COVID DETECTION BASED ON X-RAY

a project report submitted by

SANDEEP M (URK17CS023)
KUNNATH RADHESH (URK17CS028)
SARA SUMANTH (URK17CS037)
KUDUMALA ANIL KUMAR REDDY (URK17CS114)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Dr. V. KARUNAKARAN

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-University under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114, India.

APRIL 2021

ACKNOWLEDGEMENT

First and foremost, we praise and thank **ALMIGHTY GOD** for giving us the will power and confidence to carry out our project.

We are grateful to our beloved founders Late **Dr. D.G.S. Dhinakaran**, **C.A.I.I.B, Ph.D.,** and **Dr. Paul Dhinakaran**, **M.B.A., Ph.D.,** for their love and always remembering us in their prayers.

We extend our thanks to **Dr. P. Mannar Jawahar, Ph.D.,** our honorable vice chancellor, **Dr. E. J. James, Ph.D., Dr. Ridling Margaret Waller, Ph.D.,** and **Dr. T. Lazar Mathew, Ph.D.,** our honorable Pro-Vice Chancellor(s) and **Dr. R. Elijah Blessing, Ph.D.,** our respected registrar for giving us the opportunity to carry out this project.

We are thankful to **Dr. G. Prince Arulraj, M.E., Ph.D.,** Dean (Engineering & Technology) for his support and encouragement.

We would like to place our heart-felt thanks and gratitude to **Dr. J. Immanuel Johnraja, Ph.D.,** HOD, Department of Computer Science and Engineering for his encouragement and guidance.

We are grateful to our guide, **Dr. V. Karunakaran** Assistant Professor, Department of Computer Science and Engineering for his valuable support, advice and encouragement.

We also thank all the staff members of the Department for extending their helping hands to make this project work a success.

We would also like to thank all my friends and my parents who have prayed and helped me during the project work.

BONAFIDE CERTIFICATE

Certified that this project report **"COVID DETECTION BASED ON X-RAY"** is the bonafide work of " **SANDEEP M(REG NO: URK17CS023), KUNNATH RADHESH(REGNO: URK17CS028), SARA SUMANTH (REG NO: URK17CS037), KUDUMALA ANIL KUMAR REDDY(REG NO: URK17CS114)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. J. Immanuel Johnraja	**Dr. V. KARUNAKARAN**
Head of the Department Department of Computer Science and Engineering	**Supervisor** Assistant Professor Department of Computer Science and Engineering

---

Submitted for the Project Viva Voce held on……………………….

Examiner

ABSTRACT

COVID-19 is an infectious and deadly type of disease that has affected millions of people in the world. It has shown the deadly nature by affecting the respiratory system. Clinical study reveals that infected people may experience symptoms that include fever, dry cough, headache, and respiratory illness. At the same time, the lungs, respiratory tract gets infected badly with the virus infection. So the lungs can be a prominent organ to diagnose the disease. Therefore, traditional RT-PCR methodology is used for clinical diagnosis. But it takes a few hours to get the results of the patient. So to get the result quickly and acquiring the highest accuracy when predicting the infected person is affected with COVID-19 or not, Chest X-Rays can be used to detect the infected part of the lungs using Convolution Neural Network(CNN).CNN based image augmentation is used for the detection of COVID-19. Using the dataset that is available in the public repository Consisting of Normal Chest X-Rays and COVID positive Chest X-Rays. Using tensor flow, Keras to build the model with multiple CNN layers consisting of an activation function\, max pooling, optimizer as Adam, followed by classification layer which used sigmoid as the activation function in the final layer. The model has performed better when compared to the existing state of art algorithms. The proposed model has obtained an accuracy of 98.6%, Thus the designed model is generalized and shall be able to detect the disease with maximum accuracy.

TABLE OF CONTENTS — PAGE NOs

LIST OF FIGURES AND TABLES

CHAPTER-1
INTRODUCTION

1.1 OBJECTIVE

Corona Virus(COVID-19) caused by severe respiratory syndrome has spread worldwide which has been declared as a pandemic by the world health organization, it has lead to a public health crisis. The rate of spread of the disease is rapidly increasing day by day. Various Countries started forcing to implement restrictions at various places which included social distancing, traveling limitations, etc. Various places have been identified as hotspots and protective measures are being taken care of. However, the rise in the spread of this disease is continuing. The bed occupancy and high Intensive Care Unit(ICU) admissions are increasing daily and to treat the patients with utmost care and precautions have to be made with accurate diagnosis solutions.

Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

The Objective of this project is to detect the COVID-19 effected patients from Chest X-Ray images using CNN(Convolution Neural Network).Using Traditional RT-PCR methodology, it not only takes a lot of time but the results obtained may contain more false positive or false negative results. Therefore,Covid-19 screening can be adopted using X-Ray. It is a layered architecture, The sequential model will have conv2d layers followed by maxpooling, dropout, flattening, activation function as relu and optimiser as Adam. Finally it classifies whether it is covid positive or negative.

1.2 PROBLEM STATEMENT

Novel coronavirus or SARS-COV-2 strain is responsible for COVID-19 and it has already shown the deadly nature of respiratory disease by threatening the health of millions of lives across the globe. Blood test usually takes a lot of time and people need to wait for so long to get the report. The possibility to use widespread and simple chest X-ray imaging for early screening of COVID-19 patients is attracting much interest from both the clinical and the AI community. Using Deep learning models to produce faster and better results in a short interval of time

The best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face.

The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it's important that you also practice respiratory etiquette (for example, by coughing into a flexed elbow).

Many researchers have been analyzing data across the country and in some large cities, looking at number of confirmed cases and deaths based on race and ethnicity and related factors. What they found is that African Americans and the Latino-Hispanic populations have disproportionate higher rates of hospitalizations and deaths due to COVID-19.

However, Identifying positive COVID-19 cases in the early stages helps to isolate the affected patients as early as possible. Hence breaking the chain of transmission and flattening the epidemic curve. [1] Using Reverse Transcription Polymerase Chain Reaction (RT-PCR) which is currently used as standard diagnosis method to treat the covid affected patients, However, the time taken for the test is quite long and the results are not accurate because of its low true positive rate and it also requires the necessary pieces of equipment to access easily. Instead of wasting hours to get a result, Technology can place a vital role to produce accurate and faster results. Deep learning plays an important role in which Convolutional Neural Networks can be used for the prediction, classification of various data.

1.3 CHAPTER WISE SUMMARY

World Health Organisation has declared that the COVID-19 has been spread across various countries. Quick detection of the COVID19 is required to control the spread of the disease. Coronaviruses are often found in bats, cats and camels. The viruses live in but do not infect the animals. Sometimes these viruses then spread to different animal species. The viruses may change (mutate) as they transfer to other species. Eventually, the virus can jump from animal species and begins to infect humans. In the case of COVID-19, the first people infected in Wuhan, China are thought to have contracted the virus at a food market that sold meat, fish and live animals – but they are still investigating. Although researchers don't know exactly how people were infected, they already have evidence that the virus can be spread directly from person to person through close contact. Researchers are still studying other factors that may make ethnic groups more susceptible to negative COVID-19 outcomes, including genetics and possible differences in lung tissue as well as socioeconomic status and the social environment and systems.

The Deep learning models are used in various fields especially in the field of health care to diagnose the disease and produce accurate results in a short period and it has been an emerging tool in the field of Artificial Intelligence, where the model learns its self with the help of previous results. The deep learning Convolutional Neural Network consists of multiple layers that can predict and classify based on the input data. Different augmentation techniques are used such as Normalisation, Zoom Augmentation, shear Augmentation to process the data before passing it to the model. Being a sequential Model, consisting of multiple layers in which the input data is passed into the model. To avoid overfitting of the model multiple dropout layers are used. The model learns from each layer by its weight parameters.

