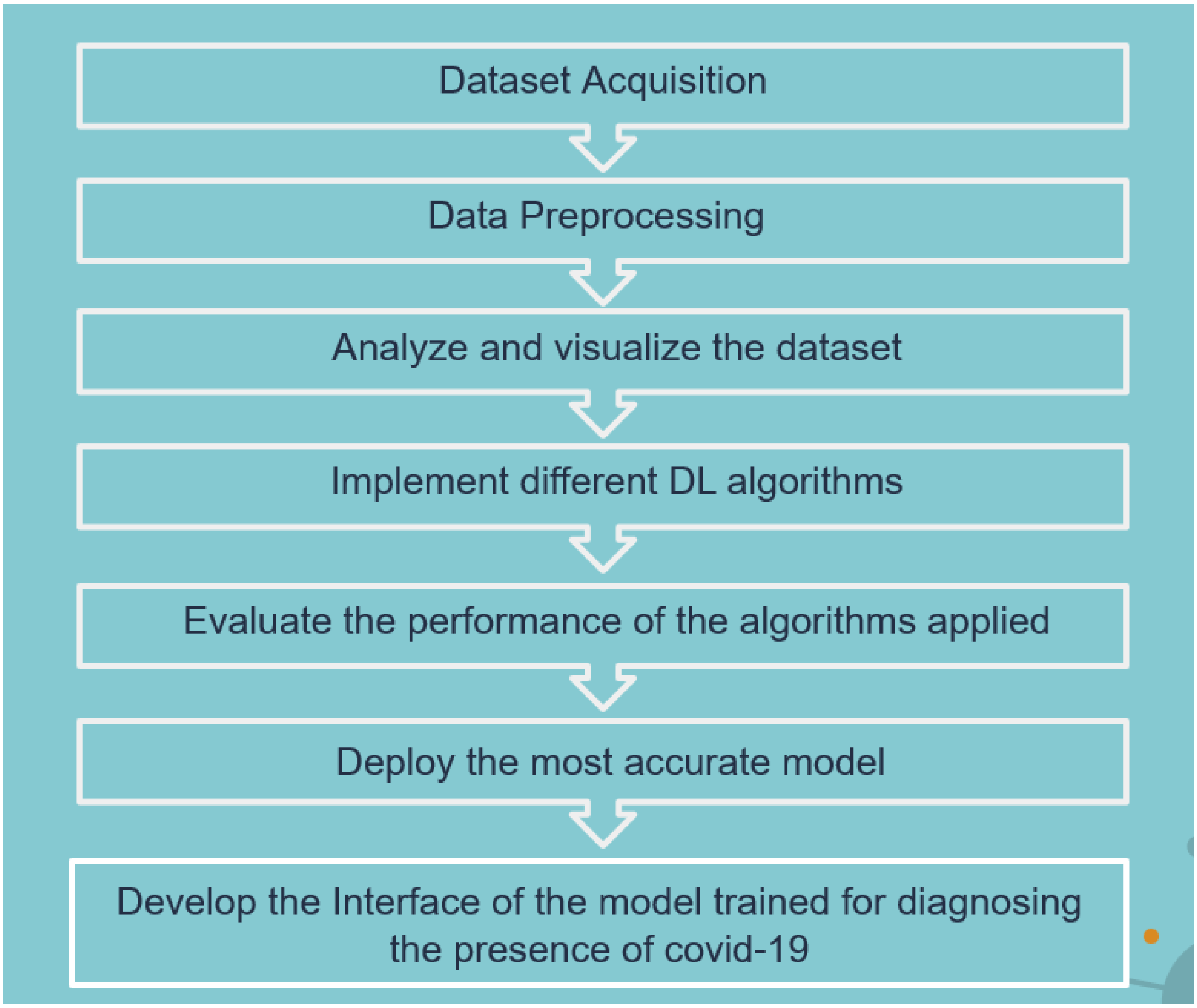


Figure 1.5: Proposed working methodology used

CHAPTER 2

LITERATURE SURVEY

S. Lafraxo , et al. [1] suggested the "CoviNet" research project, which is a deep learning network that can identify COVID19 in chest X-ray pictures automatically. They proposed a convolutional neural network architecture based on an adaptive median filter, histogram equalisation, and an adaptive median filter. It was trained from start to finish using publicly available information. Their suggested model has a binary classification accuracy of 98.62 percent and a multi-class classification accuracy of 95.77 percent.

A.A Khan, et al. [2] states that the current approaches for screening coronavirus patients have a high probability of false positives hence, an accurate and efficient screening approach for coronavirus detection is required. As a result, alternative trustworthy approaches such as CT imaging are used to reliably diagnose coronavirus. They introduced a 3D-Deep learning-based technique for screening coronavirus patients utilising 3D volumetric CT imaging data in their publication. Their suggested approach aids physicians in effectively identifying COVID-19 patients. They used different state-of-the-art 3D Deep learning-based approaches such as 3D ResNets, C3D, 3D DenseNets etc to conduct extensive experiments on two datasets, CC-19 and COVID-CT. The results of their experiments suggest that their proposed technique is competitively successful than the conventional systems.

Aritra Ghosh, et al. [3] have performed a research work in they have discussed how India is prepared to deal with an increasing number of COVID-19 cases, the current situation, including the negative effects on the economy. In their study, the death and recovery rates were analysed, and the covid patterns were meticulously watched.

S. Bouaafia,et al. [4] presented a research project on disease detection system based on deep learning. Deep transfer learning models are presented that are based on a pre-trained Deep convolutional Neural Network. In their study, many pre-trained models were used, including DensNet201, InceptionV3, VGG16, and ResNet50.The training datasets used in their research include a combination of X-ray and CT pictures divided into two categories that is Normal and COVID-19. According to the test accuracy measure stated in their research paper, the DensNet201 was the best suited deep transfer model, and it had achieved a 97 percent accuracy for detecting covid-19 patients.

R. Kumari,et al.[5] presented their work that examines recently established forecasting models in depth and forecasts the number of confirmed, recovered, and mortality cases caused by COVID-19 in India. To increase the accuracy, correlation coefficients and multiple linear regression were employed for prediction, as well as autocorrelation and autoregression .The projected number of instances is in good agreement with their actual data, with an R-squared score of 0.9992. According to their findings, lockdown and social separation are two crucial characteristics that can aid to slow the development of COVID-19.

Ranjan Gupta, et al. presents a survey of various models [6] based on ML algorithms and analyses their performance. The research conducted by them provided a thorough examination of the COVID-19 pandemic in India. They presented the infected case growth trends in India, making predictions for the number of infected cases in the coming days, number of infected cases in India, and pattern mining on coronavirus patients.

M. M. Islam, et al. [7] presented their research that seeks to provide an overview of newly developed deep learning systems that use various medical imaging modalities such as CT scans and X-rays. Their analysis focuses on deep learning-based systems for COVID-19 diagnosis, as well as well-known data sets that were utilised to train the networks. It also discusses the numerous data partitioning approaches and performance measurements created by scholars in this subject. They also proposed a taxonomy to categorise contemporary works. Finally, they assisted professionals and technicians in comprehending how deep learning techniques are applied in this field. Their proposed work stated that specialists and technicians comprehend on how deep learning methods are employed in this respect and how they may be used even more effectively to battle the COVID-19 pandemic.

Y. Jiang, et al.[11] suggested a conditional generative adversarial network-based CT image synthesis technique that can efficiently create high-quality and realistic COVID-19 CT images for use in deep-learning-based medical imaging tasks. With the created COVID-19 CT images, the suggested technique outperforms previous state-of-the-art image synthesis methods, indicating promise for different machine learning applications such as semantic segmentation and classification.

The objective behind the proposed work after reviewing the above research papers is to make an automated covid-19 detection system based on DL that can help the medical professionals in the long run. Unlike standard AI/ML approaches, which require a two-stage process of manual feature extraction followed by image recognition to categorize the medical pictures, our method uses only one step. We have created this DL-based approaches that can accurately estimate COVID-19 from raw photos without the need for feature extraction. These deep-level learning models, specifically the convolution neural networks, have recently exceeded the traditional AI based models in most computer vision and medical image processing operations, and have been used for a variety of tasks such as image grouping, image segmentation, super-resolution, and image improvement. Deep - learning based diagnostic tools for COVID-19 would provide an automated second review to healthcare professionals from multiple imaging modes available. We created an user-friendly interface of the models trained using flask which allows users to upload their chest X-rays or CT scans and receive a report on the likelihood of COVID infection basically, it will predict the presence of Sars cov-2 in the respective patient based on his/her CT scan or Chest Xray. Thus, machine learning and deep learning helps in the prevention of lives by forecasting the risk of infection at an early stage.It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale. This project when implemented on large scale also assist radiologists as these automated programs help them to identify the diseases on a much closer and accurate way.

CHAPTER 3

SYSTEM DEVELOPMENT

Over the years, several models of the design and development process have been produced, each portraying it for a different purpose and from a different perspective. For our research project, we have used two methods experimental and analytical while developing the model.

3.1 Feasibility Study on major project:

Due to the limitations of the conventional methods for testing covid-19, thereby our entire purpose of this project is to create a COVID-19 diagnosis system that is accurate using cutting-edge NN architectures and TL techniques. Using the CT scans or chest Xrays provided by the user, we created a deep learning model that can predict the existence of covid-19 in that particular patient. On chest X-rays and CT images, ten deep learning models were trained using a variety of transfer learning techniques and neural networks. VGG16, InceptionV3, Xception, Convolutional Neural Networks (CNN) and ResNet50 are the transfer learning techniques utilized to create 10 DL models. The models were trained using 80% of the pictures, while the remaining 20% were used to test their correctness. The models were trained for 100 epochs on a total of 1694 images of chest X-rays and CT scans on a Google Colab GPU. Thus, machine learning and deep learning helps in the prevention of lives by forecasting the risk of infection at an early stage.

3.2 System Requirements:

3.2.1 Functional Requirements:

3.2.1.1 Languages used:

1. Python 3.8.5

3.2.1.2 Software Used:

1.Google colaboratory

3.2.1.3 Packages Used:

1. Flask (version 1.1.2)
2. Tensor flow (version 2.4.1)
3. Matplotlib (version3.2.0)
4. Numpy (version 1.19.5)
5. Opencv-python (version 4.5.1.48)
6. Pandas
7. Folium
8. Seaborn
9. Plotly

3.2.2 Non-Functional Requirements:

3.2.2.1 Technical Requirements:

1. Intel Core i7 processor at 1.80 GHz, 8 GB of DRAM.
2. Disk space 1 TB
3. Operating System: 64-bit Windows 10 Pro.

3.3 Flow-diagram of the major project:

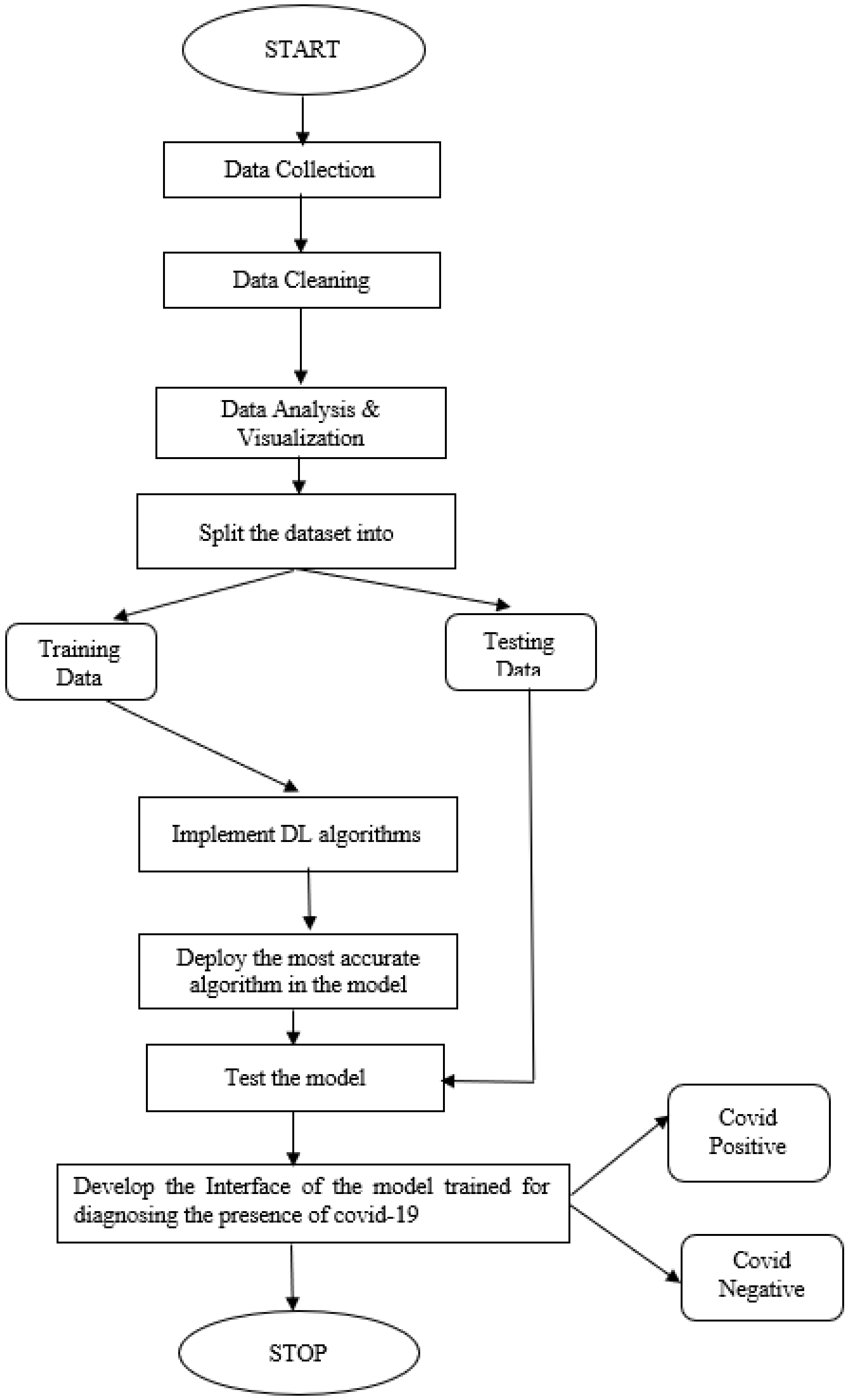


Figure 3.1: Flow diagram of the methodology used

3.4 Dataset Acquisition:

3.4.1 Covid_19_India.csv: The dataset contains 16383 records and 9 columns. The attributes contained in this dataset are Sno, Date, Time, State/UnionTeritory, ConfirmedIndianNational, ConfirmedForeignNational, Cured, Deaths, Confirmed. This dataset contains the daily records of all the states across India with their cured, confirmed and deaths during covid-19 second wave.

A	B	C	D	E	F	G	H	I
Sno	Date	Time	State/Unic	Confirmed	Confirmed	Cured	Deaths	Confirmed
1	########	6:00 PM	Kerala	1	0	0	0	1
2	########	6:00 PM	Kerala	1	0	0	0	1
3	########	6:00 PM	Kerala	2	0	0	0	2
4	########	6:00 PM	Kerala	3	0	0	0	3
5	########	6:00 PM	Kerala	3	0	0	0	3
6	########	6:00 PM	Kerala	3	0	0	0	3

Figure 3.2: Covid_19_India.csv

3.4.2 Covid_vaccine_statewise: The dataset contains 6032 records and 17 columns. The attributes/features contained in this dataset are: updated on, state, total individuals vaccinated, total sessions conducted, total sites, first dose administered, second dose administered, male(individuals vaccinated, female(individuals vaccinated), transgender(individuals vaccinated), total covaxin administered, total covishield administered, total sputnik-v administered,18-45 years, 45-60 years, 60+ years, total doses administered. This dataset contains the daily records of vaccination process in all states across India with the information about number of doses administered, gender-wise vaccination analysis, Covaxin and Covishield vaccination administered.

B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
State	Total Indiv	Total Sessi	Total Sites	First Dose	Second Dc	Male(Indiv	Female(Inc	Transgend	Total Cova	Total Covi	Total Sput	AEFI	18-45 year	45-60 year	60+ years	Total Doses Administered	
India	48276	3455	2957	48276	0	23757	24517	2	579	47697						48276	
India	58604	8532	4954	58604	0	27348	31252	4	635	57969						58604	
India	99449	13611	6583	99449	0	41361	58083	5	1299	98150						99449	
India	195525	17855	7951	195525	0	81901	113613	11	3017	192508						195525	
India	251280	25472	10504	251280	0	98111	153145	24	3946	247334						251280	
India	365965	32226	12600	365965	0	132784	233143	38	5367	360598						365965	

Figure 3.3: Covid_vaccine_statewise.csv

3.4.3 HospitalBedsIndiaLocations: The dataset contains 37 records and 14 columns. The features contained in this dataset are Sno, State/UT, NumPrimaryHealthCenters_HMIS, NumCommunityHealthCenters_HMIS,NumSubDistrictHospitals_HMIS,NumDistrictHospitals_HMIS,TotalPublicHealthFacilities_HMIS,NumPublicBeds_HMIS,NumRuralHospitals_NHP18,NumRuralBeds_NHP18,NumUrbanHospitals_NHP18,NumUrbanBeds_NHP18, Latitude, Longitude. This dataset contains the records of all the hospitals across India with the information of about how many hospital beds available across each state during covid-19 second wave.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Sno	State/UT	NumPrima	NumComn	NumSubDi	NumDistri	TotalPubli	NumPublic	NumRuralI	NumRuralI	NumUrbar	NumUrbar	Latitude	Longitude
1	Andaman &	27	4		3	34	1246	27	575	3	500	11.6234	92.7265
2	Andhra Pra	1417	198	31	20	1666	60799	193	6480	65	16658	15.9129	79.74
3	Arunachal	122	62		15	199	2320	208	2136	10	268	27.0844	93.6053
4	Assam	1007	166	14	33	1220	19115	1176	10944	50	6198	26.1433	91.7898
5	Bihar	2007	63	33	43	2146	17796	930	6083	103	5936	25.5941	85.1376

Figure 3.4: HospitalBedsIndiaLocations.csv

3.4.4 ICMRTestingLabs: The dataset contains 268 records and 6 columns. The features contained in this dataset are lab, address, pincode, city, state, type. This dataset contains the records of all the testing labs available which are approved by ICMR across different states in India.

	A	B	C	D	E	F	G
1	lab	address	pincode	city	state	type	
2	ICMR-Regi	ICMR-Regi	744103	Port Blair	Andaman &	Government Laboratory	
3	Tomo Riba	National H	791110	Naharlagu	Arunachal	Collection Site	
4	Sri Venkate	Sri Venkate	517507	Tirupati	Andhra Pra	Government Laboratory	
5	Rangaraya	Rangaraya	533001	Kakinada	Andhra Pra	Government Laboratory	
6	Sidhartha I	Siddhartha	520008	Vijayawad	Andhra Pra	Government Laboratory	
7	Governme	Governme	515001	Anantapur	Andhra Pra	Government Laboratory	
8	Guntur Me	Guntur Me	522004	Guntur	Andhra Pra	Government Laboratory	
9	Rajeev Gai	Rajiv Gand	516002	Puttampal	Andhra Pra	Government Laboratory	

Figure 3.5: ICMRTestingLabs.csv

3.4.5 StatewiseTestingDetails: The dataset contains 14659 records and 5 columns. The columns contained in the dataset are date, state, Total samples, Negative, Positive. This dataset contains the records of all the samples taken across each state in India and also contains the information about the positive and negative samples collected across India during covid-19 second wave.

	A	B	C	D	E
1	Date	State	TotalSamp	Negative	Positive
2	########	Andaman a	1403	1210	12
3	########	Andaman a	2679		27
4	########	Andaman a	2848		33
5	########	Andaman a	3754		33
6	########	Andaman a	6677		33
7	########	Andaman a	6965		33
8	########	Andaman a	7082		33
9	########	Andaman a	7167		33

Figure 3.6: StatewiseTestingDetails.csv

3.4.6 Data: This folder contains the information of covid-19 positive and negative CT scans and Chest Xray images. This dataset contains approximately 1694 images of healthy and infected lungs in the form of CT scans and chest Xray's.