Initially, the receptive field is small, it means that only a smaller part of the images are visible but as the layers increase the model will be able to find the various complex patterns from the image which indeed helps the model to perform better and produce accurate results.

As the layer in the model increases, the filter size increases which helps in capturing the larger portion of the image. Once the model is ready, training the data is performed and later on, the validation data can be used to check whether the model is working with the new set of data. The Convolutional Neural Network (CNN) model summary and working of the model will be explained at the later stage of this paper. In the proposed model, CNN architecture is used to detect the COVID-19 using the chest X-Rays, without using any pre-trained model, this model has performed better with the open-source dataset of Covid Chest X-ray images [2] and the Normal Chest X-Ray images.

The dataset consisting of 192 Covid Chest X-Ray images initially later the data set size have been increased and to make sure the model is generalized, an equal amount of Normal Chest X-Rays are taken thus the model is generalized which can be able to perform better compared to other pre-trained models and able to predict the effected patients with maximum accuracy In the later section of this paper, the complete architecture with the methodology is explained.

CHAPTER-2
SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

COVID-19 does not have a rich state-of-the-art literature in the AI-driven tool perspective where imaging techniques are used. Most existing tools have been designed for the training of X-rays or CT scans on a single type of data. Moreover, separate models for each type of data are presented in the reports. In order to integrate various modalities (data types), more information can be provided can an architectural structure be designed which can handle several data modalities? Inspired by computer vision and pattern recognition techniques, various classes are employed to train in exactly the same architecture,In this paper, we propose a single architecture for detecting COVID-19 positive cases using CT scans and CXRs. The proposed architecture was put to the test using publicly accessible datasets that included both CT scans and CXRs.

In [3] this paper, they have adapted a previously proposed convolution neural network architecture based on class decomposition, which they term Decompose, Transfer, and Compose (DeTraC) model, to improve the performance of pre-trained models on the detection of COVID-19 cases from chest X-ray images. This is by adding a class decomposition layer to the pre-trained models. The class decomposition layer aims to partition each class within the image dataset into several sub-classes and then assign new labels to the new set, where each subset is treated as an independent class, then those subsets are assembled back to produce the final predictions. Validation was performed on DeTraC with different pre-trained CNN models, where the highest accuracy has been obtained by VGG19 in DeTraC. As this is a pre-trained model a lot of layers have to fine-tuned and the dataset is also consisting of fewer images that may overfit by using VGG19.

The clinical and paraclinical aspects of COVID-19 were reported by Huang C et al. in January 2020. They said that anomalies such as Ground-Glass Opacity (GGO) can be detected using chest CT scans (based on 41 positive cases). CT scans are commonly used to detect irregular patterns in COVID-19 cases that have been confirmed. To be more specific, Li and Xia tested 51 CT scans (images) and found that COVID-19 was successfully detected in 96.1 percent of the time. Zhou S et al. studied 62 COVID-19 and Pneumonia patients, and their findings revealed a variety of trends that resemble lung parenchyma and interstitial diseases.Zheng Ye et al. also stated that typical CT manifestations help radiologists to make decisions and familiarise themselves with them. The CT indications of COVID-19 are typical for GGOs, amalgamation, cross-link marking, and crazy marking. There have been emerging atypical CT indications among COVID-19 patients including changes of airways, pleurals, nodules and fibrosis. Fang et al. have also reported that radiological imaging may be better used to diagnose COVID.The experiments were conducted with 81 patients not performing initial TB in 30 patients, and 51 were with both initial TB and TB-PCR (RT) within 3 days of the reverse transcription polymerase chain reaction (RT-PCR). Chung et al. reports that GGOs were mainly found among 57 percent of patients with CT, while peripheral distributions were found in 33 percent of cases. Normal chest CTs were reported among 14 percent of patients. Song et al. mainly described the GGOs opacity in 77 percent, but in 86 percent the peripheral distribution was larger and in 90 percent the lobe involved.In addition to the progression of imagery observed in 31% of the cases, the author reports significant improvements to chest CT images in 54% of the cases. In 21 patients, Feng Pan et al. experimented with a confirmed presence of COVID-19 pneumonia in their average age of 25 years to 63 years. The authors concluded in their study that the most serious breathing abnormalities were observed in the chest CT pictures (lungs) after 10 days after the first symptom in patients recovered from COVID-19 without any high difficulty.

In order to detect the COVID-19 cases from CT images, Wang et al. presented a profound learning approach. Experiments in 453 COVID-19 positive cases have been conducted. They have shown 82.9 percent precision. In addition, an external test data set reported an exactness of 73.1 percent, which was 67 and 74 percent with specified and sensitivity values. Butt et al. studied various CNN models for CT images identification of COVID-19. The 2D and 3D CNNs were tested and an AUC of 0.996 reported. In addition, 98.2 percent and 92.2 percent were calculated for sensitivity. During the testing, 219 COVID-19 CTs were collected.

In [4] this paper dataset is highly unbalanced, evaluating the ROC-AUC of the binary classifications and the confusion matrices. The best outcome would be to get a 0.5 ROC-AUC, which would mean that the two datasets cannot be distinguished by this model. However, one can see that AlexNet is very capable of recognizing the dataset without using the lungs. Augmentation needs to be performed to get the best accuracy. In this [5] paper a collection of 1427 X-ray images including 224 images with confirmed Covid-19 disease, 700 images with confirmed common bacterial pneumonia, and 504 images of normal conditions. Secondly, a dataset including 224 images with confirmed Covid-19 disease, 714 images with confirmed bacterial and viral pneumonia, and 504 images of normal conditions. The data was collected from the available X-ray images on public medical repositories. The results suggest that Deep Learning with X-ray imaging may extract significant biomarkers related to the Covid-19 disease, while the best accuracy, sensitivity, and specificity obtained is 96.78%, 98.66%, and 96.46% respectively

In this paper[6], a novel model called COVIDetectioNet is presented for the detection of COVID-19 viral disease using a pre-learned deep features ensemble and feature selection. The approach proposed consists of three key stages based on chest X-ray images. In the first stage, the convolution and fully-connected layers of a pre-trained AlexNet architecture are used as a feature extractor. In the second stage, the most efficient features are selected from the combined obtained deep features using the Relief algorithm. In the final stage, the classification of the effective features is conducted using the SVM method. In this study, the transfer learning approach was used for the pretrained CNN-based AlexNet architecture. The greatest disadvantage of the models that are created ex novo is that a largescale dataset is required for training purposes, and training the model developed requires a considerable amount of time. Transfer learning enables information to be used from pre-learned tasks and this information is then reapplied later in order to solve other problems

For detective COVID-19 cases of weak CT images, Zheng et al. presented a deep learning approach. Their system has been trained at 499 volumes and has been tested at 131 volumes and sensitivity, respectively, with specific value values of 0.907 and 0.911. In order to distinguish CVID-19 cases from CXR, Farooq and Hafeez presented a deep learning approach. They reported 96.23 per cent accuracy at 41 epochs with ResNet50, which included 8 COVID-19 positive cases. For the identification of COVID-19 cases from C XRs, Hall et al. used a deep learning approach. In their experiments, 89.2% were reported with an overall accuracy of 0.8039 and an AUC of 0.95% of 135 COVID-19 cases were positive. In addition, a testing set of 33 CXRs was used and 91,24% accuracy, a true positive rate of AUC, and 0.7879 and 0.94 respectively, were reported. In the detection of COVID-19 cases, Salman et al. have used a CNN approach. For 130 positive COVID-19 cases, 100 percent accuracy was reported.