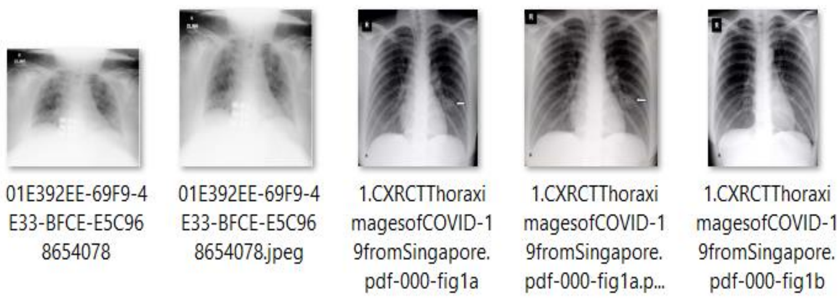


Figure 3.7: Chest covid images

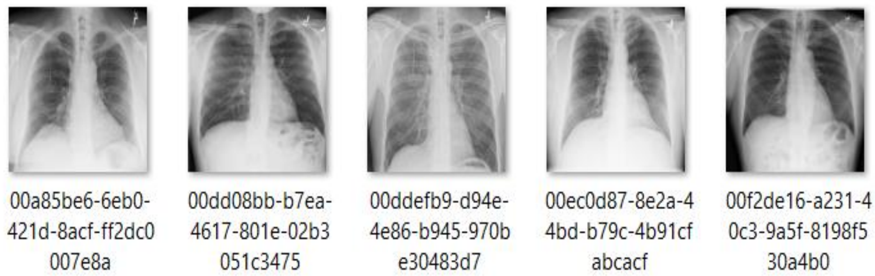


Figure 3.8: Chest non-covid images

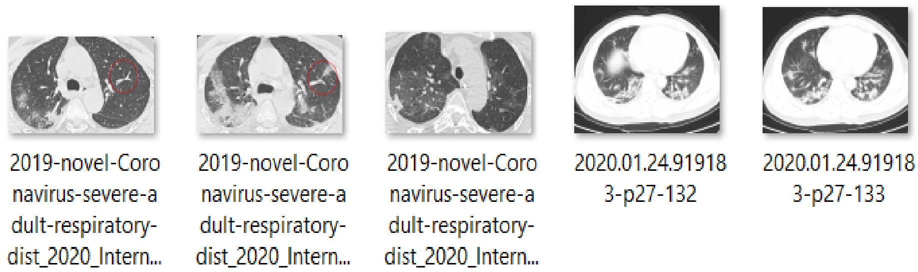


Figure 3.9: Ct-scan covid images

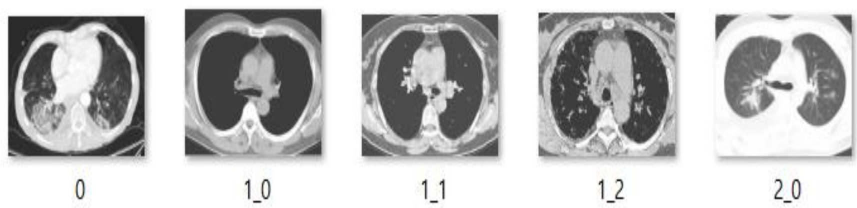


Figure 3.10: Ct-scan non-covid images

3.5 Data Analysis:

Data analysis is the process of modifying data. Obtaining raw data, organizing and cleaning it, storing it in a convenient location, and lastly applying various statistical approaches and to create visualizations are all part of data analysis.

Here, the analysis of data is done in 4 different stages-

1. Analysis of covid-19 cases across India.
2. Analysis of ICMR Lab Testing and state-wise Testing across India.
3. Analyzing the number of hospital beds available across India.
4. Analysis of Vaccination Process.

3.5.1 Analysis of Covid-19 cases across India:

The figure 3.11 shows that Maharashtra has the highest number of positive cases across different regions of India and figure 3.12 shows the number of confirmed, cured, deaths, recovery rate and mortality rate across different states of India during covid-19 2nd wave.

State	Positive	TotalSamples
Uttar Pradesh	126722	55999840
Maharashtra	1638961	40128355
Karnataka	264546	33189023
Bihar	90553	32396709
Tamil Nadu	367430	31675744
Gujarat	136004	23298784
Kerala	932639	22281273
Andhra Pradesh	235525	21361014
Delhi	151928	20975900
Telangana	124963	17888192
Assam	87908	14076432
West Bengal	135596	13839455
Odisha	97920	13283652
Madhya Pradesh	50640	11598397
Rajasthan	67954	11590904
Punjab	124535	10558558
Chhattisgarh	19459	10137066
Haryana	275137	9836144
Jammu and Kashmir	31371	9629643
Jharkhand	344914	9629486
Uttarakhand	14083	5365420
Himachal Pradesh	3993	2349178
Puducherry	44037	1258995
Tripura	63137	1227578
Goa	12333	898554
Manipur	4765	869552
Arunachal Pradesh	2658	732078
Meghalaya	1718	668949
Chandigarh	2305	555785
Mizoram	713	463729

Figure 3.11: Covid-19 cases across different regions of India.

State/UnionTerritory	Confirmed	Cured	Deaths	Recovery Rate	Mortality Rate
Maharashtra	5997587	5753290	119303	95.926745	1.989183
Kerala	2842247	2729967	12445	96.049604	0.437858
Karnataka	2819465	2668705	34287	94.652886	1.216082
Tamil Nadu	2443415	2358785	31746	96.536405	1.299247
Andhra Pradesh	1862036	1798380	12452	96.581377	0.668730
Uttar Pradesh	1704790	1678788	22336	98.474768	1.310191
West Bengal	1487363	1447510	17475	97.320560	1.174898
Delhi	1433366	1406629	24940	98.134670	1.739960
Chhattisgarh	992074	971057	13407	97.881509	1.351411
Rajasthan	951548	940465	8905	98.835266	0.935843
Odisha	886946	853012	3717	96.174062	0.419079
Gujarat	822758	807911	10040	98.195460	1.220286
Madhya Pradesh	789499	779177	8827	98.692589	1.118051
Haryana	767900	756426	9314	98.505795	1.212918
Bihar	720505	708231	9569	98.296473	1.328096
Bihar****	715730	701234	9452	97.974655	1.320610
Telangana	616688	596628	3598	96.747140	0.583439
Punjab	593572	572008	15923	96.367079	2.682573
Assam	490907	454726	4310	92.629765	0.877967
Telengana	443360	362160	2312	81.685312	0.521472
Jharkhand	344914	338446	5104	98.124750	1.479789
Uttarakhand	339127	329182	7068	97.067470	2.084175
Jammu and Kashmir	313028	301973	4273	96.468367	1.365054
Himachal Pradesh	201049	195301	3461	97.140995	1.721471
Goa	165197	159419	3013	96.502358	1.823883
Puducherry	115627	110838	1731	95.858234	1.497055
Manipur	65622	55257	1074	84.204992	1.636646
Tripura	63499	58978	660	92.880203	1.039386
Chandigarh	61520	60446	807	98.254226	1.311769
Meghalaya	46458	41349	797	89.002970	1.715528
Arunachal Pradesh	33916	31189	160	91.959547	0.471754
Nagaland	24541	22486	477	91.626258	1.943686
Ladakh	19881	19341	202	97.283839	1.016045
Sikkim	19589	16903	296	86.288223	1.511052
Mizoram	18624	14096	86	75.687285	0.461770

Figure 3.12: State wise analysis of Covid-19 cases.

The figure 3.13 depicts the visualization of Confirmed covid-19 cases across different regions of India. From the visualization, it is concluded that Maharashtra has large number of confirmed cases during the 2^{nd} wave of covid-19.

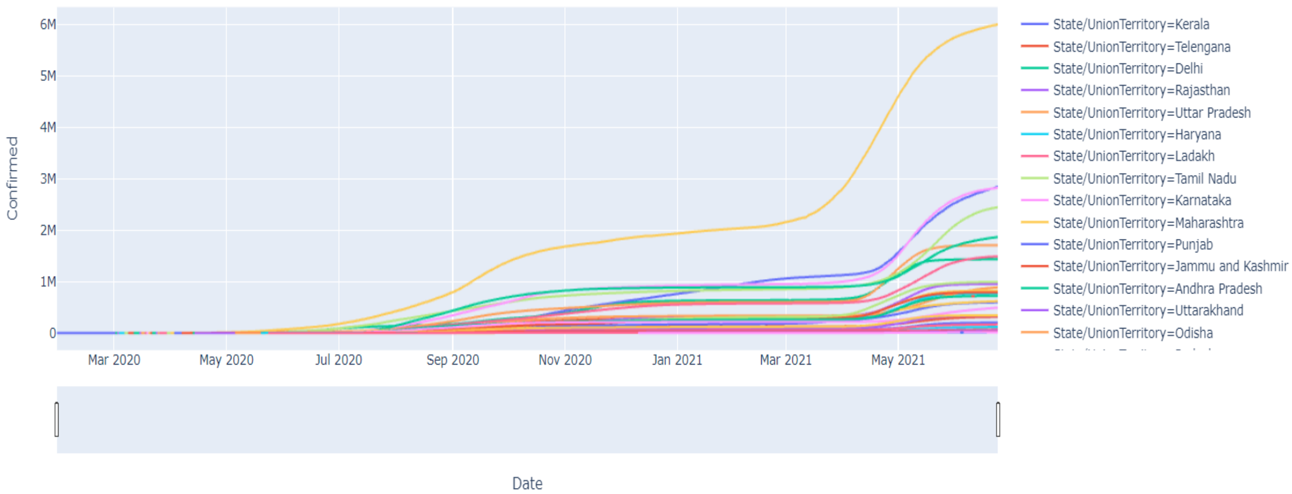


Figure 3.13: Confirmed covid-19 cases across different regions of India

Since Delhi and Maharashtra were the most affected regions during covid-19 second wave. Hence, the figure 3.14 shows the geospatial visualization of number of covid-19 cases present in these states.

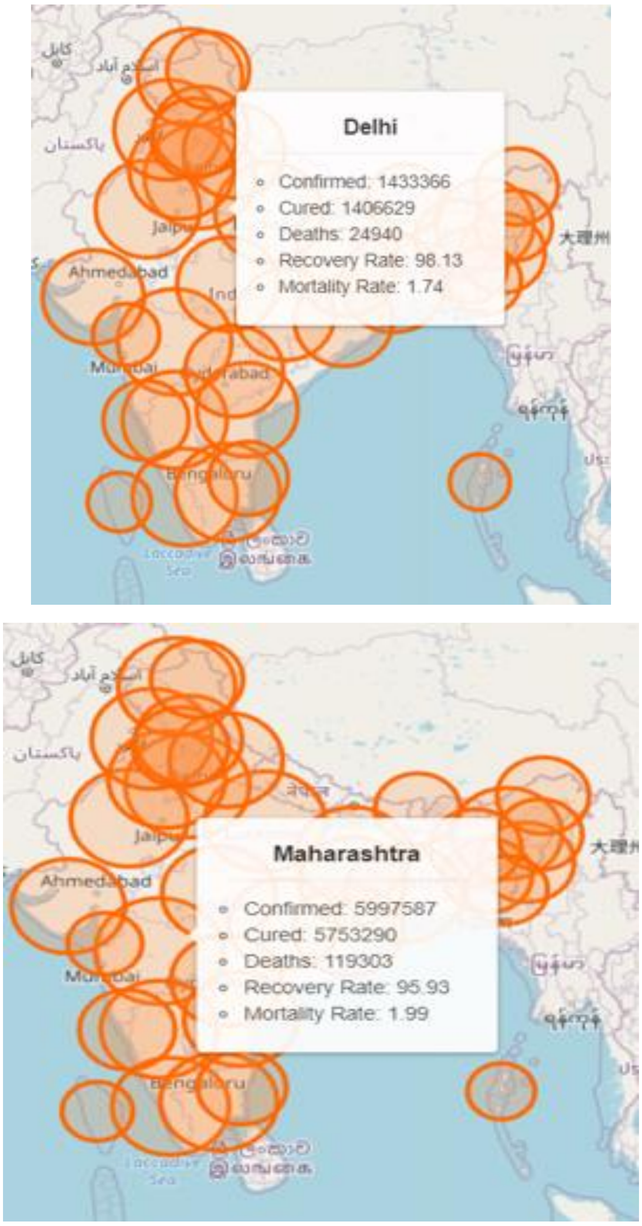


Figure 3.14: Visualization of covid-19 cases in Delhi and Maharashtra.

The figure 3.15 shows the trend of recovery rate of covid-19 cases from March 2020 till May 2021.

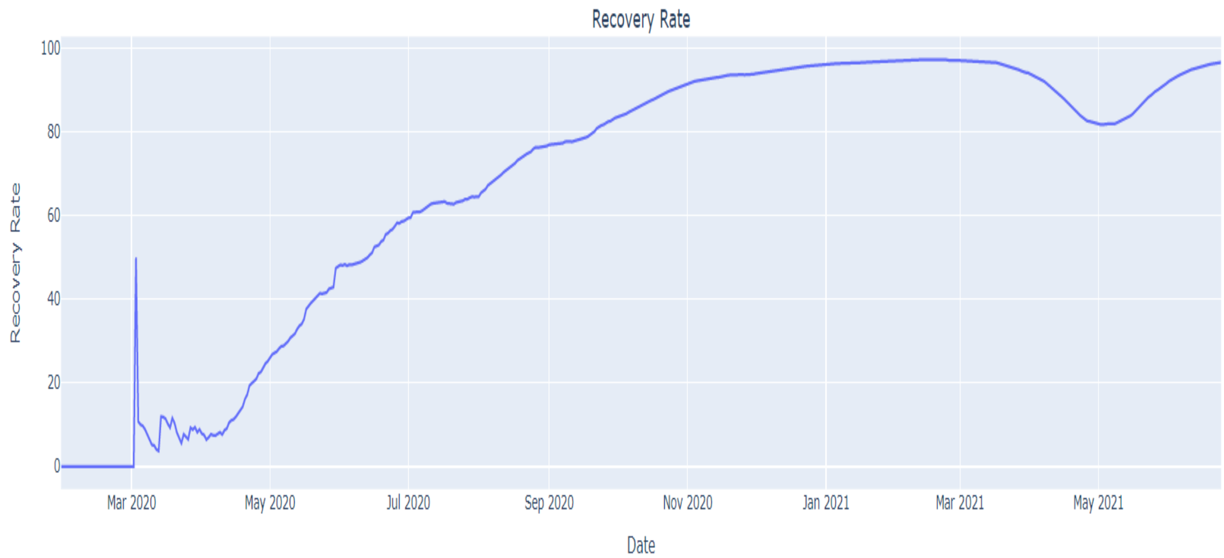


Figure 3.15: Visualization of Recovery rate

The figure 3.16 shows the trend of Mortality rate of covid-19 cases from March 2020 till May 2021.

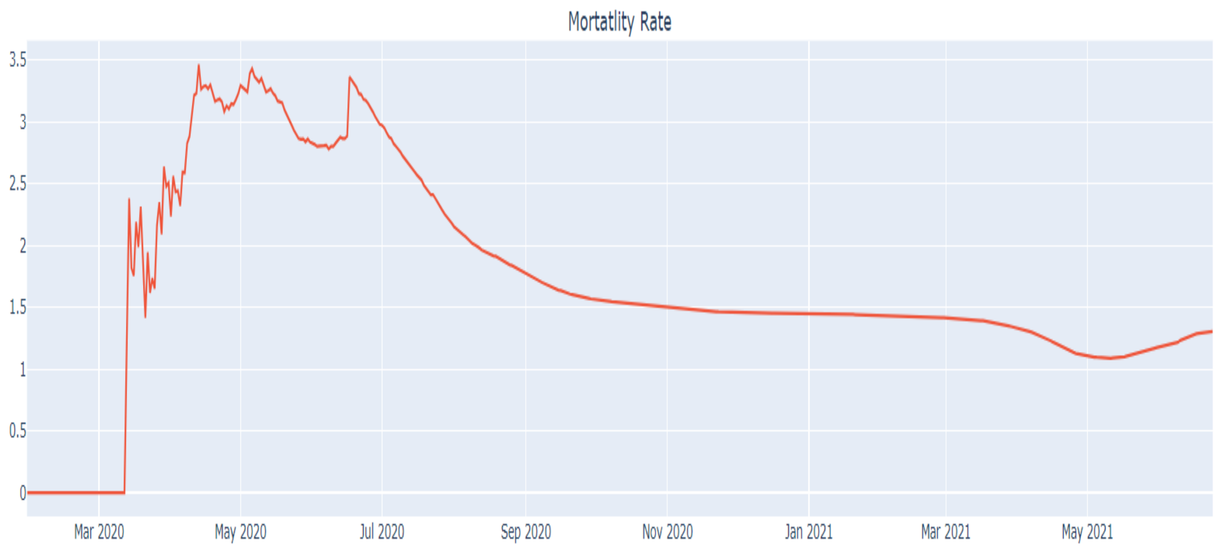


Figure 3.16: Visualization of Mortality rate

The figure 3.17 shows the presence of covid-19 cases among different age groups out of which, it is concluded that the age group of 20-29 years is more prone to covid-19.

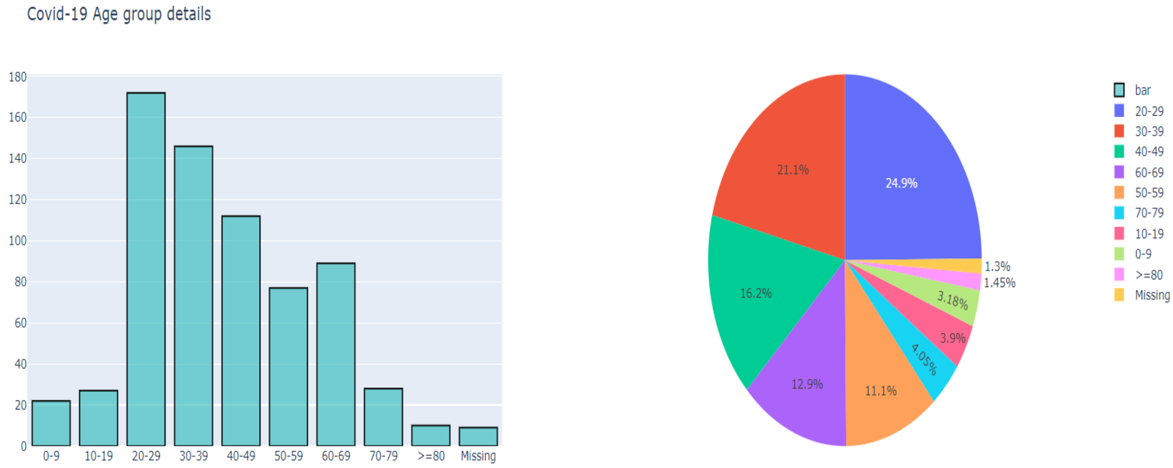


Figure 3.17: Presence of covid-19 cases among different age groups.

3.5.2 Analysis of ICMR Lab Testing and state wise Testing across India:

The dataset for ICMR lab testing consists of 267 entries with 6 columns in it which describes the location and availability of labs approved by icmr for testing. The figure 3.18 shows the geospatial visualization of different testing labs available across different regions of India.

Figure 3.18: Visualization of testing labs across India.

The figure 3.19 shows that Uttar Pradesh has conducted large number of covid-19 testing as compared to other states of India.

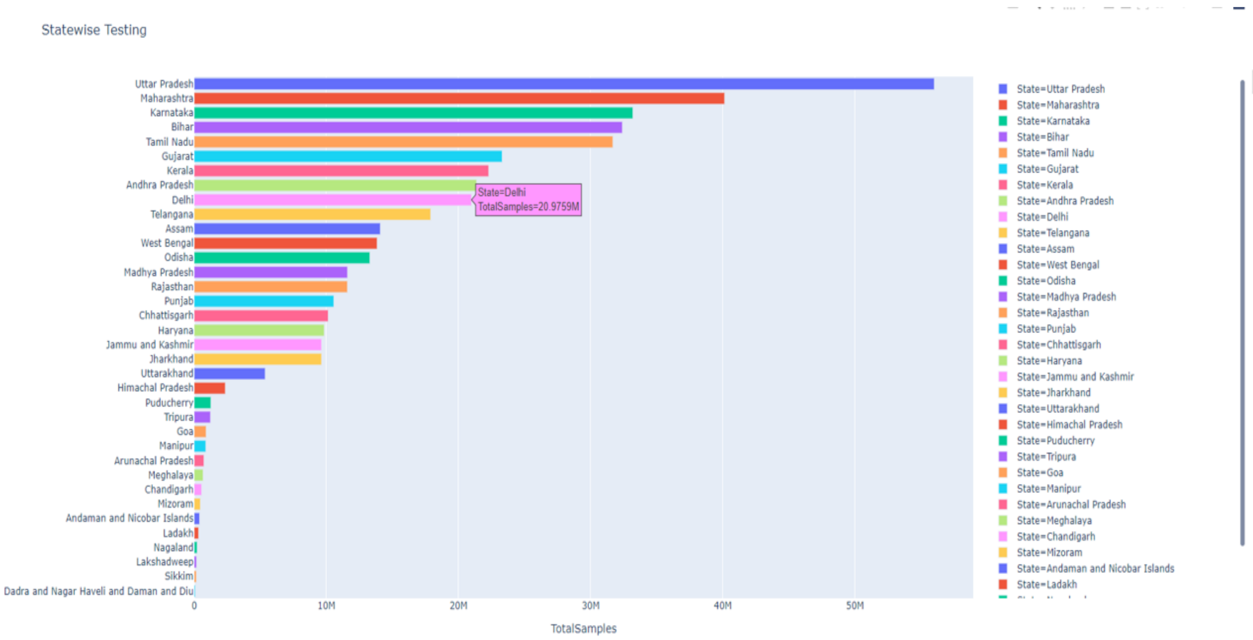


Figure 3.19: Visualization of State-wise testing across different regions of India

The figure 3.20 shows the trend of covid-19 cases in Delhi and Maharashtra during 2nd wave.

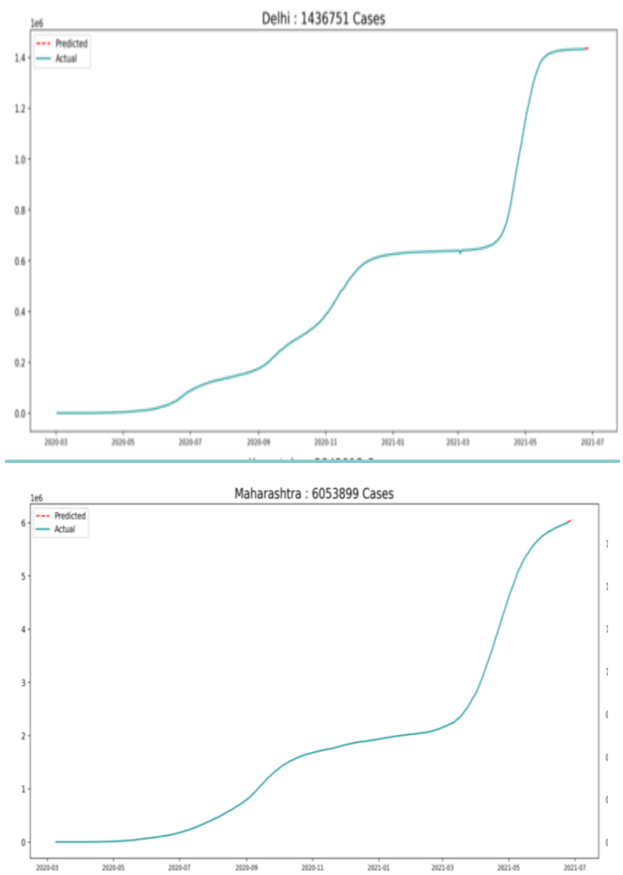


Figure 3.20: Trend of covid-19 cases in Delhi and Maharashtra.

3.5.3 Analyzing the number of hospital beds available across India:

The dataset for Hospital bed locations consists of 14 columns in it.

The figure 3.21 shows the availability of hospitals and beds across different regions of India.

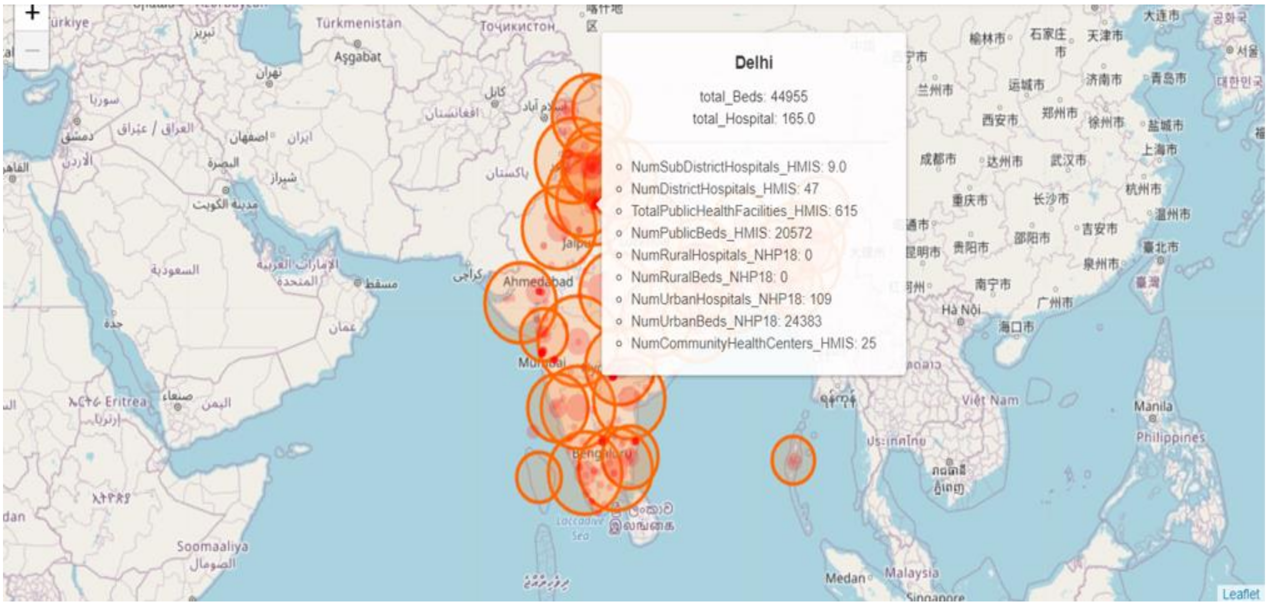


Figure 3.21: Geospatial visualization of availability of hospital beds across India.

3.5.4 Analysis of Vaccination Process:

The figure 3.22 shows the number of individuals vaccinated across different states of India.

	State	Total Individuals Vaccinated
13	India	244944880.0
34	Uttar Pradesh	23918186.0
21	Maharashtra	23663198.0
29	Rajasthan	18905925.0
10	Gujarat	18475269.0
16	Karnataka	17126371.0
20	Madhya Pradesh	15775623.0
36	West Bengal	15536230.0
4	Bihar	12863306.0
1	Andhra Pradesh	11331864.0
31	Tamil Nadu	11289696.0
17	Kerala	10116957.0
26	Odisha	8874332.0
32	Telangana	8233390.0
11	Haryana	6863194.0
6	Chhattisgarh	5541241.0
3	Assam	5233285.0

Figure 3.22: Vaccination progress across different regions of India.

Figure 3.23 demonstrates the total number of individuals vaccinated from day to day across different states of India.

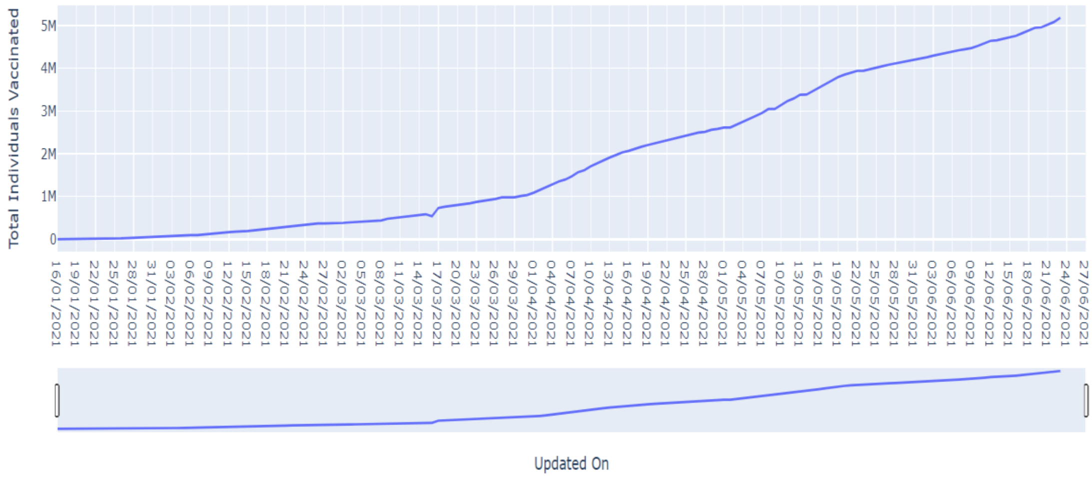


Figure 3.23: Total individuals vaccinated from day to day

The figure 3.24 shows the status of vaccination progress in Delhi region.

```
[289] delhi = df_vc[df_vc["State"]=="Delhi"]
      delhi.head()
```

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Individuals Vaccinated)	Transgender(Individuals Vaccinated)	Total Covaxin Administered	Total CoviShield Administered
67	16/01/2021	Delhi	892.0	91.0	81.0	892.0	0.0	525.0	367.0	0.0	33.0	859.0
68	17/01/2021	Delhi	1005.0	288.0	81.0	1005.0	0.0	603.0	402.0	0.0	36.0	969.0
69	18/01/2021	Delhi	1366.0	347.0	82.0	1366.0	0.0	840.0	526.0	0.0	46.0	1320.0
70	19/01/2021	Delhi	3041.0	357.0	83.0	3041.0	0.0	1837.0	1204.0	0.0	164.0	2877.0
71	20/01/2021	Delhi	3041.0	456.0	84.0	3041.0	0.0	1837.0	1204.0	0.0	164.0	2877.0

Figure 3.24: Delhi vaccination progress

The figure 3.25 depicts that 41.2% of females are vaccinated, 58.8% of males are vaccinated and 0.0191% of transgender are vaccinated in Delhi.

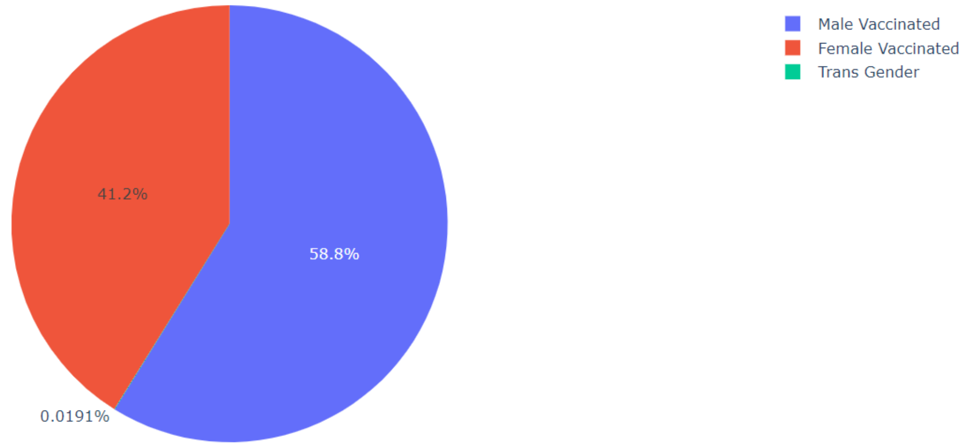


Figure 3.25: Gender wise analysis of vaccination

The figure 3.26 demonstrates that 29.6% of the people are vaccinated with covaxin and 70.4% of the people are vaccinated with covishield in delhi.

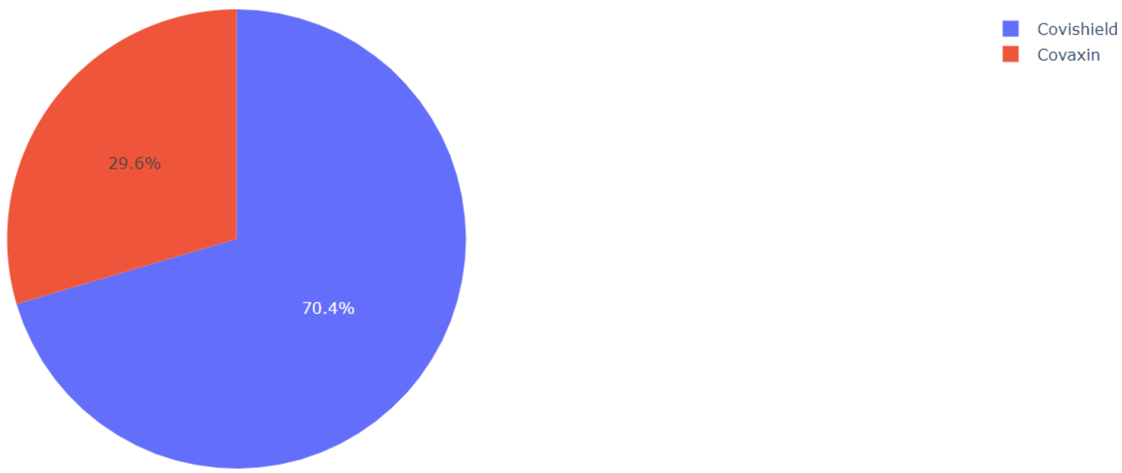


Figure 3.26: Covaxin and Covishield vaccination ratio

The figure 3.27 demonstrates that 21.6% of the people are partially vaccinated while 78.4% of the people are fully vaccinated in delhi.

Figure 3.27: First and Second dosage vaccination ratio

The figure 3.28 shows the status of vaccination progress in Maharashtra region.

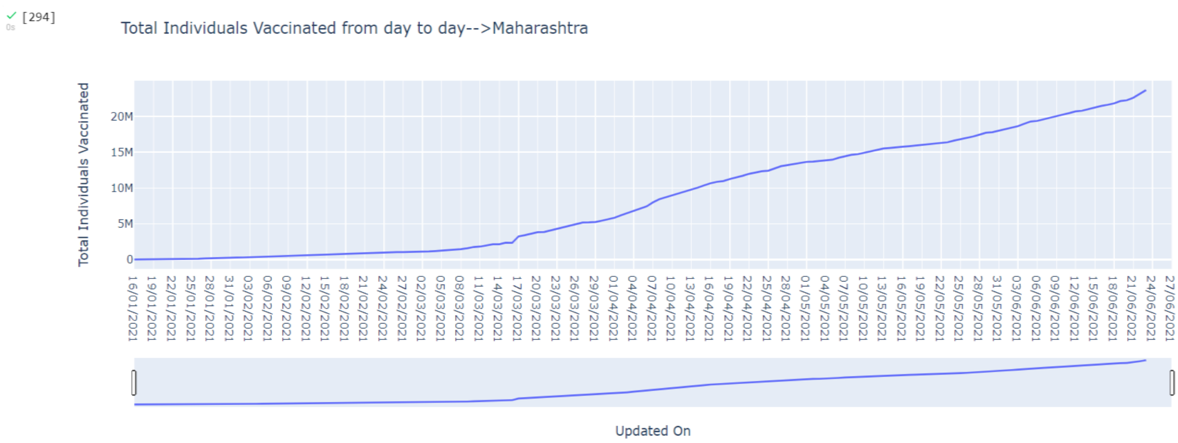


Figure 3.28: Total individuals vaccinated in Maharashtra.

The figure 3.29 shows that 46.2% of females are vaccinated, 53.7% of males are vaccinated and 0.0135% of transgender are vaccinated in Maharashtra.

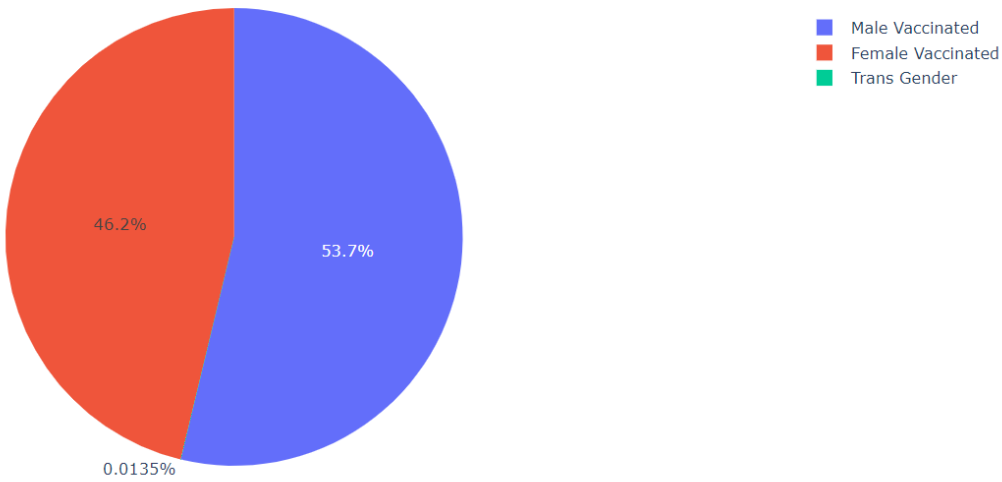


Figure 3.29: Gender wise analysis of vaccination

The figures 3.30 demonstrates that 11.5% of the people are vaccinated with covaxin and 88.5% of the people are vaccinated with covishield in Maharashtra.

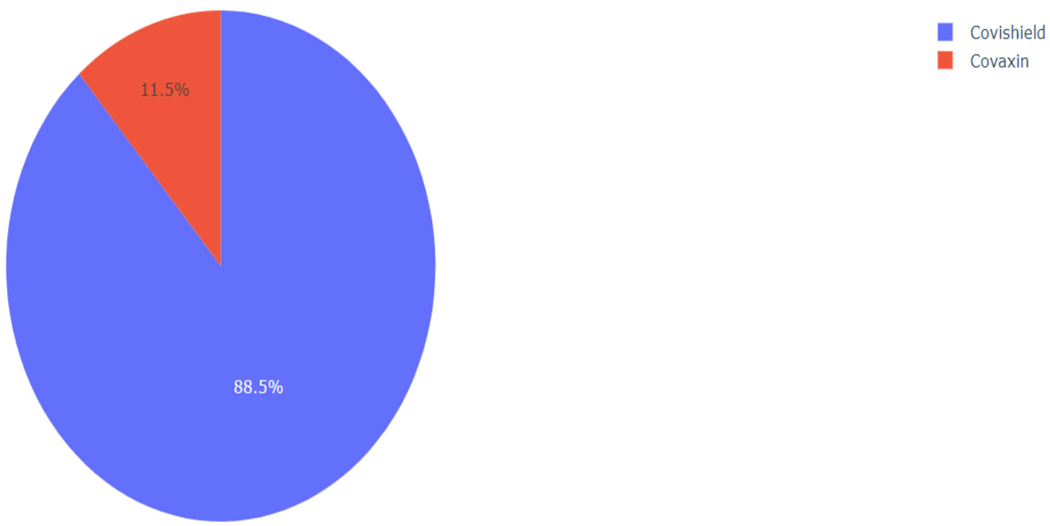


Figure 3.30: Covaxin and Covishield vaccination ratio

The figure 3.31 demonstrates that 17.7% of the people are partially vaccinated while 82.3% of the people are fully vaccinated in Maharashtra.

Figure 3.31: First and Second dosage vaccination ratio

3.6 Employing Transfer learning algorithms and Model Development:

3.6.1 Image Augmentation:

Image augmentation is a method of modifying existing images in order to generate additional data for the model training process. In other words, it is the technique of artificially increasing the dataset available for training a deep learning model.

In the figure 3.32, only the original image is shown on the left, and the rest of the images are derived from the original training image by augmenting it.

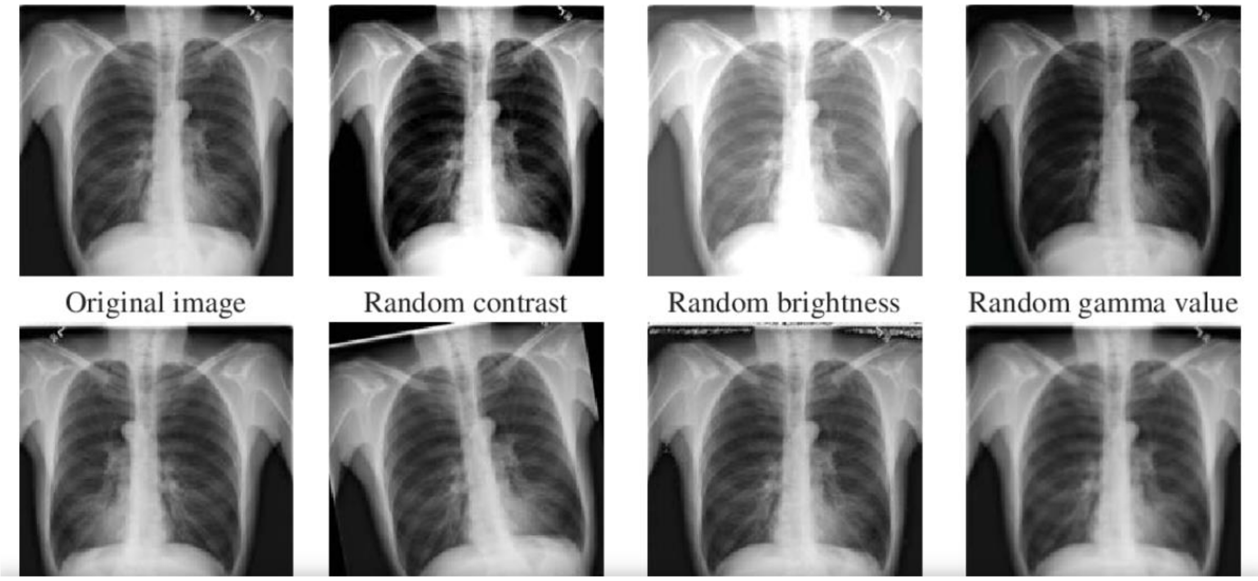


Figure 3.32: Image Augmentation

We don't need to manually acquire these photos because they all are generated from the training data. This increases the training sample without gathering the information. All of the photos will have the same label, which is the label of the original image that was used to create them.