With the aid of X-ray images, the authors proposed a structure model focused on Capsule Networks for diagnosing Covid-19 (i.e., COVID-CAAPS) disease. To solve the problem of class imbalance, multiple convolution layers and capsules are used in this proposed work. They demonstrated COVID-satisfactory CAPS's output on a smaller number of trainable parameters in an experimental study. The authors listed the considered trained model, which is open source and publicly available on Github. As a result, they concluded that the proposed model has a 95.7 percent accuracy, a 90 percent sensitivity, and a 95.80 percent specificity by using a smaller number of trainable parameters.The authors considered the first three cases of Covid-19 infected cases in France. Out of these three persons, two were diagnosed in Paris and one in Bordeaux. Before coming in contact with Covid-19 diseases, they were staying in Wuhan, China.

The author proposed a hybrid artificial intelligence system that incorporated machine learning and deep learning algorithms (e.g., Convolutional Neural Network (CNN) with softmax classifier). Using chest X-ray images, the proposed device is specifically designed to detect Covid-19 events. The authors conducted a radiologic examination of MERS (Middle East Respiratory Syndrome) on a novel coronavirus. They looked at the case of a 30-year-old man who was experiencing diarrhoea, fever, and abdominal pain. The authors looked at how contaminated people were treated with chest X-rays.

They also used this model to boost the results of a dataset of chest X-ray and CT images that they had obtained. They also spoke about what kind of procedures hospital personnel should adopt to keep healthy patients safe and what precautions they should take when caring for covid-19 contaminated patients. The writers spoke about the aetiology outbreak in Wuhan, China. They have addressed the issue of the epidemic's exact cause. They assess the effect of travel (by commercial or air) on covid-19 in this report.

To detect pneumothorax, the authors used the SVM technique. They mined the characteristics of lung images using a Local Binary Pattern (LBP). The authors used multi-scale texture segmentation to segment the regions of abnormal lungs in the proposed detection model by removing impurities from chest images. This transformation was also used to adjust the texture in order to find several overlapping blocks. Finally, the authors used rid boundary (with Sobel Edge detection) to locate an entire disease area containing the abnormal component. The authors looked at the chest CT scans of 21 covid-19 patients in Wuhan, China.The authors are particularly interested in the effects of covid-19 disease on human lungs. The authors then proposed a COVID-RENet model for extracting features (i.e., edge and region-based) and classifying them using CNN. The authors obtained features in this study by using CNN, and then used SVM to enhance classification accuracy. On a compiled dataset of Covid-19, they used 5-fold cross-validation. This proposed method is primarily intended for use by a medical professional in the early detection of Covid-19-infected patients.

Furthermore, an author has presented a report on the effects of covid-19 on the kidney and acute renal failure. They looked at a dataset of 50 patients with Covid-19 disease and divided them into two recovery classes (i.e., good and poor). The dynamics of serological and viral shedding were investigated. The authors then discovered a connection between poor recovery and lung infections as a risk factor. As a result, they came to the conclusion that 58 percent of the patients' recovery was fragile.

The authors conducted research on the overall number of patients infected with Covid-19 and death cases worldwide. The authors suggested using X-ray images to detect patients infected with Covid-19 using a deep based approach (with vector gadget classifier). This approach is useful for hospital doctors in identifying cases of covid-19 infection early on. With the support of various matrices parameters, they find that the proposed model for lung classification has a 97.48 percent accuracy. The writers spoke about how a new coronavirus was discovered as a new pneumonia disease in Wuhan, China. The primary goal of this paper was to introduce COVIDX-Net, a new deep learning system for assisting clinical practitioners in automatically diagnosing Covid-19 disease using X-ray images.

The authors also explored the various methodologies for detecting covid-19 disease as well as the difficulties they encountered. They also suggested that an automated system for detecting the Covid-19 virus be developed in order to prevent the disease from spreading through contact. The researchers then looked at various chest X-rays to see whether they could detect pneumonia and came to the conclusion that it's difficult to say if Covid-19 induces pneumonia or whether any other symptoms are to blame.

The authors addressed the use of chest radiography (CXR) to detect lung abnormalities. Because of its complete availability and decreased infection control, they demonstrate that the medical community can depend on CXR. For the identification of Covid-19 pathogens, they used 123 front views of X-rays. The authors also talked about the importance of AI in healthcare. They also discussed the difficulties of applying AI tools on a smaller dataset of X-ray images (which is available publically). The researchers used deep learning and transfer learning algorithms to detect Covid-19 diseases using a dataset of X-rays and CT images from several sources.As a result, they discovered that the pre-trained model had a 98 percent accuracy rate, while CNN showed that the model had a 94.1 percent accuracy rate. The authors used 150 CT images to remove two subsets (16*16 and 32*32) patches to create sub-datasets, and 3000 X-ray images were labelled for Covid-19.

To improve the efficiency of a proposed approach, fusion and ranking approaches have been used. To identify the processed data, the authors used SVM, and to pass learning, they used the CNN model. As a result, they discovered that set 2 had better accuracy than set 1.

The authors suggested a model that uses Chest X-ray images to detect the Covid-19 automatically. On two separate classification models, the planned model is used to provide reliable diagnostics (i.e., binary and multi-class).To identify the real-time object detection process, they used the DarkNet model. The use of thermoplasmonic in the identification of Covid-19 diseases was explored by the scientists. The authors look at patients who were diagnosed with covid-19 pneumonia and admitted to a hospital in Wuhan, China. They divided CT scan patients into different categories, and the image's features and distribution were further analysed and compared in order to detect Covid-19 diseases. The authors suggested a KE Sieve Neural Network architecture that uses Chest X-ray images to help locate Covid-19 analyses. Their proposed model has an accuracy rate of 98 percent.

The authors conducted an analysis of covid-19 on people who had been isolated in Rawalpindi's BBH hospital ward. Using a chest X-ray dataset, the authors developed a CNN-based algorithm for evaluating pneumonia. For moving learning on CNN, they used two separate models (VGG16 and InceptionV3). They went on to use SVM to improve their performance. The authors discussed the use of a deep anomaly detection system for accurate Covid-19 patient screening. They obtained 100 chest X-ray photographs, with 70 people testing positive for Covid-19. Finally, they spoke about how Covid-19 affects humans. They looked at a dataset of 101 Covid-19-infected pneumonia cases. The study's main aim was to equate the clinical state of Covid-19 pneumonia to CT images. Covid-19 is a virus disease that not only affects humans but also a region, according to all of the viewpoints and proposed work presented by various researchers. They addressed various methods for early detection of Covid-19 events. On a compiled dataset of chest X-ray images, we used three templates (Inception V3, XCeption, and ResNeXt).

In this work[7], They have proposed the Truncated Inception Net deep learning model to detect COVID-19 positive patients using chest X- rays. For validation, experimental tests were done on six different experimental datasets by combining COVID-19 positive, Pneumonia positive, Tuberculosis positive, and healthy CXRs. The proposed model outperforms the state-of-the-art results in detecting COVID-19 cases from non-COVID ones. Besides, considering the number of parameters used in our proposed model, it is computationally efficient as compared to the original Inception Net V3 model and other works proposed in the literature. It is important to note that the study has no clinical implications. Instead, we solely aimed to check whether the proposed Truncated Inception Net could be used in detecting COVID- 19 positive cases using CXRs. The proposed framework consists of several Capsule and convolutional layers, and the lost function is modified to account for the class imbalance problem. COVID-CAPS achieved an Accuracy of 95.7%, Sensitivity of 90%, Specificity of 95.8%, and Area Under the Curve (AUC) of 0.97 while having far less number of trainable parameters in comparison to its counterparts. To potentially and further improve the diagnosis capabilities of the COVID-CAPS, pre-training and transfer learning are utilized based on a new dataset constructed from an external dataset of X-ray images. This is contrary to existing works where pre-training is performed based on natural images. Pre-training with a dataset of similar nature further improved accuracy to 98.3% and specificity to 98.6%.