3.6.2 Training and Testing Data Split:

The dataset consists of approximately 1694 images of healthy and infected lungs in the form of CT scans and chest Xray’s. Out of total data, 80% of the data is used for training the data and 20% of the data is used for testing the data.

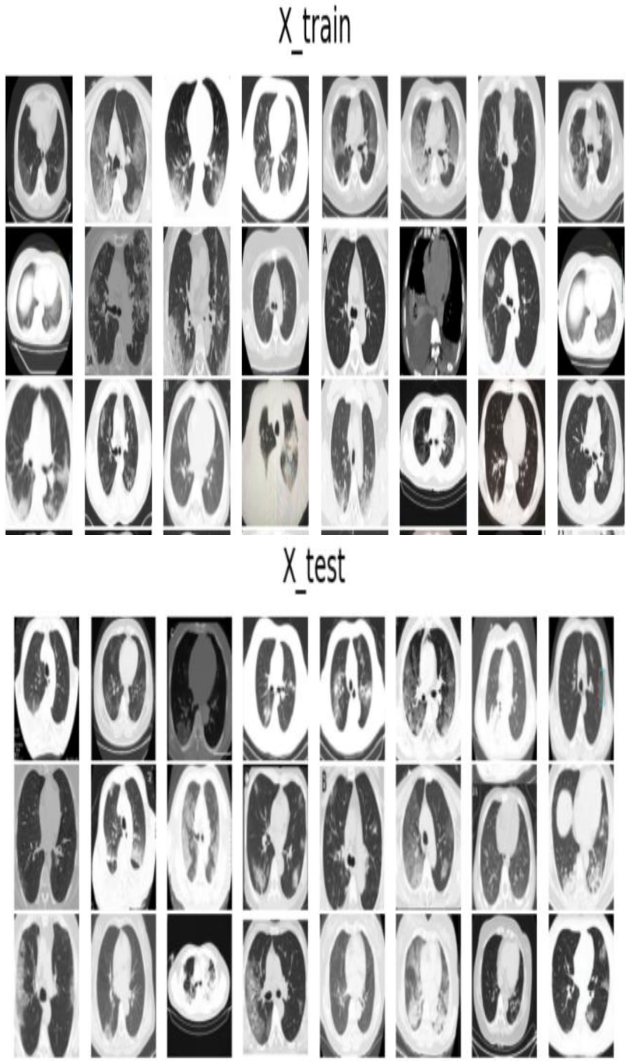


Figure 3.33: Training and testing data split of CT scan.

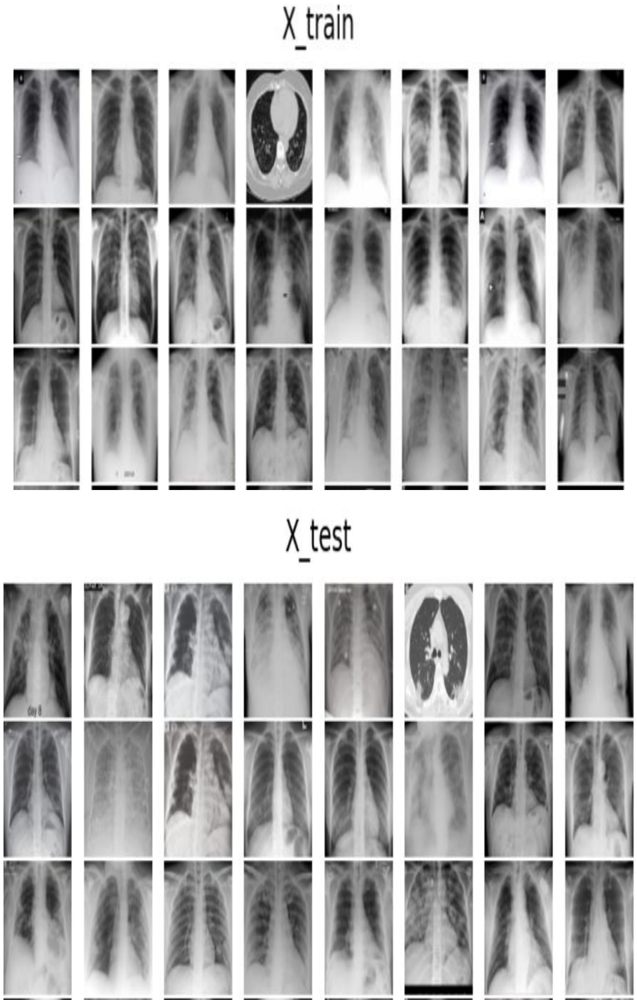


Figure 3.34: Training and testing data split of Chest Xray's.

3.6.3 Implementing various Transfer Learning Algorithms & Model Development:

In Transfer learning, a machine uses knowledge from a previous task to improve prediction about a new task. Basically, transfer learning refers to the reuse of a previously trained model on a new task. The information of an already trained machine learning model is transferred to a separate but closely related task. It has several advantages, the most significant of which are shorter training times, greater neural network performance (in most cases), and the lack of a vast amount of data.

Imagenet is a project aimed on classifying and categorizing photos into over 22,000 separate object categories for the objectives of computer vision research. The goal of this image classification challenge is to build a model that can accurately classify 1,000 various object categories from an input image. Models are trained on 1355 training photos, followed 339 testing images. The dataset that we have taken in order to implement this research project has two labels 0 and 1 out of which 0 signifies the presence of covid-19 in a person and 1 signifies the absence of covid-19 in a person.

Figure 3.35: Visualization of neural networks

3.6.3.1 VGG 16:

Simonyan and Zisserman proposed the VGG network architecture. The simplicity of this network is shown by the use of only 3x3 convolutional layers layered on top of each other in increasing depth. Max pooling is used to shrink the volume size. The SoftMax classifier is therefore followed by two fully connected layers, each with 4,096 nodes. The number "16" refers to the number of weight layers in a network. VGGNet has two major limitations, first is it’s training process is extremely slow and the other one is that it’s network architectural weights are relatively large. The accuracy achieved by this model in the case of Chest x-rays and CT-scans are 87% and 75% respectively.

The figure 3.36 shows the building of a Vgg_ct model.

```
# Building Vgg_ct Model
vggModel = VGG19(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = vggModel.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=vggModel.input, res=res)

for layer in vggModel.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)

train_aug = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
)
```

Figure 3.36: Building Vgg_ct model

The figure 3.37 shows the Saving and loading of a Vgg_ct model.

```
[ ] # Saving and loading the Vgg_ct Model
    model.save('vgg_ct.h5')
    model.save_weights('vgg_weights_ct.hdf5')
```

```
[ ]
    model = load_model('vgg_ct.h5')
```

Figure 3.37: Saving and loading Vgg_ct Model

The figure 3.38 shows the building of a Vgg_chest model.

```
# Building a Vgg_chest Model
vggModel = VGG19(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = vggModel.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=vggModel.input, res=res)

for layer in vggModel.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)
```

Figure 3.38: Building Vgg_chest model

The figure 3.39 shows the saving and loading of a Vgg_chest model.

```
# Saving and Loading the Vgg_chest Model
model.save('vgg_chest.h5')
```

```
[ ] model.save_weights('vggweights_chest.hdf5')
```

```
[ ] model = load_model('vgg_chest.h5')
```

Figure 3.39: Saving and Loading Vgg_chest model

3.6.3.2 ResNet 50:

ResNet, on the other hand, is a kind of "exotic architecture" based on micro-architecture modules also referred as "network-in-network architectures".The set of "building blocks" required to construct the network is known as micro architecture. The macro-architecture is made up of a set of micro-architecture building pieces . Here,"50" in ResNet50 refers to the number of weight layers. Despite the fact that ResNet is much deeper than VGG16, the model size is significantly smaller due to the use of global average pooling rather than fully-connected layers – thus, the model size for ResNet50 is just 102MB.

The accuracy achieved by this model in the case of Chest x-rays and CT-scans are 68% and 67% respectively.

The figure 3.40 shows the building of a ResNet_ct model.

```
# Building a Resnet_ct Model
resnet = ResNet50(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = resnet.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=resnet.input, res=res)

for layer in resnet.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)
```

Figure 3.40: Building of a ResNet_ct model.

The figure 3.41 shows the saving and loading of a ResNet_ct model.

```
# Saving and loading resnet_ct Model
model.save('resnet_ct.h5')
model.save_weights('resnet_weights_ct.hdf5')

/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py
  layer_config = serialize_layer_fn(layer)

[ ] model = load_model('resnet_ct.h5')
```

Figure 3.41: Saving and Loading resNet_ct model.

The figure 3.42 shows the building of a ResNet_chest model.

```
# Building a Resnet_chest model
resnet = ResNet50(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = resnet.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=resnet.input, res=res)

for layer in res.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)

train_aug = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
)
```

Figure 3.42: Building of a ResNet_chest model.

The figure 3.43 shows the saving and loading of a ResNet_chest model.

```
[ ] #Saving and loading resnet_chest model.
    model.save('resnet_chest.h5')
    model.save_weights('resnetweights_chest.hdf5')
    model = load_model('resnet_chest.h5')

    /usr/local/lib/python3.7/dist-packages/keras/engine/f
      layer_config = serialize_layer_fn(layer)
```

Figure 3.43: Saving and Loading resNet_chest model

3.6.3.3: Inception V3:

Szegedy et al. were the first to introduce the "Inception" micro-architecture, which has 48 layers.

The inception module's purpose is to operate as a "multi-level feature extractor" by computing 1x1, 3x3, and 5x5 convolutions within the same network module; the output of these filters is then layered along the channel dimension before being sent into the next layer.

The original version of this architecture was known as GoogLeNet, but later iterations have simply been referred to as Inception vN, where N corresponds to the Google version number.The weights for Inception V3 are 96MB, which is less than VGG and ResNet.

The accuracy achieved by this model in the case of Chest x-rays and CT-scans are 75% and 81% respectively.

The figure 3.44 shows the building of a InceptionV3_ct model.

```
inception = InceptionV3(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = inception.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=inception.input, res=res)

for layer in inception.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)

Downloading data from https://storage.googleapis.com/tensorflow/keras-applicat
87916544/87910968 [==============================] - 1s 0us/step
87924736/87910968 [==============================] - 1s 0us/step
```

Figure 3.44: Building of a InceptionV3_ct model.

The figure 3.45 shows the saving and loading of a InceptionV3_ct model.

```
# Saving and loading inception_ct model
model.save('inception_ct.h5')
model.save_weights('inception_weights_ct.hdf5')

[ ]
model = load_model('inception_ct.h5')
```

Figure 3.45: Saving and Loading InceptionV3_ct model

The figure 3.46 shows the building of a InceptionV3_chest model.

```
#Building a InceptionV3_chest model
inception = InceptionV3(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = inception.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

mo1 = Model(inputs=inception.input, res=res)

for layer in inception.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)

train_aug = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    horizontal_flip=True
)
```

Figure 3.46: Building of a InceptionV3_chest model.

The figure 3.47 shows the saving and loading of a InceptionV3_chest model.

```
# Saving and loading inceptionv3_chest model
model.save('inceptionv3_chest.h5')
```

```
[ ] model.save_weights('inceptionv3_chest.hdf5')
```

```
[ ] model = load_model('inceptionv3_chest.h5')
```

Figure 3.47: Saving and Loading InceptionV3_chest model.

3.6.3.4: Xception:

Xception was recommended by none other than François Chollet, the Keras library's founder and major maintainer. It has a total of 71 layers. Xception is an Inception architecture extension that uses depth wise separable convolutions to replace the regular Inception modules. At only 91MB, Xception has the smallest weight serialisation.

The accuracy achieved by this model in the case of Chest x-rays and CT-scans are 89% and 81% respectively.

The figure 3.48 shows the building of a Xception_ct model.

```
# Building a xception_ct Model
xception = Xception(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = xception.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=xception.input, res=res)

for layer in xception.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)
```

Figure 3.48: Building of a Xception_ct model.

The figure 3.49 shows the saving and loading of a Xception_ct model.

```
# Saving and loading xception_ct model
model.save('xception_ct.h5')
model.save_weights('xception_weights_ct.hdf5')

/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py:
  layer_config = serialize_layer_fn(layer)

[ ]
model = load_model('xception_ct.h5')
```

Figure 3.49: Saving and Loading Xception_ct model

The figure 3.50 shows the building of a Xception_chest model.

```
# Building a Xception_chest Model
xception = Xception(weights="imagenet", include_top=False,
    input_tensor=Input(shape=(224, 224, 3)))

res = xception.output
res = Flatten(name="flatten")(res)
res = Dropout(0.5)(res)
res = Dense(2, activation="softmax")(res)

m1 = Model(inputs=xception.input, res=res)

for layer in xception.layers:
    layer.trainable = False

m1.compile(
        loss='categorical_crossentropy',
        optimizer='adam',
        metrics=['accuracy']
)
```

Figure 3.50: Building of a Xception_chest model.

The figure 3.51 shows the saving and loading of a Xception_chest model.

```
# Saving and loading xception_chest model
model.save('xception_chest.h5')
model.save_weights('xception_weights_chest.hdf5')
```

```
/usr/local/lib/python3.7/dist-packages/keras/engine/functional.py:
  layer_config = serialize_layer_fn(layer)
```

```
[ ]
model = load_model('xception_chest.h5')
```

Figure 3.51: Saving and Loading Xception_chest model.

3.6.3.5: Convolutional Neural Networks (CNN):

A Convolutional Neural Network (ConvNet/CNN) is a deep learning algorithm that can take an input image and assign priority to different regions of it, allowing it to differentiate between them. A ConvNet takes significantly less pre-processing than conventional classification techniques. The architecture of a ConvNet is very similar to the connectivity pattern of Neurons in the brain. Using the right filters, a ConvNet may successfully capture the spatial and temporal relationships in an image.

The accuracy achieved by this model in the case of Chest x-rays and CT-scans are 65% and 58% respectively.

The figure 3.52 shows the building of a cnn_ct model.

```
[ ]  model = Sequential()
     model.add(Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(224,224,3)))
     model.add(Conv2D(64,(3,3),activation='relu'))
     model.add(MaxPooling2D(pool_size=(2,2)))
     model.add(Dropout(0.25))

     model.add(Conv2D(64,(3,3),activation='relu'))
     model.add(MaxPooling2D(pool_size=(2,2)))
     model.add(Dropout(0.25))

     model.add(Conv2D(128,(3,3),activation='relu'))
     model.add(MaxPooling2D(pool_size=(2,2)))
     model.add(Dropout(0.25))

     model.add(Flatten())
     model.add(Dense(64,activation='relu'))
     model.add(Dropout(0.5))
     model.add(Dense(1,activation='sigmoid'))

     model.compile(loss=keras.losses.binary_crossentropy,optimizer='adam',metrics=['accuracy'])
```

Figure 3.52: Building of a cnn_ct model.

The figure 3.53 shows the saving and loading of a cnn_ct model.

```
[ ]  model.save("cnn_ct.h5")

[ ]  model.evaluate_generator(train_generator)

     /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
       """Entry point for launching an IPython kernel.
     [0.6931294202804565, 0.5035211443901062]

[ ]  model.evaluate_generator(validation_generator)

     /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
       """Entry point for launching an IPython kernel.
     [0.6931156516075134, 0.5357142686843872]

[ ]  model = load_model('cnn_ct.h5')
```

Figure 3.53: Saving and Loading cnn_ct model.

The figure 3.54 shows the building of a cnn_chest model.

```
[ ] model = Sequential()
    model.add(Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(224,224,3)))
    model.add(Conv2D(64,(3,3),activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(64,(3,3),activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))

    model.add(Conv2D(128,(3,3),activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(Dropout(0.25))

    model.add(Flatten())
    model.add(Dense(64,activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1,activation='sigmoid'))

    model.compile(loss=keras.losses.binary_crossentropy,optimizer='adam',metrics=['accuracy'])
```

Figure 3.54: Building of a cnn_chest model.

The figure 3.55 shows the saving and loading of a cnn_chest model.

```
[ ] model.save("cnn_chest.h5")

[ ] model.evaluate_generator(train_generator)

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
      """Entry point for launching an IPython kernel.
    [0.4360455870628357, 0.7892857193946838]

[ ] model.evaluate_generator(validation_generator)

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
      """Entry point for launching an IPython kernel.
    [0.4658704400062561, 0.7142857313156128]

[ ] model = load_model('cnn_chest.h5')
```

Figure 3.55: Saving and Loading cnn_chest model.

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Analysis of model's trained:

Now after applying all the Transfer Learning Techniques on the respective deep learning-based models, different accuracy score, precision score, recall score and f-score have been achieved. Following is the performance analysis of different Transfer learning algorithms on the basis of different performance parameters. The methods for comparing outcomes that we employed are based on a statistical and analytical approach. Here, the performance evaluation is done on the basis of confusion matrix and its corresponding parameters.

4.1.1 Confusion Matrix:

A confusion matrix computes the summary of prediction results on a classification problem. It is a table that's often required to describe the performance of a classification model on a group of test values which are known. This matrix compares the actual values with those predicted by the model. Basically, the confusion matrix shows the performance of the classification model.

4.1.2 Parameters:

1.Positive(P): When the observation is positive (for example: is an apple)

2.Negative(N): When the observation is not positive (for example: is not an apple)

3.True Positive (TP): When the observation is positive, and is predicted to be positive.

4.False Negative (FN): When the observation is positive, and is predicted to be negative.

5.True Negative (TN): When the observation is negative, and is predicted to be negative.

6.False Positive (FP): When the observation is negative, and is predicted to be positive.

7.Recall Score: It is defined as the ratio of total number of correctly classified positive examples to the total number of positive examples.

$$\boxed{\textbf{RECALL = TP / TP + FN}}$$

8.Precision/Positive predictive value: It is defined as the ratio of total number of correctly classified positive examples to the total number of predicted positive examples.

$$\boxed{\textbf{PRECISION = TP / TP + FP}}$$

9. F-Measure: F-measure is calculated by computing the harmonic mean instead of arithmetic mean as it increases the extreme values. F-measure is always nearer to the smaller value of precision or recall score.

F-MEASURE = 2*RECALL*PRECISION / RECALL + PRECISION

10. Accuracy: The accuracy of the classification-based model is one of the parameters used to evaluate it. The percentage of correct predictions made by the model is referred to as accuracy.

ACCURACY = TP + TN / FN+TP+TN+FP

The findings of all research carried out in this project are presented in this portion of our project report. Experiments were conducted using ten CNN architectures: Visual Geometric Group-16 (VGG-16), InceptionV3, Xception, Residual Network-50 (ResNet-50) and Convolutional neural network (CNN). The loss and accuracy curves, confusion matrices, and Receiver Operating Characteristic curves show some performance similarities as well as the significant variations among them. Hence, the analysis of 10 DL based models are stated below:

Here, the figure 4.1 depicts the ROC curve of Visual Geometry Group that is VGG16 CNN model. This model is designed to analyse the CT scan images of healthy and infected patients using VGG16 TL technique.

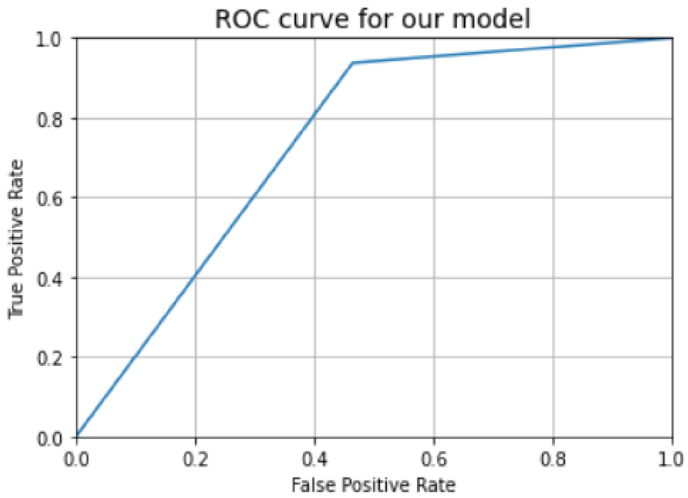


Figure 4.1: ROC curve for Vgg_ct model.

The figure 4.2 depicts the confusion matrix of VGG16_CT CNN model with normalized values.

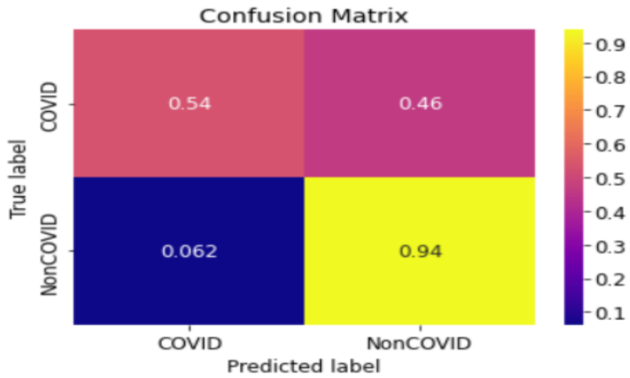


Figure 4.2: Confusion Matrix for Vgg_ct model.

The figure 4.3 shows the performance analysis report of VGG16_CT CNN model that has all the details about all the performance metrics that were analysed like precision score, recall score,f1-score and accuracy score. It is analysed that VGG16_CT model has an accuracy of 75% in predicting the covid-19 infection using CT scans as input.

	precision	recall	f1-score	support
0	0.88	0.54	0.67	71
1	0.69	0.94	0.80	80
accuracy			0.75	151
macro avg	0.79	0.74	0.73	151
weighted avg	0.78	0.75	0.74	151

Figure 4.3: Performance Analysis report for Vgg_ct model.

Here, the figure 4.4 depicts the ROC curve of Visual Geometry Group that is VGG16 CNN model. This model is designed to analyse the Chest Xray images of healthy and infected patients using VGG16 TL technique.

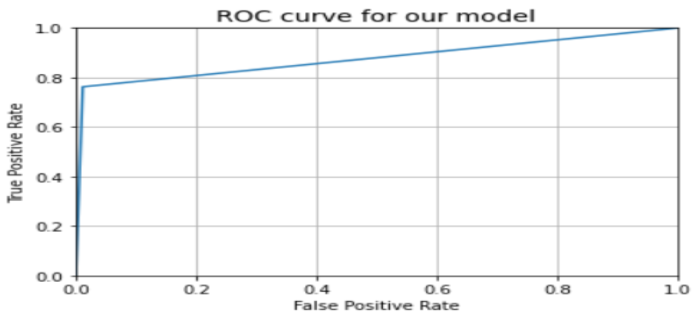


Figure 4.4: ROC curve for Vgg_chest model.

The figure 4.5 depicts the confusion matrix of VGG16_CHEST CNN model with normalized values.

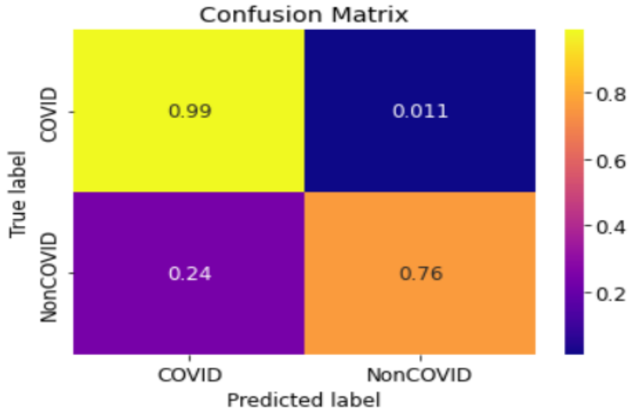


Figure 4.5: Confusion Matrix for Vgg_chest model.

The figure 4.6 shows the performance analysis report of VGG16_CHEST CNN model. It is analysed that VGG16_CHEST model has an accuracy of 87% in predicting the covid-19 infection using chest x-rays as input.

	precision	recall	f1-score	support
0	0.78	0.99	0.87	88
1	0.99	0.76	0.86	101
accuracy			0.87	189
macro avg	0.89	0.88	0.87	189
weighted avg	0.89	0.87	0.87	189

Figure 4.6: Performance Analysis report for Vgg_chest model.

Here, the figure 4.7 depicts the ROC curve of ResNet_CT CNN model. This model is designed to analyse the CT scan images of healthy and infected patients using ResNet TL technique.

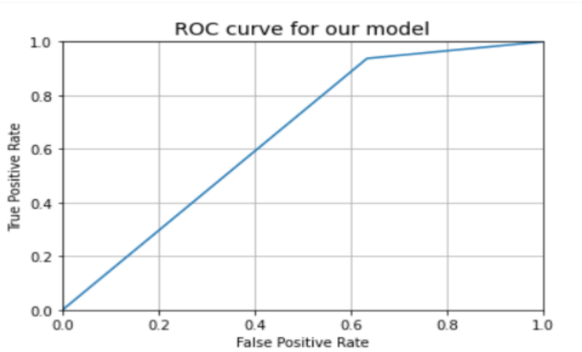


Figure 4.7: ROC curve for ResNet_ct model.

The figure 4.8 depicts the confusion matrix of ResNet_CT CNN model with normalized values.

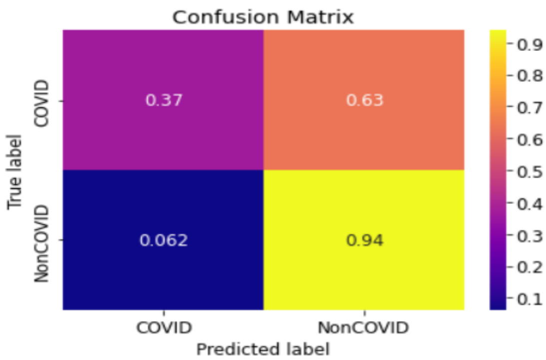


Figure 4.8: Confusion Matrix for ResNet_ct model.

The figure 4.9 shows the performance analysis report of ResNet_CT CNN model. It is analysed that ResNet_CT model has an accuracy of 67% in predicting the covid-19 infection using CT scans as input.

	precision	recall	f1-score	support
0	0.84	0.37	0.51	71
1	0.62	0.94	0.75	80
accuracy			0.67	151
macro avg	0.73	0.65	0.63	151
weighted avg	0.73	0.67	0.64	151

Figure 4.9: Performance Analysis report for ResNet_ct model.

Here, the figure 4.10 depicts the ROC curve of ResNet_CHEST CNN model. This model is designed to analyse the Chest Xray images of healthy and infected patients using ResNet TL technique.

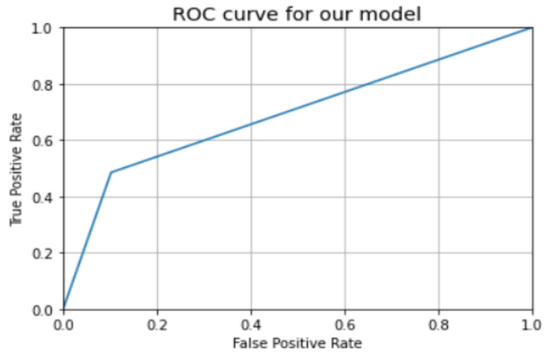


Figure 4.10: ROC curve for ResNet_chest model.

The figure 4.11 depicts the confusion matrix of ResNet_CHEST CNN model with normalized values.

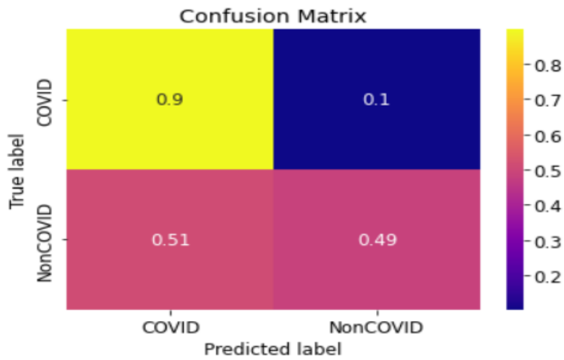


Figure 4.11: Confusion Matrix for ResNet_chest model.

The figure 4.12 shows the performance analysis report of ResNet_CHEST CNN model. It is analysed that ResNet_CHEST model has an accuracy of 67% in predicting the covid-19 infection using Chest xray images as input.

	precision	recall	f1-score	support
0	0.60	0.90	0.72	88
1	0.84	0.49	0.62	101
accuracy			0.68	189
macro avg	0.72	0.69	0.67	189
weighted avg	0.73	0.68	0.67	189

Figure 4.12: Performance Analysis report for ResNet_chest model.

Here, the figure 4.13 depicts the ROC curve of InceptionV3_CT CNN model. This model is designed to analyse the CT scan images of healthy and infected patients using InceptionV3 TL technique.

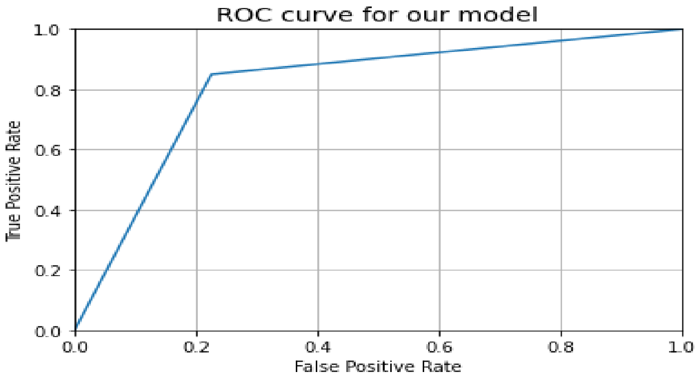


Figure 4.13: ROC curve for InceptionV3_ct model.

The figure 4.14 depicts the confusion matrix of InceptionV3_CT CNN model with normalized values.

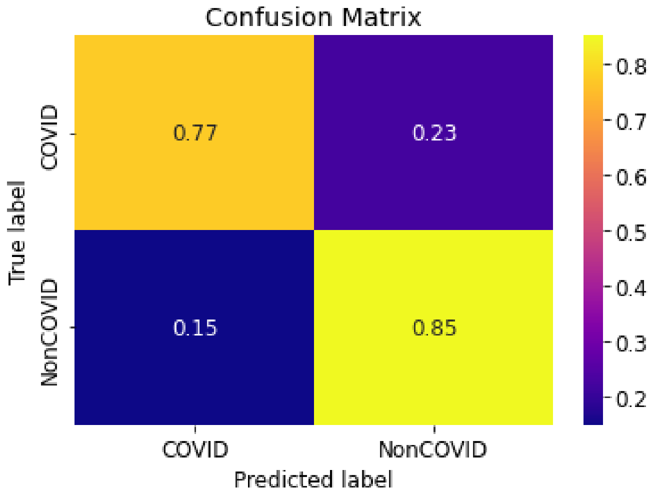


Figure 4.14: Confusion Matrix for InceptionV3_ct model.

The figure 4.15 shows the performance analysis report of InceptionV3_CT CNN model. It is analysed that InceptionV3_CT model has an accuracy of 81% in predicting the covid-19 infection using CT scan images as input.

	precision	recall	f1-score	support
0	0.82	0.77	0.80	71
1	0.81	0.85	0.83	80
accuracy			0.81	151
macro avg	0.82	0.81	0.81	151
weighted avg	0.81	0.81	0.81	151

Figure 4.15: Performance Analysis report for InceptionV3_ct model.

Here, the figure 4.16 depicts the ROC curve of InceptionV3_CHEST CNN model. This model is designed to analyse the Chest Xray images of healthy and infected patients using InceptionV3 TL technique.

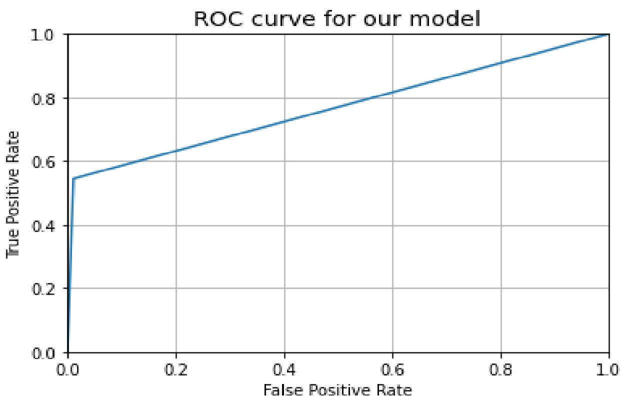


Figure 4.16: ROC curve for InceptionV3_chest model.

The figure 4.17 depicts the confusion matrix of InceptionV3_CHEST CNN model with normalized values.

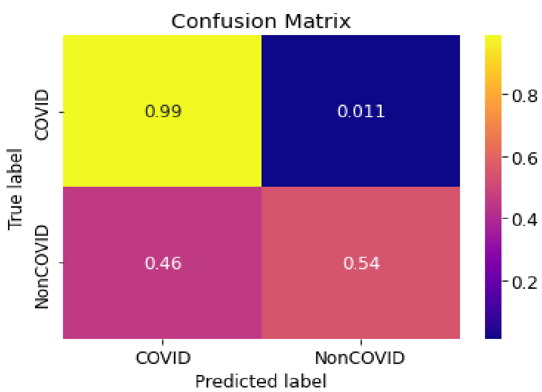


Figure 4.17: Confusion Matrix for InceptionV3_chest model.

The figure 4.18 shows the performance analysis report of InceptionV3_CHEST CNN model. It is analysed that InceptionV3_CHEST model has an accuracy of 75% in predicting the covid-19 infection using Chest xray images as input.

	precision	recall	f1-score	support
0	0.65	0.99	0.79	88
1	0.98	0.54	0.70	101
accuracy			0.75	189
macro avg	0.82	0.77	0.74	189
weighted avg	0.83	0.75	0.74	189

Figure 4.18: Performance Analysis report for InceptionV3_chest model.

Here, the figure 4.19 depicts the ROC curve of Convolutional Neural Network that is CNN model. This model is designed to analyze the CT scan images of healthy and infected patients using CNN technique.

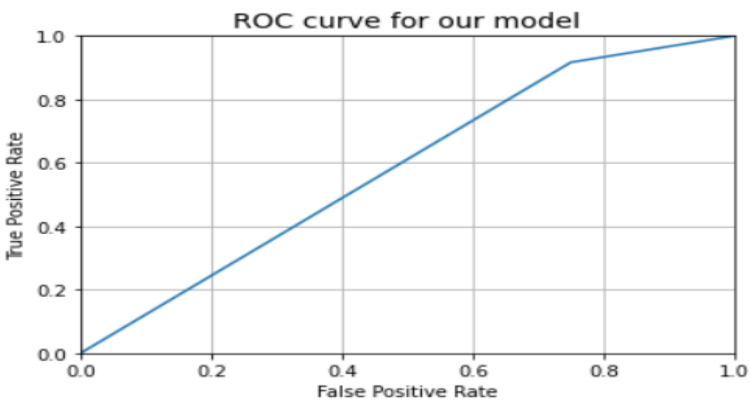


Figure 4.19: ROC curve for cnn_ct model.

The figure 4.20 depicts the confusion matrix of CNN_CT model with normalized values.

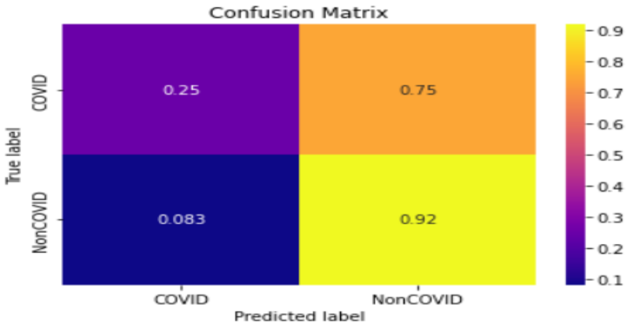


Figure 4.20: Confusion Matrix for cnn_ct model.

The figure 4.21 shows the performance analysis report of CNN_CT model that has all the details about all the performance metrics that were analyzed like precision score, recall score, f1-score and accuracy score. It is analyzed that CNN_CT model has an accuracy of 58% in predicting the covid-19 infection using CT scans as input.

	precision	recall	f1-score	support
0	0.75	0.25	0.38	60
1	0.55	0.92	0.69	60
accuracy			0.58	120
macro avg	0.65	0.58	0.53	120
weighted avg	0.65	0.58	0.53	120

Figure 4.21: Performance Analysis report for cnn_ct model.

Here, the figure 4.22 depicts the ROC curve of Convolutional Neural Networks that is CNN model. This model is designed to analyze the Chest Xray images of healthy and infected patients using CNN technique.

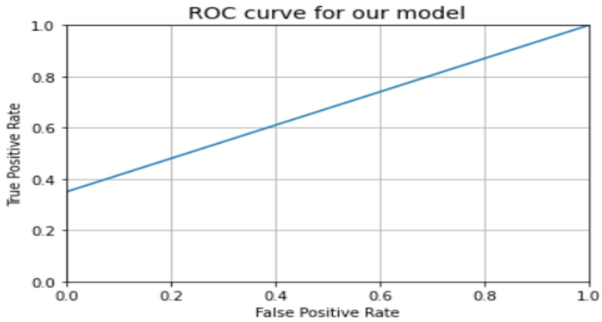


Figure 4.22: ROC curve for cnn_chest model.

The figure 4.23 depicts the confusion matrix of CNN_CHEST model with normalized values.

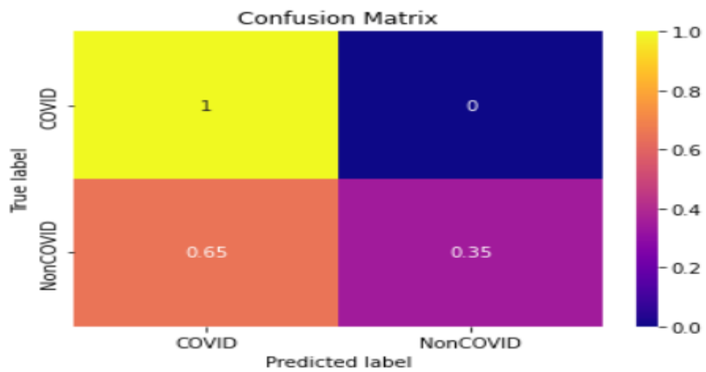


Figure 4.23: Confusion Matrix for cnn_chest model.

The figure 4.24 shows the performance analysis report of CNN_CHEST model. It is analyzed that CNN_CHEST model has an accuracy of 65% in predicting the covid-19 infection using chest x-rays as input.

	precision	recall	f1-score	support
0	0.57	1.00	0.73	52
1	1.00	0.35	0.52	60
accuracy			0.65	112
macro avg	0.79	0.68	0.62	112
weighted avg	0.80	0.65	0.62	112

Figure 4.24: Performance Analysis report for cnn_chest model.

Here, the figure 4.25 depicts the ROC curve of Xception_CT CNN model. This model is designed to analyse the CT scan images of healthy and infected patients using Xception TL technique.

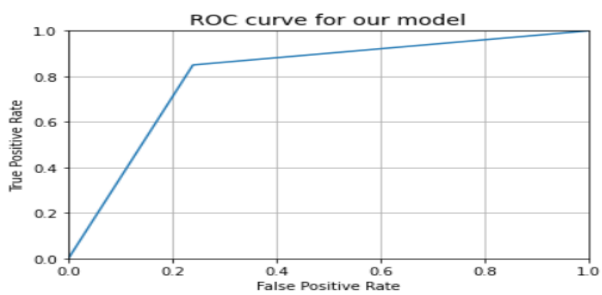


Figure 4.25: ROC curve for Xception_ct model.

The figure 4.26 depicts the confusion matrix of Xception_CT CNN model with normalized values.

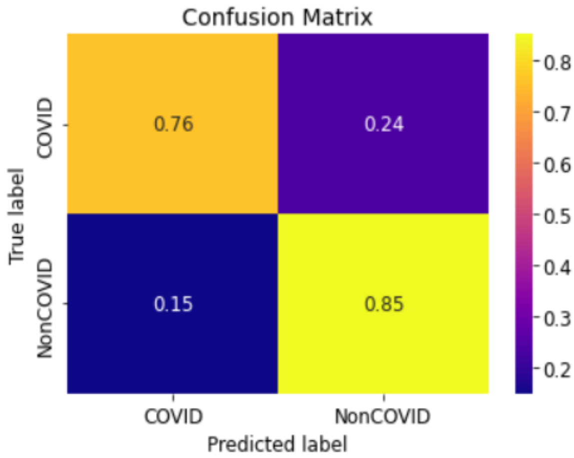


Figure 4.26: Confusion Matrix for Xception_ct model.

The figure 4.27 shows the performance analysis report of Xception_CT CNN model. It is analysed that Xception_CT model has an accuracy of 81% in predicting the covid-19 infection using CT scan images as input.

	precision	recall	f1-score	support
0	0.82	0.76	0.79	71
1	0.80	0.85	0.82	80
accuracy			0.81	151
macro avg	0.81	0.81	0.81	151
weighted avg	0.81	0.81	0.81	151

Figure 4.27: Performance Analysis report for Xception_ct model.

Here, the figure 4.28 depicts the ROC curve of Xception_CHEST CNN model. This model is designed to analyse the CHEST scan images of healthy and infected patients using Xception TL technique.

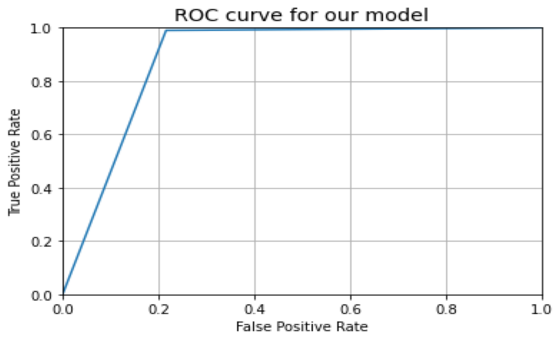


Figure 4.28: ROC curve for Xception_chest model.