In this paper[8] They have developed a highly accurate model for COVID-19 CT diagnosis by exploring the benefits of joint learning from heterogeneous datasets of different data sources. proposed a novel joint learning framework through redesigning the recently proposed COVID-Net from architecture and learning strategy as a strong backbone. The joint learning framework explicitly mitigates the inter-site data heterogeneity by conducting separate feature normalization for each site. Experiments on two large-scale public datasets demonstrate. In Comparision with various existing works, The proposed model has obtained the highest accuracy of 98.6% and in the above-mentioned papers where the pre-trained model is used but the model can likely be overfitted because of very less dataset and the layer have to carefully fine-tuned to obtain the best accuracy for which the proposed model will be able to achieve it.

2.2 PROPOSED SYSTEM

To prevent the spread of COVID-19, a large number of suspected cases must be screened, accompanied by appropriate medication and quarantine. While RT-PCR testing is considered the gold standard, it has a high rate of false-negative results. To combat the epidemic, efficient and rapid analytical techniques are eagerly awaited. We propose to create a deep learning model that would mine the specific features of COVID-19 to offer a clinical determination in front of the pathological examination, thereby sparing crucial time for sickness regulation.

Understanding the fundamental concept of COVID-19 and its subtypes, varieties, and variations should be rendered and spread all over the world. Baidu Research has released its LinearFold calculation and administrations, which can be used for whole-genome optional structure forecasts on COVID-19 and is clearly faster than other calculations. COVID-19 pathological findings linked to severe respiratory pain disorder are shown in. Investigations in included the following:1) a summary of the patient's characteristics; 2) an evaluation of age appropriateness 3) calculation of case fatality and death rates; 4) geographical and temporal analysis of viral spread 5) the emergence of epidemiological bends, and 6) subgroups

In Comparision with various existing works, The proposed model has obtained the highest accuracy of 98.6% and in the above-mentioned papers where the pre-trained model is used but the model can likely be overfitted because of very less dataset and the layer have to carefully fine-tuned to obtain the best accuracy for which the proposed model will be able to achieve it.

2.3 USE CASE ANALYSIS

To investigate the extent of COVID-19 disease in the selected population, as measured by positive neutralizer tests in all. Chen H suggested that pregnant women infected with the COVID-19 virus have restricted access to details. The aim of this study was to evaluate the clinical characteristics of COVID-19 in pregnant women. CT screening, according to Shan F, is critical for the diagnosis, evaluation, and organisation of COVID-19 disease. For illness movement, it is recommended that you check in every 3 to 5 days.Peripheral and bilateral Ground Glass Opacification (GGO) with consolidation will be more common in COVID-19 infected patients, according to the findings..

Koreans has similar characteristics to COVID-19 pneumonia in Chinese people, according to Yoon SH. When analysed for patients, they are found to have the same characteristics. Taking age structure into account, assessed the effect of social distancing on this pandemic. The papers provide a possible solution to the pandemic brought about by a new coronavirus. Subhankar Roy recently suggested the use of deep learning to classify Lung Ultrasonography (LUS) images. With the aid of the evolution-based CNN network, COVID-19 was classified using CT Chest images. Harrison X. Bai suggested that peripheral distribution, ground-glass opacity, and vascular thickening are the most important features for distinguishing COVID-19 from Viral pneumonia in a sample. Using CT scan images and CT scan features, this study can differentiate COVID-19 from viral pneumonia with high specificity. According to the same report, patients infected with COVID-19 and those infected with Viral pneumonia develop central+peripheral spread, air bronchogram, pleural thickening, pleural effusion, and lymphadenopathy with no substantial differences. Both viruses cause respiratory disease, which can be asymptomatic or mild, but can also lead to serious disease and death. Second, both viruses are spread through droplets or touch. These striking resemblances to pneumonia viruses (influenza and SARS-CoV-2) prompted us to pursue the proposed research. The following are the major drawbacks that have been identified so far

1. CT and CXR images are used: The COVID-19 virus targets cells in the respiratory tracts, mostly lung tissues; we can detect the virus without using a test kit by looking at photographs of Thorax. The chest X-Ray is thought to be of little use in the early stages, though CT scans of the chest are useful even before symptoms occur.

2. Shorter testing period: The test used to classify the COVID-19 is insufficiently long. It's difficult to evaluate patients, particularly in the early stages of a virus's growth. Radiologists must manually analyse CXR and CT scans of several patients, which takes a long time. As a result, we need an automated system that can save radiologist time.

3. Repurposing an existing model: In this paper, a novel existing method for detecting COVID-19 using CT scan and Chest X-Ray images is repurposed. This is capable of detecting the irregular features that have been detected in the images with pinpoint accuracy.

Here the Input Image is either COVID positive Chest X-Ray or Normal Chest X-Ray image. After removing the Noisy data from the dataset, it is passed to the CNN model for detection.

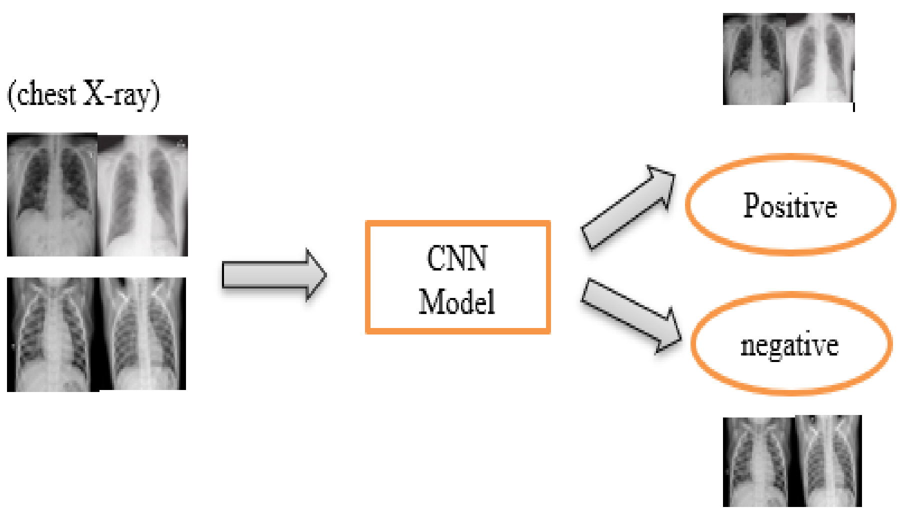


Fig 2.3.1 Basic Flow Diagram

2.4 REQUIREMENT SPECIFICATION

2.4.1 FUNCTIONAL REQUIREMENTS

The System should provide an input function which can take the chest X-Ray images and figure out the probable outcome

The System needs a Binary classifier to which is well trained to produce the desired output for the chest X-Ray Images

The system should extract all the images required to perfom analysis over all the set of given input

The System needs the remove unwanted data because the dataset might contain different variation of images belonging to the lungs such as posterior, anterior views etc. and removing the noisy data and passing into the model

The system should provide a complete functionality when the user uploads the chest X-Ray image for the classification and keras image data generator is used for the validation purpose.

The system should provide communication between the server and client with necessary network functions such sending the file and processing it

The system takes considerable amount of time to train the data and the model learns by itself when several images are validated

The System should provide taking new data from admin to train and classify whether the chest X-ray image is Covid Positive or Negative image. Once the output is ready ,it can be used by doctors for their assistance in finding the virus is present or not.

2.4.2 NON-FUNCTIONAL REQUIREMENTS

Usability:

The system should be easy to use. The user should reach the outcome with one button press if possible. Because one of the software's features is timesaving. The system also should be user friendly for admins because anyone can be admin instead of programmers. Training the data and classifiers are used too many times, so it is better to make it easy

Reliability:

The preprocessing step transforms the raw sourced data into a format that enables successful model training

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. Also, user provided data will be used to compare with result and measure reliability. With recent machine learning techniques, user gained data should be enough for reliability if enough data is obtained.

Performance:

On Training the model and testing it with various data the maximum accuracy obtained was 98.6%. Working on pre-trained model earlier has made the model to overfit so using CNN helped to resolve this issue and obtained a maximum accuracy.

Using Google Colab, with runtime as GPU trains the model faster and produce the result in very quick time

The system should require python knowledge for maintenance. Any problem acquire in server side and deep learning methods, it requires code knowledge and deep learning background to solve.

2.4.3 SOFTWARE REQUIREMENTS

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It can be used as a scripting language or can be compiled to byte-code for building large applications. It provides very high-level dynamic data types and supports dynamic type checking. It is widely used programming language for Machine Learning and Deep learning .

Anaconda-Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. Packages which are used inside this software includes numpy, pandas, keras, matplot etc are installed to extract the data.

CHAPTER-3

SYSTEM DESIGN

3.1 DETAILED DESIGN

It is a layered architecture. The sequential model will have a Conv2d layer with 32 number of filters at the beginning with the kernel size of 3 X 3 and relu as the activation function. As this is the first Layer the input size specified is 224 X 224 X 3.

In the second layer, the filters are increased to 64 and the conv2d layers are stacked on top of each other with the same filter size of 3 X 3 because this can reduce the overall weight parameters instead of using a single layer and it is followed by Maxpooling2d layer with a pool size of 2 X 2 and dropout layer which is used to avoid overfitting. Now the above-mentioned layers are repeated and stacked upon each other so that inner layers can easily detect the larger portion of the image. Once the above layers are created, we need to flatten the layer. The below-given figure gives a clear understanding of our CNN model.

In the output layer, we have a single neuron because this is a binary classification so we use sigmoid as the activation function. Finally compile the model with binary cross-entropy loss, optimizer used is Adam, with the metrics as Accuracy. It consists of more than 55 lakh parameters in total.

In the training part, Keras Image data generator library is used to make the data ready for the Model.

Augmentation is performed on the original image such as rescaling the image(Normalization), shear augmentation, zoom augmentation, and horizontal flip. similarly, in the testing part, we perform normalization and pass it to the image data generator library.

In the train data generator, the target size of 224 X 224 is specified because the model input should be similar to the target size that is being fed into the model, with the batch size of 32, and the mode of the class is binary.it is then passed into the model and similarly, the validation data generator needs to be specified with the same target size, batch size, and mode of class.

The final part is to train the model bypassing the trained generator, steps per epoch, no of epochs, validation generator. The model performed better as the no of epochs increases and the highest accuracy of 98.6% is recorded

The below given architecture diagram gives a clear understanding of our CNN model

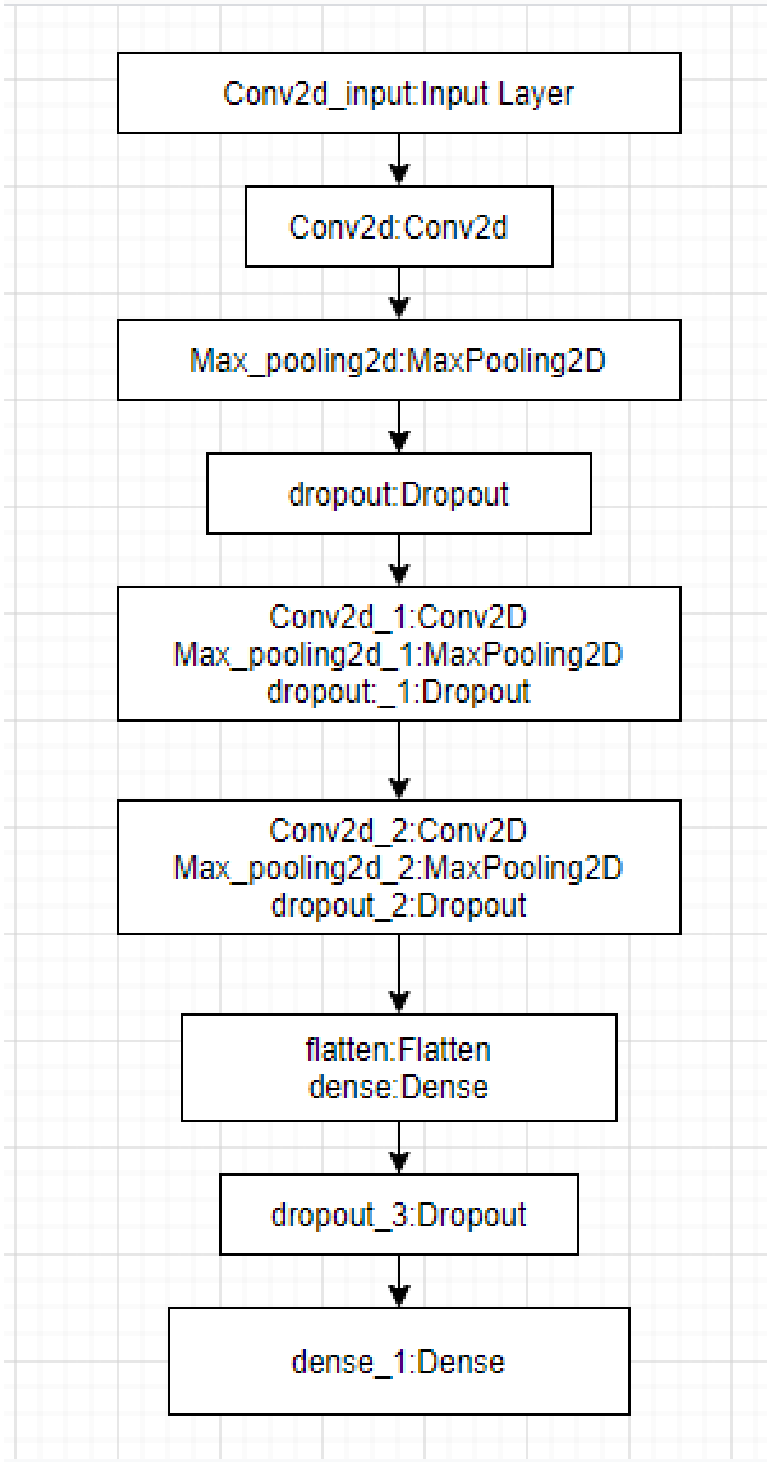


Fig 3.1.1 Architecture

3.2 DESIGN OF METHODOLOGY

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN image classifications takes an input image, process it and classify it under certain categories (Eg., Dog, Cat, Tiger, Lion). Computers sees an input image as array of pixels and it depends on the image resolution. Based on the image resolution, it will see h x w x d(h = Height, w = Width, d = Dimension). Eg., An image of 6 x 6 x 3 array of matrix of RGB (3 refers to RGB values) and an image of 4 x 4 x 1 array of matrix of grayscale image.