The figure 4.29 depicts the confusion matrix of Xception_CHEST CNN model with normalized values.

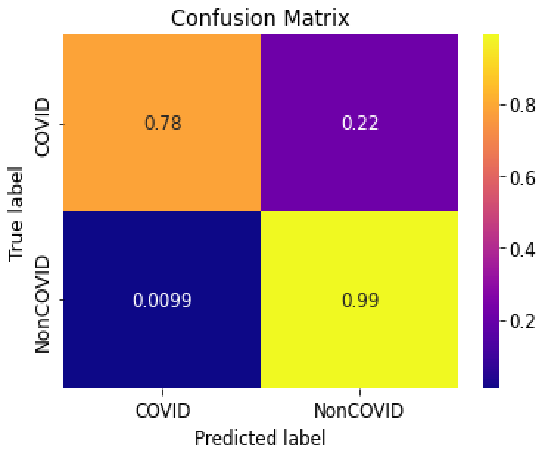


Figure 4.29: Confusion Matrix for Xception_chest model.

The figure 4.30 shows the performance analysis report of Xception_CHEST CNN model. It is analysed that Xception_CHEST model has an accuracy of 89% in predicting the covid-19 infection using Chest xray images as input.

	precision	recall	f1-score	support
0	0.99	0.78	0.87	88
1	0.84	0.99	0.91	101
accuracy			0.89	189
macro avg	0.91	0.89	0.89	189
weighted avg	0.91	0.89	0.89	189

Figure 4.30: Performance Analysis report for Xception_chest model.

Out of all the proposed TL techniques, the Xception technique is the most accurate one to detect the presence of covid-19 in a person. Hence, the model loss and model accuracy for the Xception_CT and Xception_Chest NN models are shown in the graphs below. The models based on Xception TL has the best validation accuracy of 81% and 89%, respectively, with the abovementioned approach, outperforming the other models in terms of accuracy.

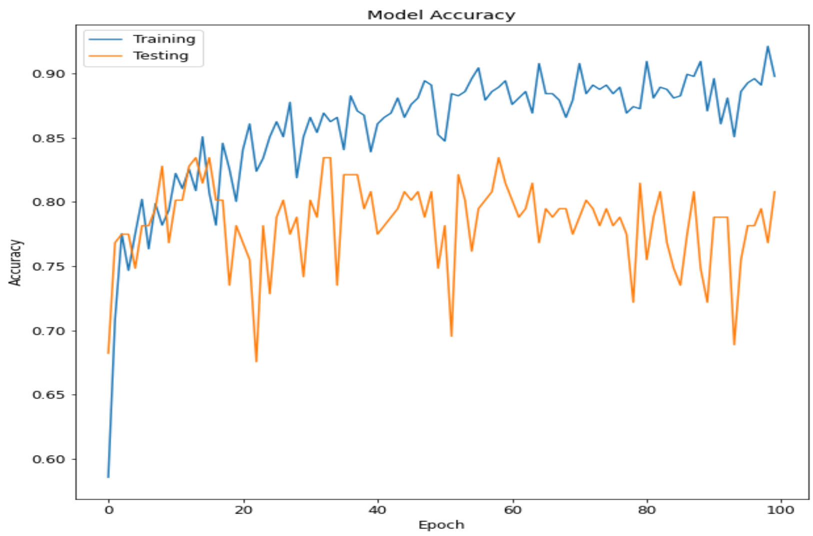


Figure 4.31: Model Accuracy Plot for Xception_ct model.

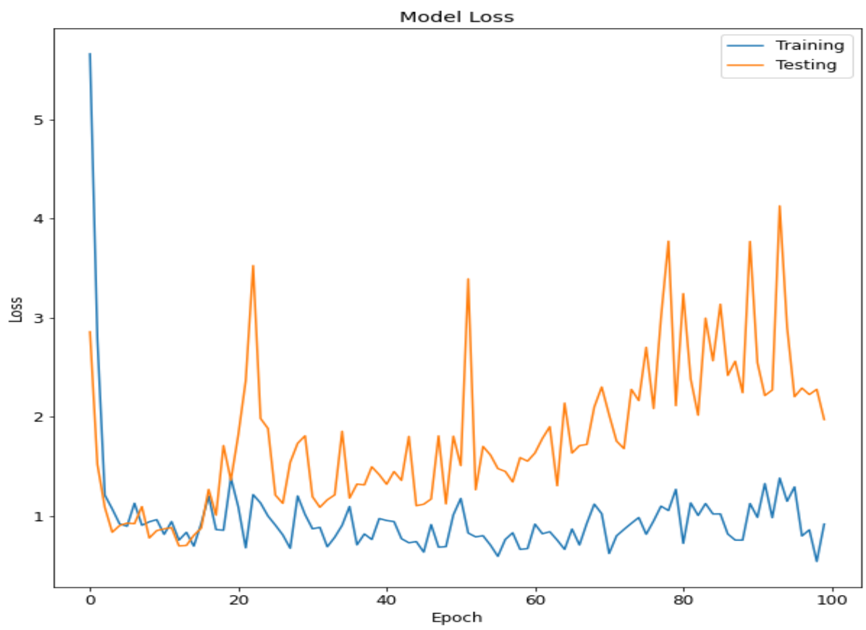


Figure 4.32: Model Loss Plot for Xception_ct model.

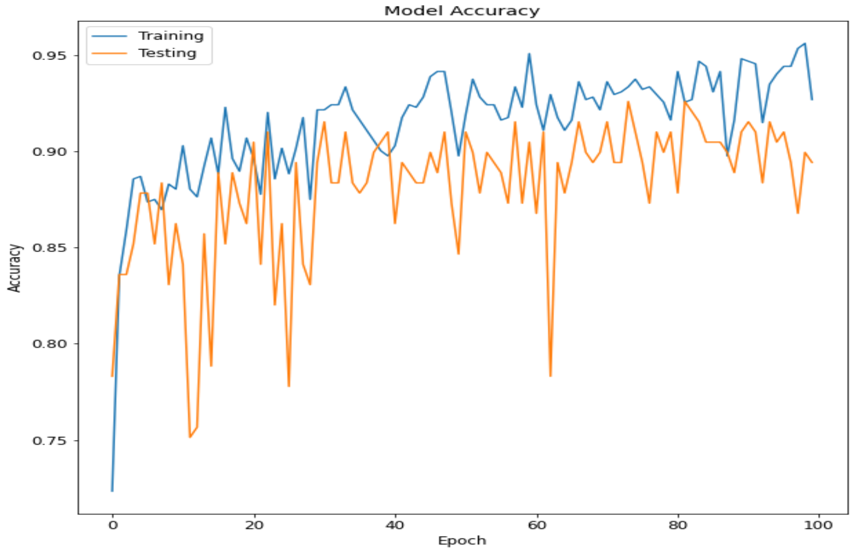


Figure 4.33: Model Accuracy Plot for Xception_chest model.

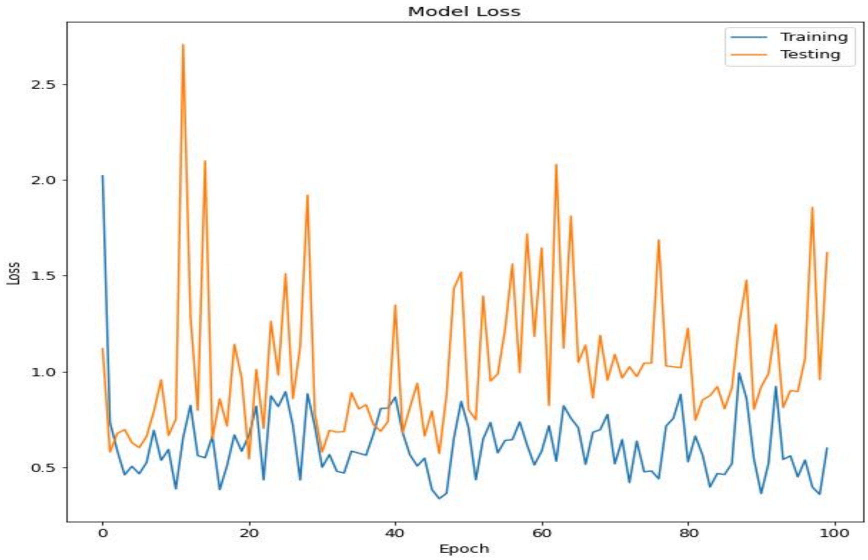


Figure 4.34: Model Loss Plot for Xception_chest model.

4.1.3 Comparison of different models on testing data:

Table 1 shows the comparison of different accuracies achieved by 10 model's named as vgg_chest,vgg_ct,Resnet_chest,Resnet_ct,InceptionV3_chest,InceptionV3_ct,Xception_chest and Xception_ct ,cnn_chest, cnn_ct on testing data set. The testing data set contains the records of 339 images that is 20% of the total dataset taken.

Table 1. Comparison of different models on testing data

Models	**Accuracy(%)**
VGG_Chest	75
VGG_CT	87
Resnet_Chest	68
Resnet_CT	67
InceptionV3_Chest	75
InceptionV3_CT	81
Xception_Chest	89
Xception_CT	81
CNN_Chest	65
CNN_CT	58

4.1.4 Graphical analysis of the model's trained:

Figure 4.35 depicts the accuracy analysis of models trained on CT scans.

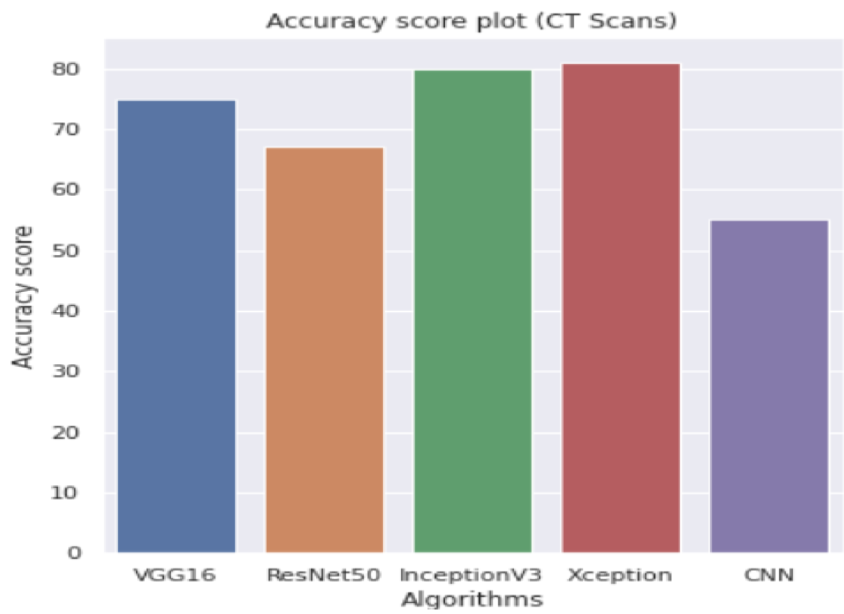


Figure 4.35: Performance Analysis of models based on CT scans.

Figure 4.36 depicts the accuracy analysis of models trained on Chest Xray's.

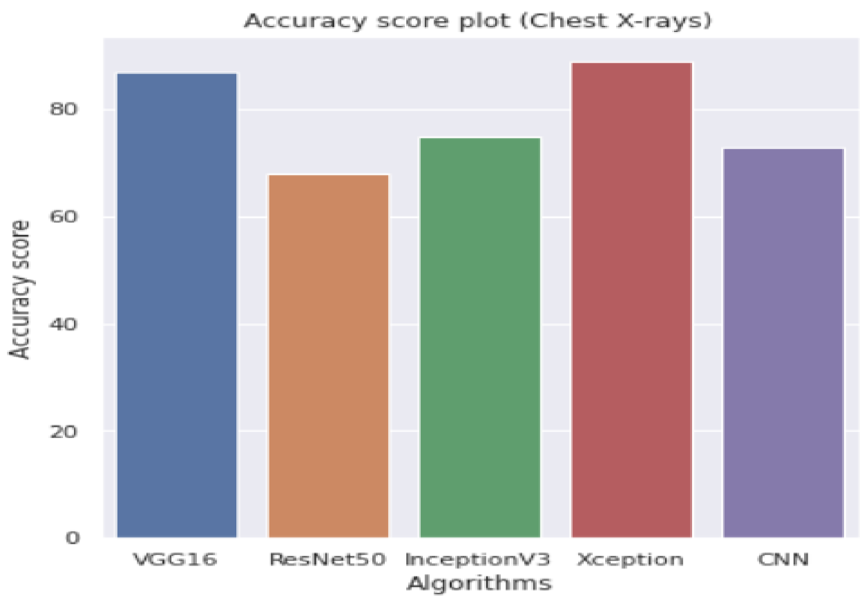


Figure 4.36: Performance Analysis of models based on Chest Xray's

4.2 Interface Development phase of Diagnostic System:

A diagnostic system interface has been built that takes a chest x-ray or CT scan picture as an input and predicts the severity of covid-19 using the four TL algorithms as an output. This interface is developed with the help of flask.

The figure 4.37, figure 4.38 and figure 4.39 shows the homepage of the interface developed.

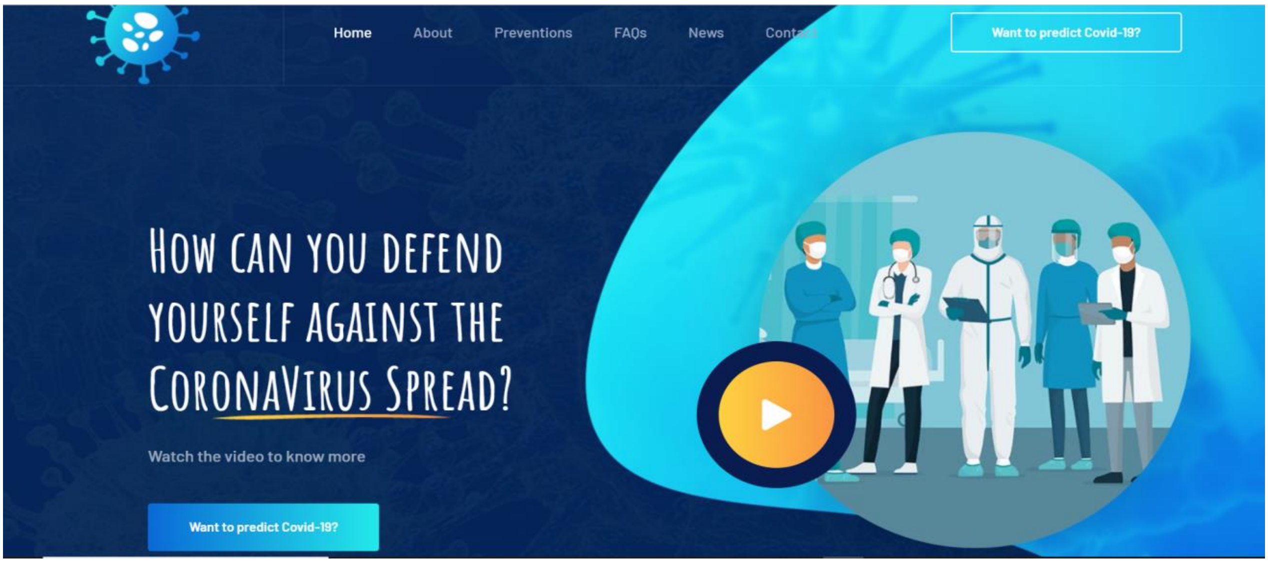


Figure 4.37: Top view of home page of the interface.

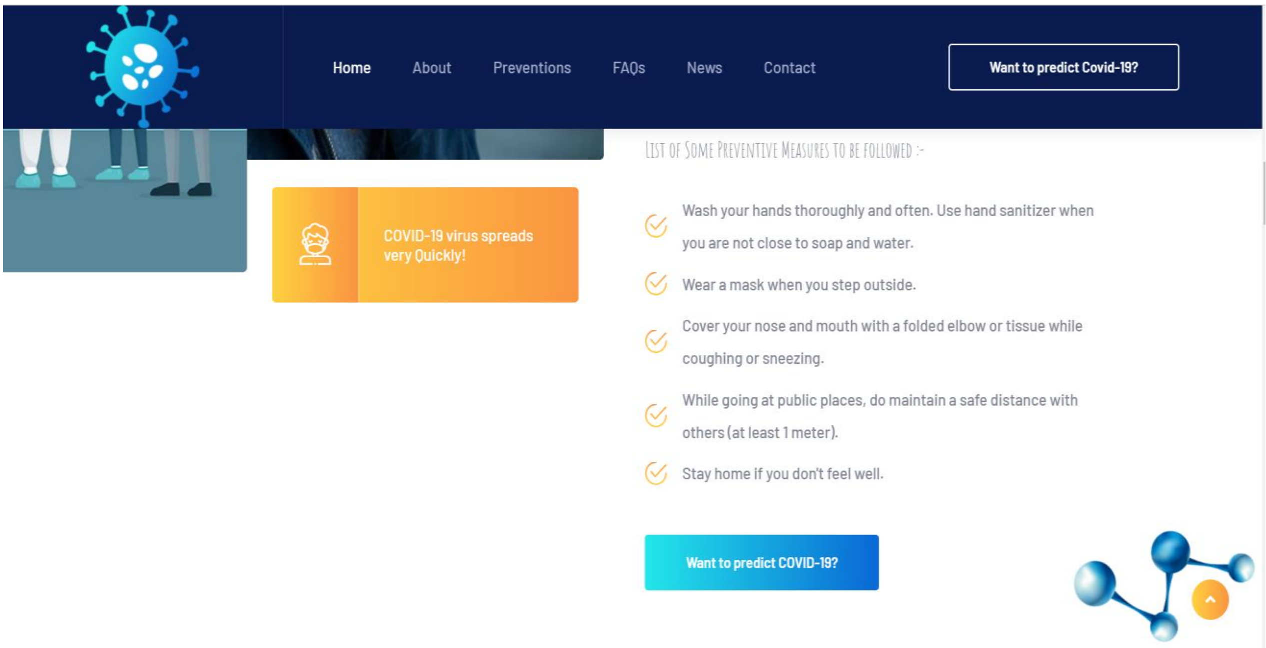


Figure 4.38: Middle view of home page of the interface.

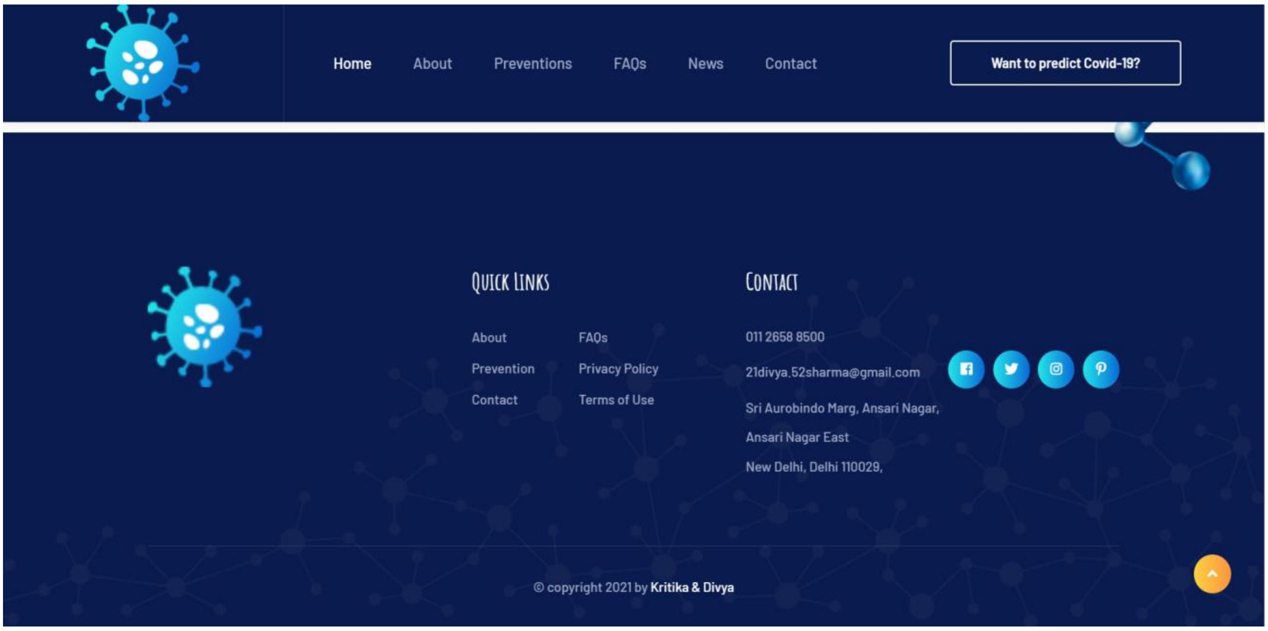


Figure 4.39: Bottom view of home page of the interface.

The figure 4.40 shows few of the frequently asked questions about covid -19.