Dataset:

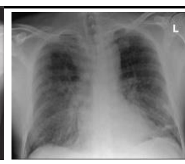
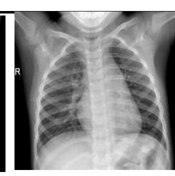
- ➔ Link to Covid chest X-ray images

https://github.com/ieee8023/covid-chestxray-dataset

- ➔ Link to Normal Chest X-Ray images

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

Table 3.2.1: Summary of Dataset

Dataset	No of COVID positive images	No of Normal images
Chest X-Ray	 	 
Training	160	160
Testing	36	36
Total	196	196

Technically, deep learning CNN models to train and test, each input image will pass it through a series of convolution layers with filters (Kernals), Pooling, fully connected layers (FC) and apply Softmax function to classify an object with probabilistic values between 0 and 1. The below figure is a complete flow of CNN to process an input image and classifies the objects based on values. Convolution is the first layer to extract features from an input image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

Stride is the number of pixels shifts over the input matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on. The below figure shows convolution would work with a stride of 2.

Padding Sometimes filter does not fit perfectly fit the input image. We have two options: Pad the picture with zeros (zero-padding) so that it fits .Drop the part of the image where the filter did not fit. This is called valid padding which keeps only valid part of the image.

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is $f(x) = \max(0,x)$. ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values.

The first block makes the particularity of this type of neural network since it functions as a feature extractor. To do this, it performs template matching by applying convolution filtering operations. The first layer filters the image with several convolution kernels and returns "feature maps", which are then normalized (with an activation function) and/or resized.

This process can be repeated several times: we filter the features maps obtained with new kernels, which gives us new features maps to normalize and resize, and we can filter again, and so on. Finally, the values of the last feature maps are concatenated into a vector. This vector defines the output of the first block and the input of the second.

Pooling layers section would reduce the number of parameters when the images are too large. Spatial pooling also called subsampling or downsampling which reduces the dimensionality of each map but retains important information. Spatial pooling can be of different types such as Max Pooling, Average Pooling, Sum Pooling

Max pooling takes the largest element from the rectified feature map. Taking the largest element could also take the average pooling. Sum of all elements in the feature map call as sum pooling.

The second block is not characteristic of a CNN: it is in fact at the end of all the neural networks used for classification. The input vector values are transformed (with several linear combinations and activation functions) to return a new vector to the output. This last vector contains as many elements as there are classes: element i represents the probability that the image belongs to class i. Each element is therefore between 0 and 1, and the sum of all is worth These probabilities are calculated by the last layer of this block (and therefore of the network), which uses a logistic function (binary classification) or a softmax function (multi-class classification) as an activation function.

As with ordinary neural networks, the parameters of the layers are determined by gradient backpropagation: the cross-entropy is minimized during the training phase. But in the case of CNN, these parameters refer in particular to the image features.

There are four types of layers for a convolutional neural network: the convolutional layer, the pooling layer, the ReLU correction layer and the fully-connected layer. The convolutional layer is the key component of convolutional neural networks, and is always at least their first layer

Its purpose is to detect the presence of a set of features in the images received as input. This is done by convolution filtering: the principle is to "drag" a window representing the feature on the image, and to calculate the convolution product between the feature and each portion of the scanned image. A feature is then seen as a filter: the two terms are equivalent in this context

The convolutional layer thus receives several images as input, and calculates the convolution of each of them with each filter. The filters correspond exactly to the features we want to find in the images.

We get for each pair (image, filter) a feature map, which tells us where the features are in the image: the higher the value, the more the corresponding place in the image resembles the feature.

Unlike traditional methods, features are not pre-defined according to a particular formalism (for example SIFT), but learned by the network during the training phase! Filter kernels refer to the convolution layer weights. They are initialized and then updated by backpropagation using gradient descent

A pre-learned features approach based on the learned visual features from the fully-connected convolution layers of the pretrained deep model is proposed. In the experimental results, this approach was proven to significantly contribute to the classification performance for diagnosis of COVID-19, both individually and in a combined format.

Like a traditional neural network, a CNN has neurons with weights and biases. The model learns these values during the training process, and it continuously updates them with each new training example. However, in the case of CNNs, the weights and bias values are the same for all hidden neurons in a given layer.

This means that all hidden neurons are detecting the same feature, such as an edge or a blob, in different regions of the image. This makes the network tolerant to translation of objects in an image. For example, a network trained to recognize cars will be able to do so wherever the car is in the image.

After learning features in many layers, the architecture of a CNN shifts to classification.The next-to-last layer is a fully connected layer that outputs a vector of K dimensions where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified.

The final layer of the CNN architecture uses a classification layer such as softmax to provide the classification output

3.3 MODULES

There are various modules that are used in the Deep learing model which inludes numpy,keras,matplot etc.

NumPy is a Python library used for working with arrays. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Keras is an Open Source Neural Network library written in Python that runs on top of Theano or Tensorflow. It is designed to be modular, fast and easy to use. Keras doesn't handle low-level computation. Instead, it uses another library to do it, called the "Backend.

There are multiple layers in the CNN architecture Diagram such as Conv2D,Maxpooling, dropout. The pooling operation consists in reducing the size of the images while preserving their important characteristics.

To do this, we cut the image into regular cells, then we keep the maximum value within each cell. In practice, small square cells are often used to avoid losing too much information. The most common choices are 2x2 adjacent cells that don’t overlap, or 3x3 cells, separated from each other by a step of 2 pixels (thus overlapping).

We get in output the same number of feature maps as input, but these are much smaller. The pooling layer reduces the number of parameters and calculations in the network. This improves the efficiency of the network and avoids over-learning.

The maximum values are spotted less accurately in the feature maps obtained after pooling than in those received in input — this is a big advantage! For example, when you want to recognize a dog, its ears do not need to be located as precisely as possible:

ReLU (Rectified Linear Units) refers to the real non-linear function defined by ReLU(x)=max(0,x)

The ReLU correction layer replaces all negative values received as inputs by zeros. It acts as an activation function.

The fully-connected layer is always the last layer of a neural network, convolutional or not — so it is not characteristic of a CNN.This type of layer receives an input vector and produces a new output vector. To do this, it applies a linear combination and then possibly an activation function to the input values received.

The last fully-connected layer classifies the image as an input to the network: it returns a vector of size N, where N is the number of classes in our image classification problem. Each element of the vector indicates the probability for the input image to belong to a class.
To calculate the probabilities, the fully-connected layer, therefore, multiplies each input element by weight, makes the sum, and then applies an activation function (logistic if N=2, softmax if N>2). This is equivalent to multiplying the input vector by the matrix containing the weights. The fact that each input value is connected with all output values explains the term fully-connected.

The convolutional neural network learns weight values in the same way as it learns the convolution layer filters: during the training phase, by backpropagation of the gradient.
The fully connected layer determines the relationship between the position of features in the image and a class. Indeed, the input table being the result of the previous layer, it corresponds to a feature map for a given feature: the high values indicate the location (more or less precise depending on the pooling) of this feature in the image. If the location of a feature at a certain point in the image is characteristic of a certain class, then the corresponding value in the table is given significant weight.

The parmetrization of the layers

A convolutional neural network differs from another by the way the layers are stacked, but also parameterized.The layers of convolution and pooling have indeed hyperparameters, that is to say parameters whose you must first define the value.The size of the output feature maps of the convolution and pooling layers depends on the hyperparameters.
Each image (or feature map) is W×H×D, where W is its width in pixels, H is its height in pixels and D the number of channels (1 for a black and white image, 3 for a colour image).
The convolutional layer has four hyperparameters:

1. The number of filters K
2. The size F filters: each filter is of dimensions F×F×D pixels.

3. The S step with which you drag the window corresponding to the filter on the image. For example, a step of 1 means moving the window one pixel at a time.
4. The Zero-padding P: add a black contour of P pixels thickness to the input image of the layer.

Without this contour, the exit dimensions are smaller. Thus, the more convolutional layers are stacked with P=0, the smaller the input image of the network is. We lose a lot of information quickly, which makes the task of extracting features difficult.For each input image of size W×H×D, the pooling layer returns a matrix of dimensions Wc×Hc×Dc, where:
Choosing P=F-1/2 and S=1 gives feature maps of the same width and height as those received in the input.
The pooling layer has two hyperparameters:
1. The size F of the cells: the image is divided into square cells of size F×F pixels.
2. The S step: cells are separated from each other by S pixels.

For each input image of size W×H×D, the pooling layer returns a matrix of dimensions Wp×Hp×Dp, where:
Just like stacking, the choice of hyperparameters is made according to a classic scheme:

- For the convolution layer, the filters are small and dragged on the image one pixel at a time. The zero-padding value is chosen so that the width and height of the input volume are not changed at the output. In general, we then choose F=3,P=1,S=1 or F=5,P=2,S=1
- For pooling layer, F=2 and S=2 is a wise choice. This eliminates 75% of the input pixels.We can also choose F=3 and S=2: in this case, the cells overlap. Choosing larger cells causes too much loss of information

In the output layer, we have a single neuron because this is a binary classification so we use sigmoid as the activation function. Finally compile the model with binary cross-entropy loss, optimizer used is Adam, with the metrics as Accuracy. It consists of more than 55 lakh parameters in total. In the training part, Keras Image data generator library is used to make the data ready for the Model.

CHAPTER-4

SYSTEM IMPLEMENTATION

4.1 MODULE IMPLEMENTATION

In the training part, Keras Image data generator library is used to make the data ready for the Model.

Augmentation is performed on the original image such as rescaling the image(Normalization), shear augmentation, zoom augmentation, and horizontal flip. similarly, in the testing part, we perform normalization and pass it to the image data generator library.

In the train data generator, the target size of 224 X 224 is specified because the model input should be similar to the target size that is being fed into the model, with the batch size of 32, and the mode of the class is binary.it is then passed into the model and similarly, the validation data generator needs to be specified with the same target size, batch size, and mode of class.

The final part is to train the model bypassing the trained generator, steps per epoch, no of epochs, validation generator. The model performed better as the no of epochs increases and the highest accuracy of 98.6% is recorded

Experimental setup

This experiment is carried out both in the local machine as well as in Google Colab. Collecting all the data from the open-source repositories and removing the noisy data is carried out in Jupyter Notebook(local machine) and for training the model, Google Colab with runtime environment as GPU is used for producing faster results. A total of 20 minutes is taken to complete the training of this model.

Experiment Analysis

To check the performance of the model we have carried out a few analysis of various datasets which are different from the other normal chest X-Ray images. In the first Dataset consisting of 224 images of both the classes are taken for the training dataset and 60 images for validation. The accuracy obtained for the Training dataset is 96 and the validation dataset is 98. After a lot of understanding, the size of the dataset is increased.

In the second part of the Analysis, we have done some augmentation and increased the dataset size. A total of 320 images are taken into consideration for Training and 72images for Testing. The reason behind taken in the equal ratio is to make our model generalized without overfitting.

The below table shows the results after performing with various datasets.

Table 4.1.1: Experiment Analysis

Dataset (Total size)	Training Accuracy	Validation Accuracy
A(284)	96.4	98.3
B(392)	93.4	98.6

So the maximum Accuracy obtained for training data set B is considered for evaluation as it consists of more images to the previous dataset and the validation accuracy is high.

4.2 TESTING

When working initially in the local machine(Anaconda- Jupyter Notebook) ,few modules were missing even after downloading all packages .Later switching to Google Colab has helped to resolve this issue

Datasets related to COVID-19 positive chest X-Rays are not available properly in the public repositories. After Exploring , a Public repository is found consisting of all the chest X-Ray images. So removing all the noisy data and collecting Chest X-Ray images of only positive and negative cases are taken.

Confusion Matrix:

Confusion matrix helps in performance measurement for classification of machine learning models and it can be plotted For performance evaluation, the adopted Accuracy (ACC), Specificity (SP), and Sensitivity (SN) metrics. They are defined as:

Accuracy(ACC) = (T P + T N)/N

Sensitivity (SN) = T P /(T P + F N)

Specificity (SP) = T N /(T N + FP)

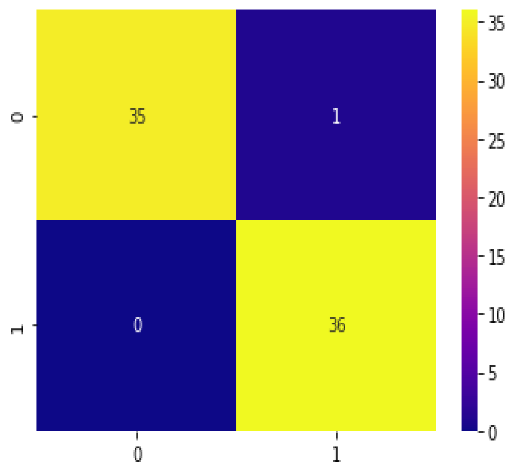
Where TP=True Positive,

TN=True Negative

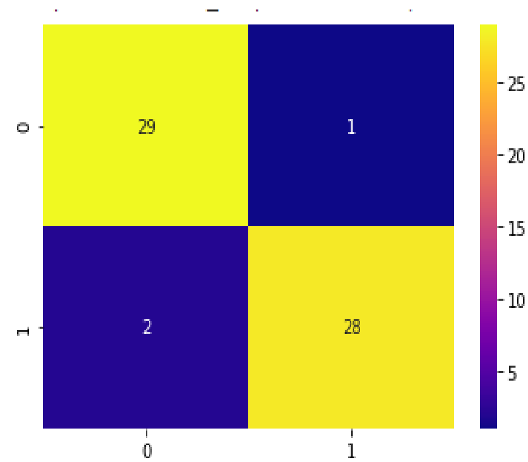
FP=False Positive

FN=False Negative

The validation data set consisting of 36 chest X-Ray images of Covid patients and 36 Chest X-Ray images of normal patients and also comparing with the previous data set having 30 chest X-ray images of Covid patients and 36 Chest X-Ray images of normal patients



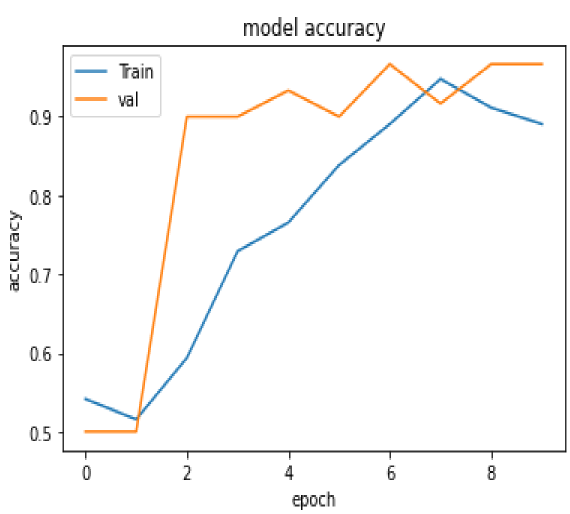
4.2.1 ConfusionMatrix of Dataset A(Table 4.1.1)



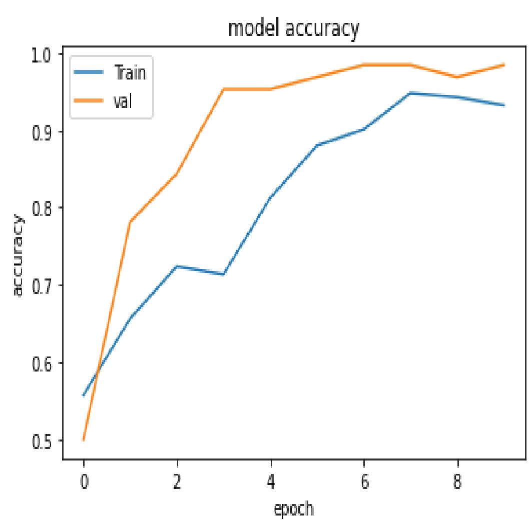
4.2.2 ConfusionMatrix of Dataset B(Table 4.1.1)

Using the Dataset A and B the performance analysis is here, The model has predicted 35 chest X-Ray images as covid positive out of 36, and similarly It has predicted 36 Chest X-Ray images as normal out of 36.