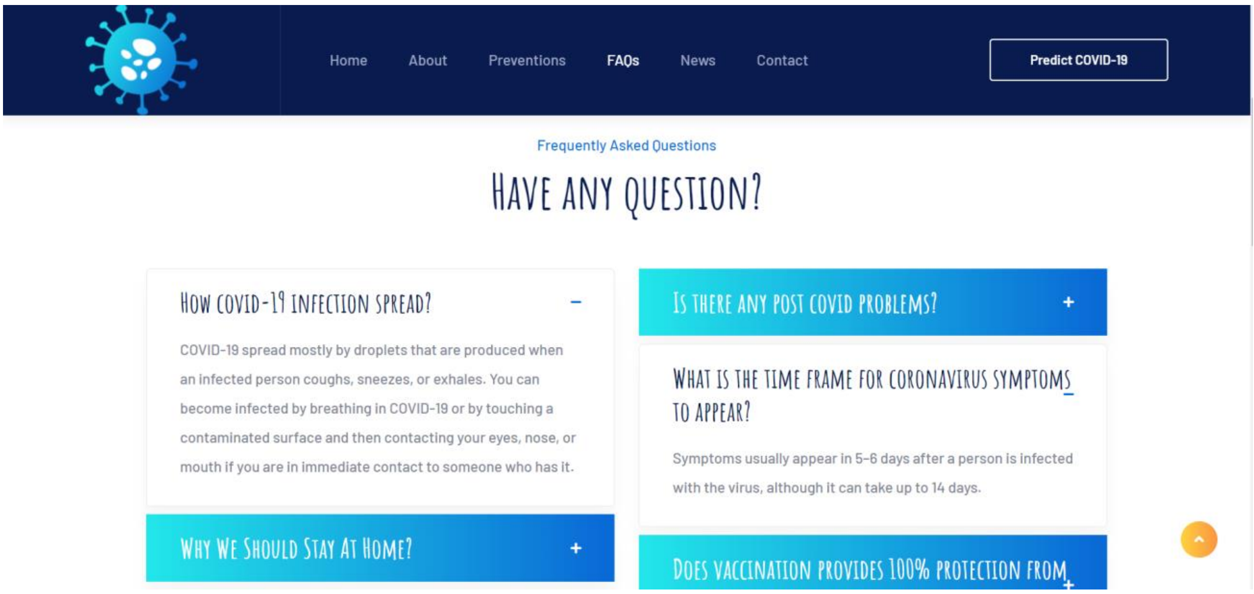


Figure 4.40: FAQs page of the interface.

The figure 4.41 shows the contact form page of the interface in which a user can ask a covid-19 related query and get the solution to his/her query immediately.

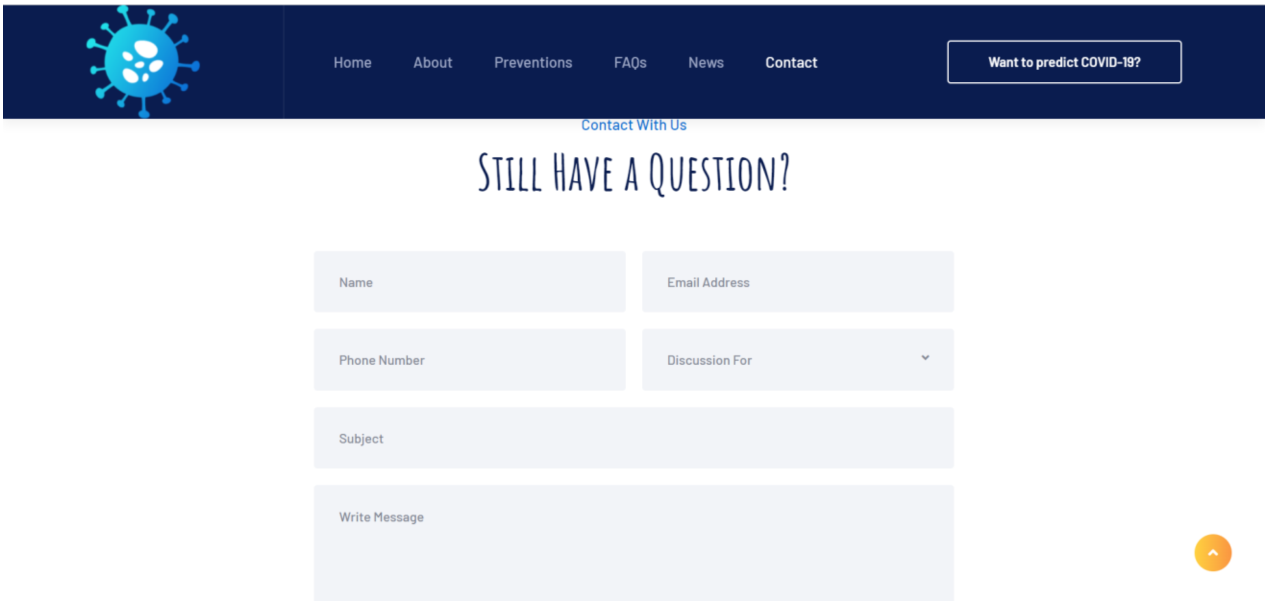


Figure 4.41: Contact page of the interface

The figure 4.42 demonstrates the upload page of the interface. A User has 2 options to upload an image in the form of chest x-rays and ct scans.

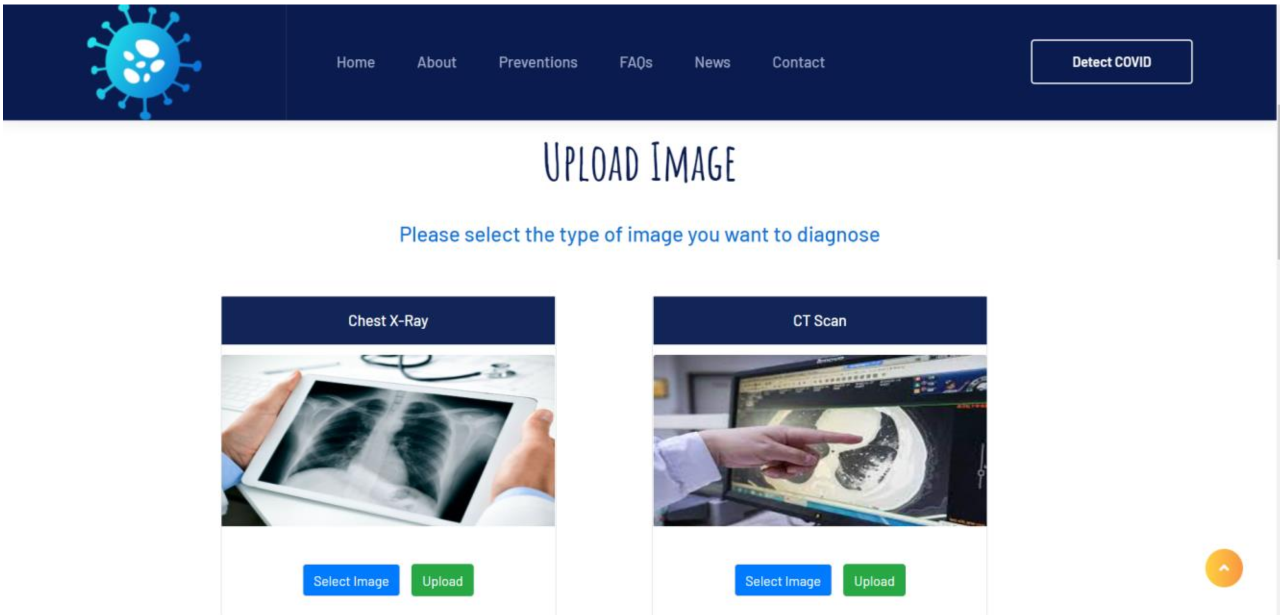


Figure 4.42: Upload page of the interface.

The figure 4.43 shows that the user has uploaded chest x-ray image on the interface.

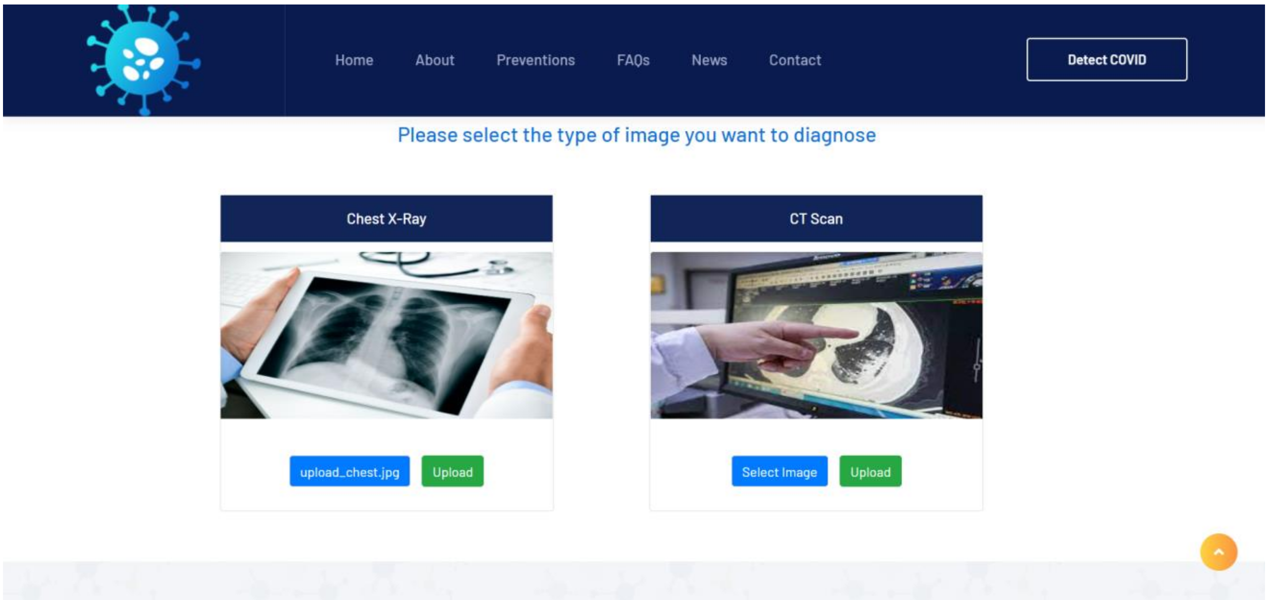


Figure 4.43: User Uploaded image on the interface.

The figure 4.44 shows the uploaded chest x-ray image that was entered by the user.

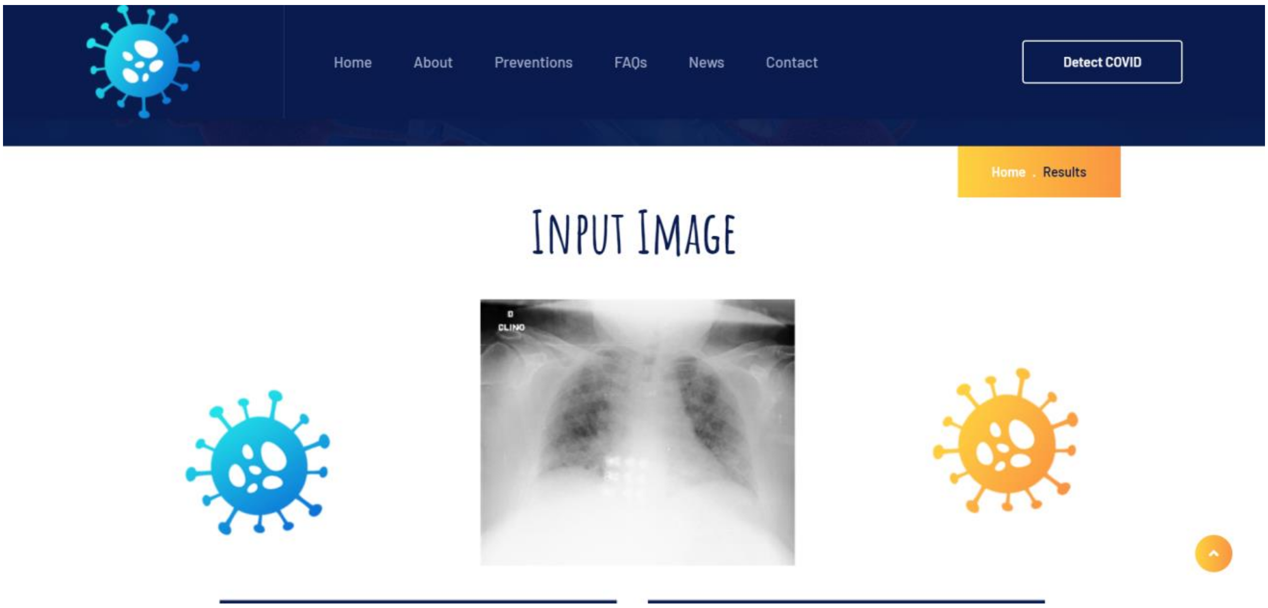


Figure 4.44: Uploaded chest x-ray image

The figure 4.45 and figure 4.46 predicts the results using four transfer learning algorithms based upon the image entered by the user.

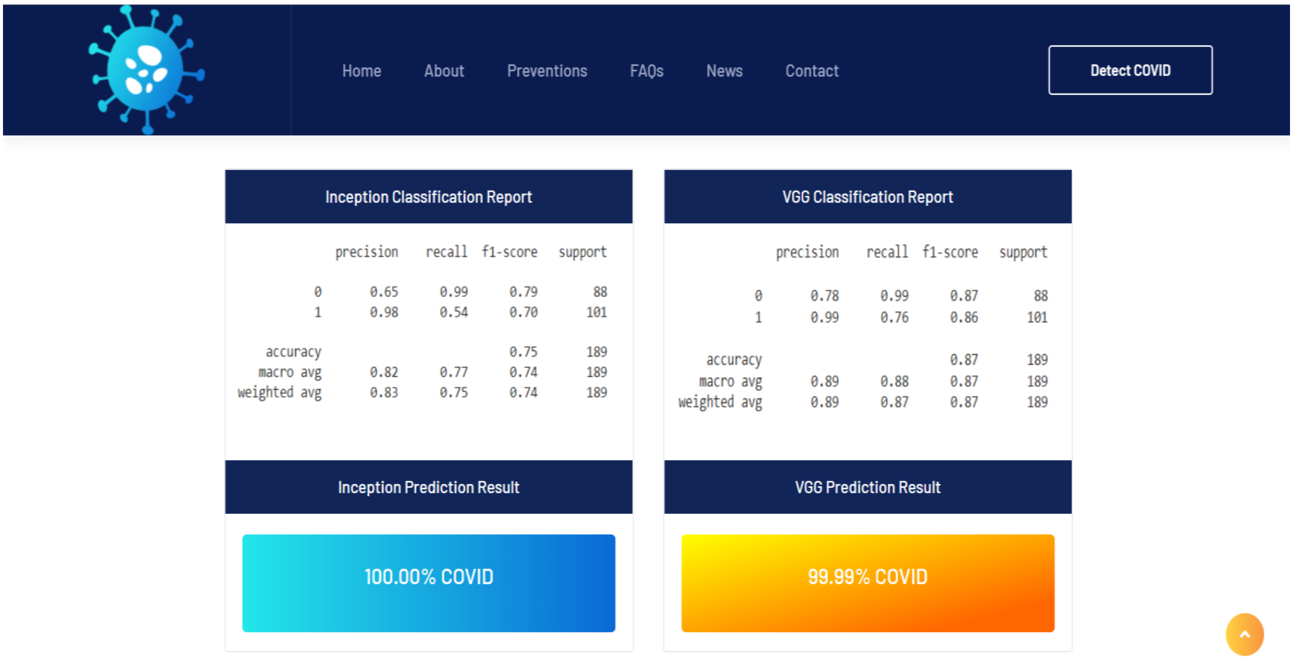


Figure 4.45: Results of the image entered by the user.

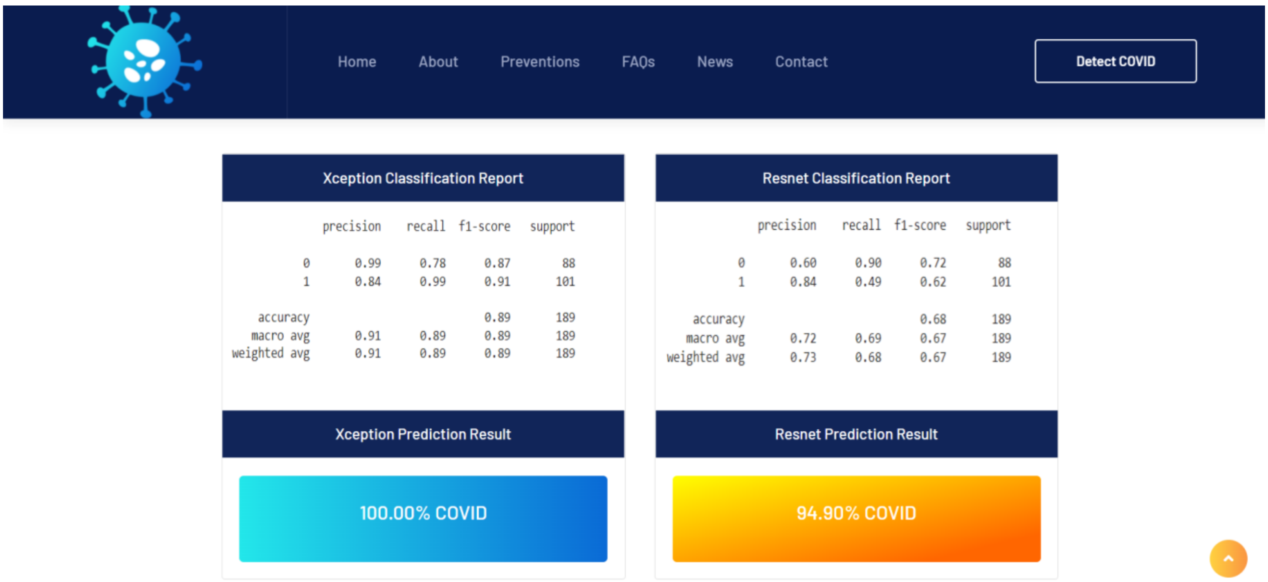


Figure 4.46: Results of the image entered by the user.

The figure 4.47 shows that the user has uploaded the CT-scan image on the interface.

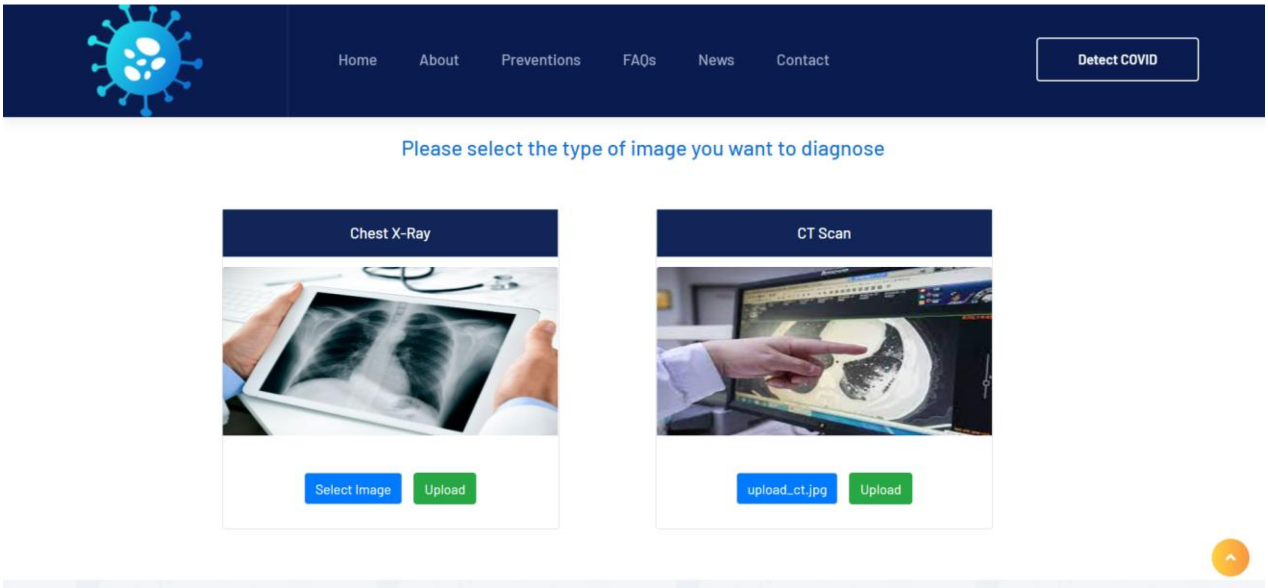


Figure 4.47: Upload page of the interface.

The figure 4.48 shows the uploaded CT scan image that was entered by the user.

Figure 4.48: Uploaded CT scan image

The figure 4.49 and figure 4.50 predicts the results using four transfer learning algorithms based upon the image entered by the user.

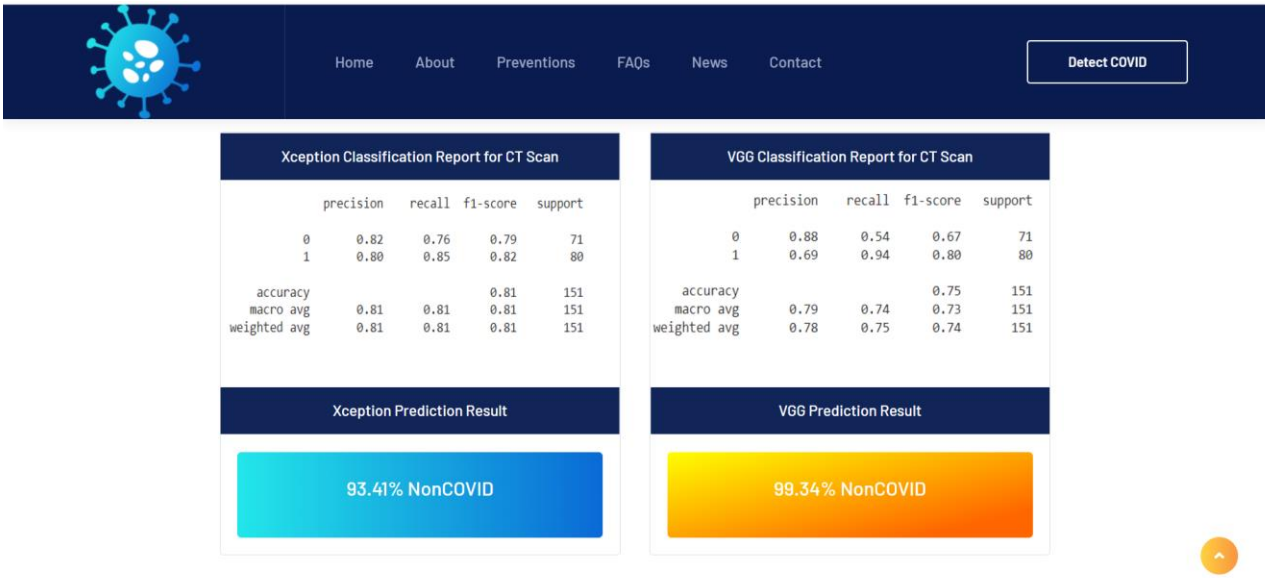


Figure 4.49: Results of the image entered by the user.

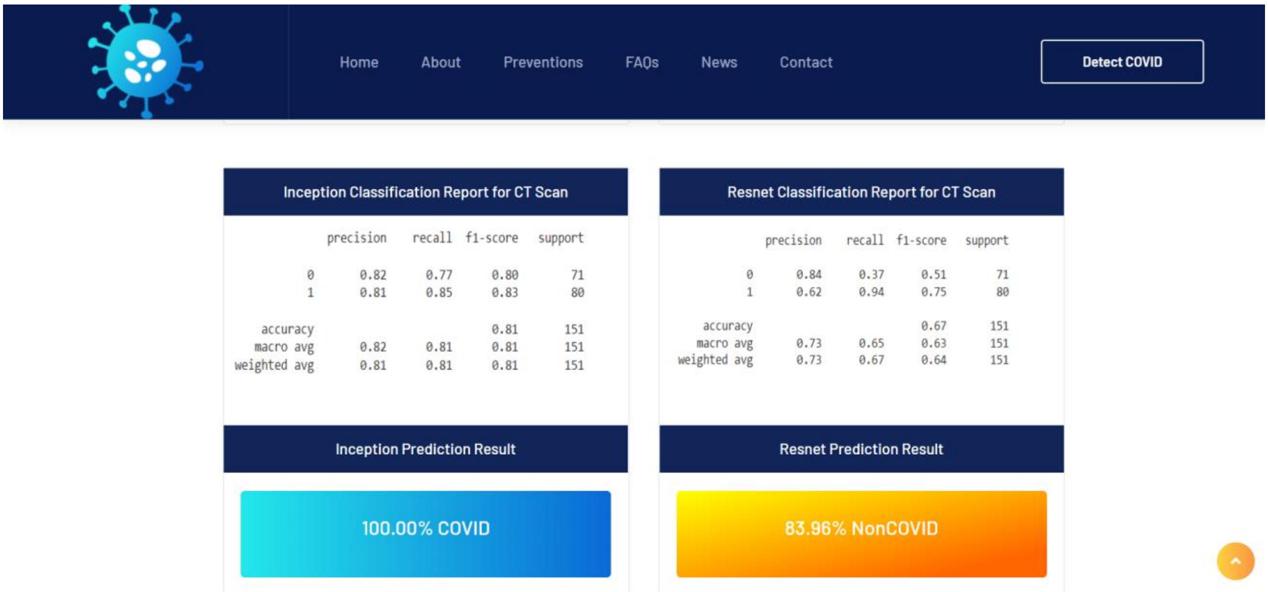


Figure 4.50: Results of the image entered by the user.

CHAPTER 5

CONCLUSIONS

5.1 Conclusions:

The virology research centre’s are currently seeking to overcome the current limitations of RT-PCR testing in order to more accurately detect the coronavirus. Chest imaging is a useful and effective method of detecting the clinical indications of COVID-19 suspects. Given the high prevalence of COVID-19 and the scarcity of competent radiologists, automated medical imaging approaches for detecting COVID-19 infection can help to speed up the diagnostic process and ensure a high-precision diagnosis. AI, ML, and DL are all potent technologies that can be used to develop early detection strategies for COVID-19. We began this research project by gathering the dataset, which consists of hundreds of X-ray and CT scan pictures of COVID-19 patients and healthy people. The visuals were then enhanced by resizing them to their respective CNN models typical sizes. After that, we partitioned the dataset into training and testing sets and used the VGG16, InceptionV3, Xception, Convolutional Neural Networks (CNN) and ResNet50 pretrained CNN models to create the assessment report of a patient, i.e., determine if a patient is a COVID-19 positive patient or not. Out of all these NN models, it is analyzed that the model based on Xception is the most accurate one in determining the presence of the disease with an accuracies of 81% and 89% on CT scans and Chest Xray’s. This two-classifier approach divides X-ray and CT scan pictures into two groups: healthy cases and COVID-19 positive cases. To classify COVID-19 from Chest Xray and CT scan pictures, the two-classifier diagnostic system employs an end-to-end DL architecture. Our two-classifier DL-based system explicitly predicts COVID-19 from raw images, without involving the extraction of features, and classifies these scan and radiology images into two categories: COVID-19 suspect and healthy cases, unlike standard medical image classification algorithms, which employ a three-step process of manual feature extraction followed by picture recognition and then diagnosis. Furthermore, because we know that Covid-19 is a disease that spreads at an exponential rate, understanding its pattern is crucial for controlling this lethal disease. Individuals found it incredibly difficult to obtain hospital beds and immunization Centre’s across India during the second wave. To solve this issue, we've used cutting-edge technology to analyse and visualize the number of confirmed cases, fatalities, and recovered cases in every state throughout the country. In addition, we looked at the number of hospital beds and icmr labs in each state, as well as the vaccine availability. We also have developed a user-friendly interface for the DL-based model that analyses radiological records to forecast the severity and existence of Sars cov-2 in the respective patient.

5.2 Future Scope:

We intend to use more visualization in the future to show lab availability and an updated immunization supply chain. We also intend to add additional COVID-19 positive and negative instances to our DL-based model's dataset. In addition, we also plan to train our models with more epochs. We will also expand our online interface in the future by implementing a live chat bot system, which will allow users/patients to communicate with a doctor who is available through the portal. Here the user will login to their account on the portal with their username and password and enter his/her medical details like symptoms that he/she are getting along with other medical history and on the basis of these values, the chat bot system will connect the user to the respective doctor. In this way the patient will be able to receive the online consultation without having to go to the hospital unless the condition is not serious. When it comes to diseases that are communicable and infectious, just like covid-19, this notion of teleconsultation or online consultation is quite useful.

5.3 Applications:

AI-based ML and DL systems are the most effective weapons against this deadly virus. Our automatic covid-19 diagnostic method may be used in a variety of situations. During pandemics, like as covid-19, the use of this automated image classification system would successfully prevent disease transmission from patients to radiologists. It will be able to construct remote video diagnostic programmes and robot prediction systems for first-class diagnostics when this combination of AI and DL approaches is employed on a large scale**.** Thus, machine learning and deep learning helps in the prevention of lives by forecasting the risk of infection at an early stage. It also aids healthcare workers in predicting the incidence of covid-19 and assisting them in analysing the pandemic scenario across India when this project is implemented on a large scale. This project when implemented on large scale also assist radiologists as these automated programs help them to identify the diseases on a much closer and accurate way. This computerized diagnosis method will aid patients in their fight against illnesses like covid-19. DL is simply one technique to give potential data-driven solutions to assist humanity in dealing with the devastating COVID-19.

REFERENCES

1. S. Lafraxo and M. el Ansari, "CoviNet: Automated COVID-19 Detection from X-rays using Deep Learning Techniques," *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, 2020, pp. 489-494, doi: 10.1109/CiSt49399.2021.9357250.

2. A. A. Khan, S. Shafiq, R. Kumar, J. Kumar and A. U. Haq, "H3DNN: 3D Deep Learning Based Detection of COVID-19 Virus using Lungs Computed Tomography," *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2020, pp. 183-186, doi: 10.1109/ICCWAMTIP51612.2020.9317357.

3. Ghosh, A., Nundy, S., & Mallick, T. K. (2020). How India is dealing with COVID-19 pandemic. *Sensors International*, *1*, 100021.

4. S. Bouaafia, S. Messaoud, R. Khemiri and F. E. Sayadi, "COVID-19 Recognition based on Deep Transfer Learning," *2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)*, 2021, pp. 1-4, doi: 10.1109/DTS52014.2021.9498052.

5. R. Kumari *et al*., "Analysis and predictions of spread, recovery, and death caused by COVID-19 in India," in *Big Data Mining and Analytics*, vol. 4, no. 2, pp. 65-75, June 2021, doi: 10.26599/BDMA.2020.9020013.

6. Ranjan Gupta, Saibal K. Pal & Gaurav Pandey ,"A Comprehensive Analysis of COVID-19 Outbreak situation in India." ,*2020 International Journal of Engineering & Technology*, *7*(2.8), pp. 684-687.

7. M. M. Islam, F. Karray, R. Alhajj and J. Zeng, "A Review on Deep Learning Techniques for the Diagnosis of Novel Coronavirus (COVID-19)," in *IEEE Access*, vol. 9, pp. 30551-30572, 2021, doi: 10.1109/ACCESS.2021.3058537.

8. Kaggle, online available at: https://www.kaggle.com

9. Covid19India.org (2020a). India Covid-19 Tracker 2020. Accessed from https://www.covid19india.org/ on 7th august 2021

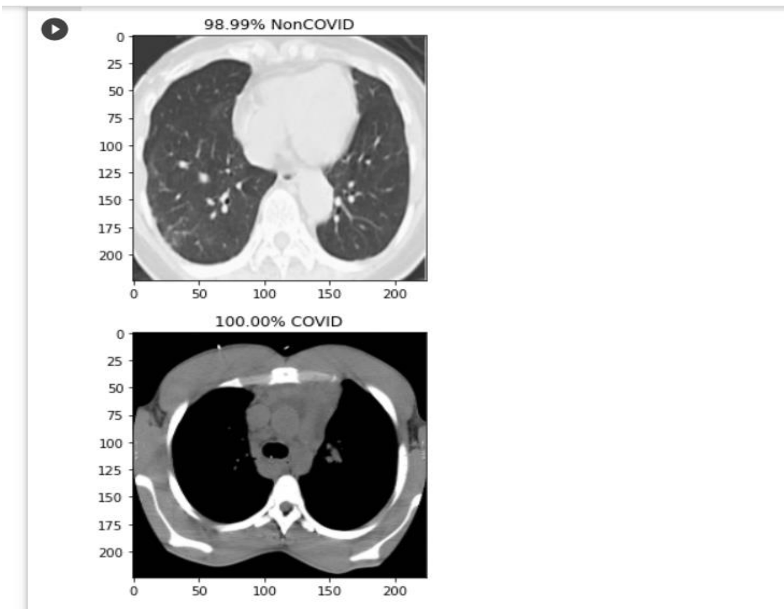
10. Health IT Analytics, online available at: https://healthitanalytics.com/features/what-is-deep-learning-and-how-will-it-change-healthcare

11. Y. Jiang, H. Chen, M. Loew and H. Ko, "COVID-19 CT Image Synthesis With a Conditional Generative Adversarial Network," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 2, pp. 441-452, Feb. 2021, doi: 10.1109/JBHI.2020.3042523

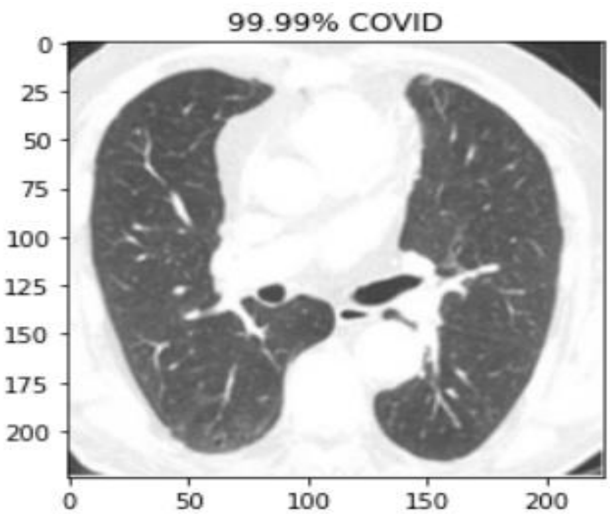
APPENDICES

Out of all these NN models, it is analysed that the model based on Xception is the most accurate one in determining the presence of the disease with an accuracy of 81% and 89% on CT scans and Chest Xray's. The following figures demonstrates the predictions made by the final model based on the Xception TL technique when we are testing the model on testing dataset.

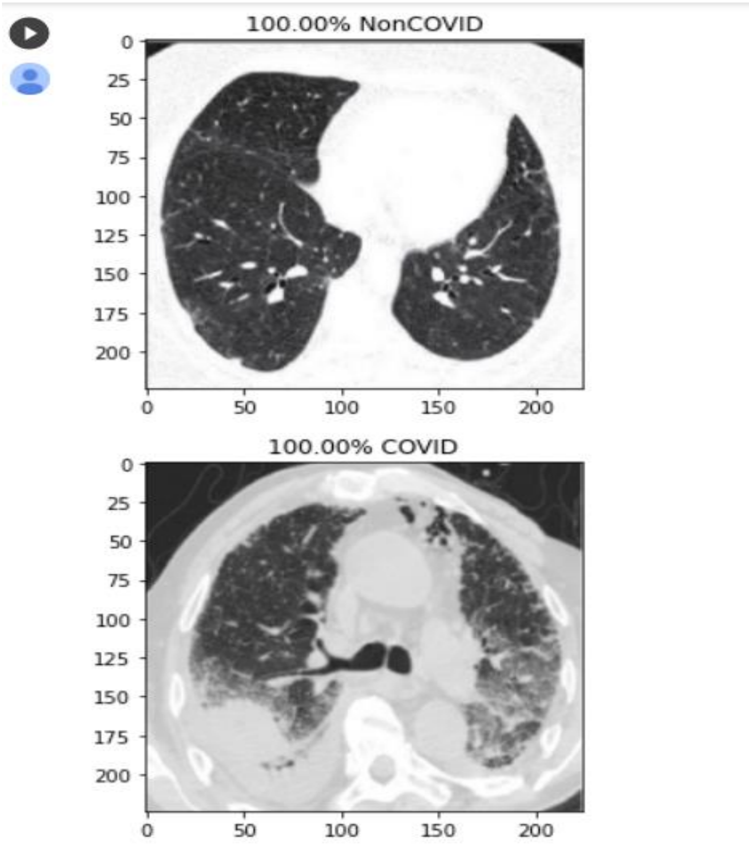
The predictions made by Xception_CT model are stated below:



Appendix 1: Xception_CT

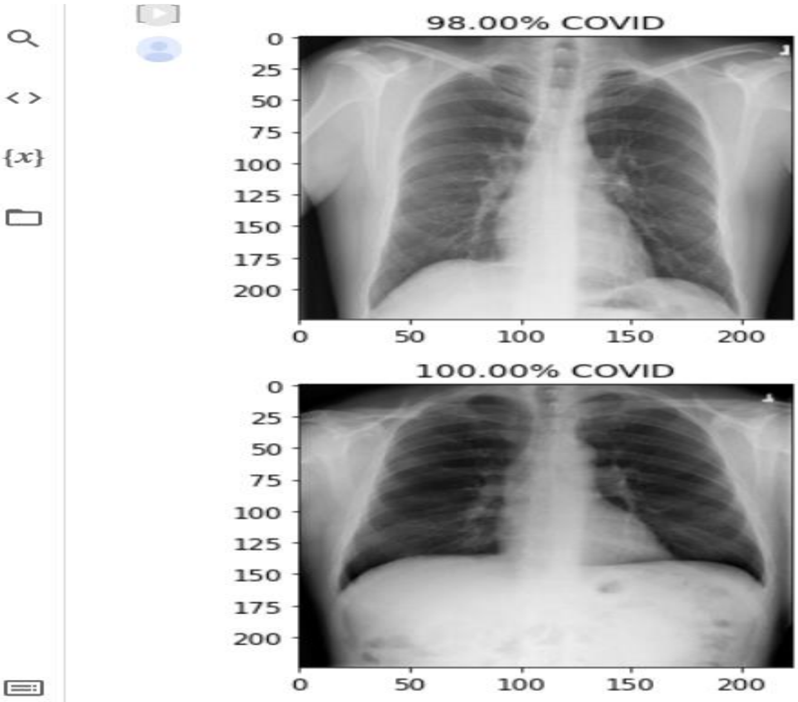


Appendix 2: Xception_CT

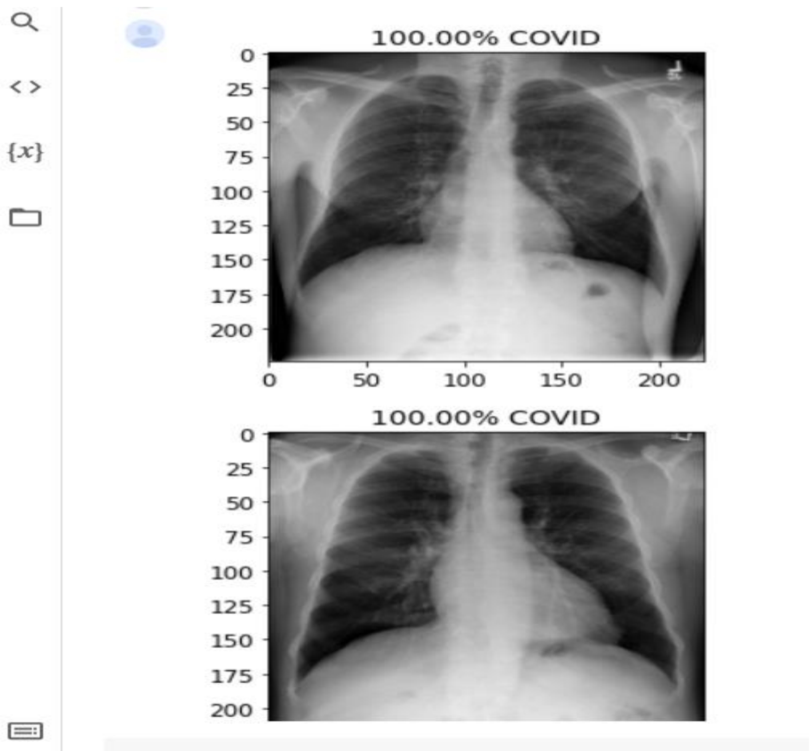


Appendix 3: Xception_CT

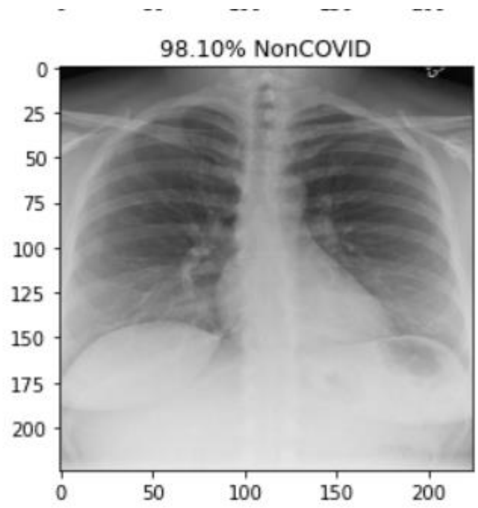
The predictions made by the Xception_CHEST model are stated below:



Appendix 4: Xception_Chest X-Ray



Appendix 5: Xception_Chest X-Ray



Appendix 6: Xception_Chest X-Ray