Validation Accuracy

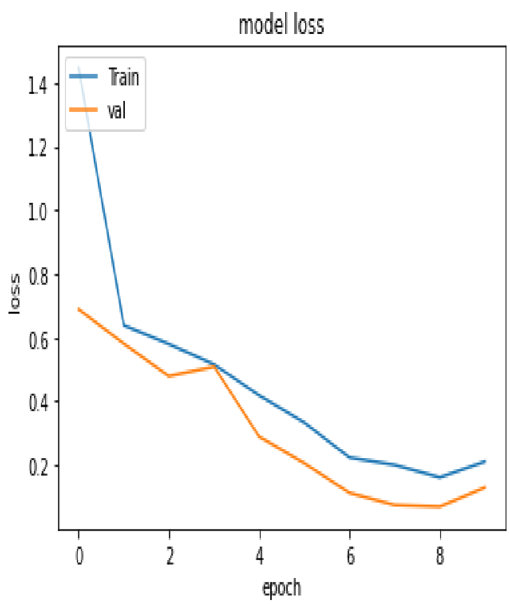


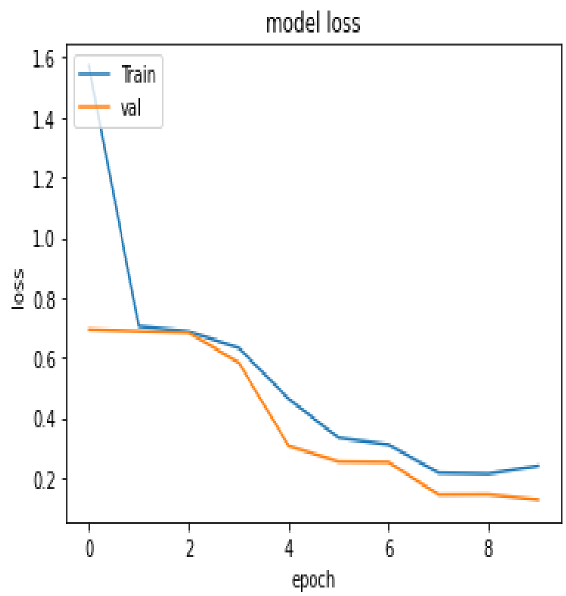
4.2.3 ValidationAccuracy of Dataset A(Table 4.1.1)



4.2.4 Validation Accuracy of Dataset B(Table 4.1.1)

Validation Loss





4.2.5 Validation Loss of Dataset A(Table 4.1.1)   4.2.6 Validation Loss of Dataset B(Table 4.1.1)

Comparing with other pre-trained networks

Our model has Achieved an Accuracy of 98.6% with the Dataset Consisting of 392 chest X-Ray images which include both Normal and Covid chest X-Rays. Below table shows the results of other pre-trained models that have been performed on various datasets from various locations.

Table 4.2.7: Performance Analysis of Various Methods

Technique	Dataset	Performance evaluation
Resnet50 and VGG16 [9]	102 both positive COVID-19 cases and pneumonia cases	An accuracy of 89.2% and an AUC of 0.95 is obtained.
Inception-Net [10]	73 COVID-19 positives and 340 healthy images from the Shenzhen, China collections	The Inception Net model achieved an accuracy of 99.96% and AUC of 1 in classifying COVID- 19 positive cases from combined pneumonia and normal patients.

Generative Adversarial Networks, AlexNet, GoogleNet, ResNet 18 and squeezeNet [11]	5863 images divided into two classes: normal and pneumonia	ResNet18 achieved precision, recall, and F1 score of 98.97%.
ResNet50, InceptionV3 and Inception- ResNetV2 [12]	50 images of each COVID-19 positive cases and normal cases	An accuracy of 98% obtained with ResNet50, 97% accuracy with InceptionV3, and Inception-ResNetV2 gives 87% accuracy
CNN models namely, AlexNet, VGG16, VGG19, GoogleNet, and ResNet50 [13]	349 COVID positive and 397 Non-COVID CT scan	ResNet50 is the best performing model and achieved 82.91% testing accuracy.
Joint Classification and Segmentation (JCS) [14]	Collected large scale COVID-19 dataset with 144,167 images of 400 COVID-19 patients and 350 Non-COVID cases.	JCS system provided an average sensitivity of 95% and 93% specificity for the classification, and dice score of 78.3% on the segmentation test set.
Deep learning-based model (composed of three components: a) backbone network, b) classification head, and anomaly detection [15]	The binary classification of a person into COVID-19 affected or not.	The proposed model outperforms from other CNN models in terms of F-measure, sensitivity, specificity, Kappa statistics, accuracy by 2.09%, 1.82%, 1.68%, 1.92%, and 1.97%, respectively.

Random forest (RF) model [16]	176 chest CT images of COVID-19 positive cases.	For the COVID-19 severity detection, the accuracy achieved is 87.5%, AUC is 0.91, and True positive rate (TP) is 93.3%.
Otsu based method [17]	90 slices of coronal-view and 20 slices of axial-view of lungs	COVID-19 pneumonia infection and its rate are detected in CT scan images of both coronal and axial views.
3-D CNN model (Residual networks) [18]	528/90 (Training and validation data: COVID-19 = 189, Influenza-A viral pneumonia = 194, and normal= 145) (Testing data: COVID-19 = 30, Influenza-A viral pneumonia= 30, and normal= 30)	AUC is 0.996, 98.2% sensitivity and 92.2% specificity.

CHAPTER-5
CONCLUSION AND FUTURE SCOPE

The proposed model detects COVID-19 from chest X-Ray using CNN. Dataset consisting of Chest X-Ray images are taken as input and passed to the CNN model. Before passing the data, Augmentation is performed on the training dataset. The Convolutional Neural Network model consisting of multiple layers such as conv2d, max-pooling, dropout, etc. As we go deeper into the network, the receptive field increases, and inner layers can capture a larger portion of the complex image. we perform normalization and pass it to the image data generator library. In the train data generator, the target size of 224 X 224 is specified because the model input should be similar to the target size that is being fed into the model, with the batch size of 32, and the mode of the class is binary.it is then passed into the model and similarly, the validation data generator needs to be specified with the same target size, batch size, and mode of class. The final part is to train the model bypassing the trained generator, steps per epoch, no of epochs, validation generator. The model performed better as the no of epochs increases .The accuracy obtained for the Validation data is 98.6%

Future work will involve the collection of more Chest X-Ray images from the early stage, improving the accuracy by fine-tuning all the layers in the pre-trained model so that the weights of the pre-trained model match with the given dataset and designing an end to end software model

APPENDIX-1

Extracting the COVID Chest X-Ray images from GitHub repository

```
import pandas as pd
import numpy as np

filepath="covid-chestxray-dataset-master/metadata.csv"
imagespath="covid-chestxray-dataset-master/images"

df=pd.read_csv(filepath)
print(df.shape)
(950, 30)

df.head()
```

	patientid	offset	sex	age	finding	RT_PCR_positive	survival	intubated	intubation_present	went_icu	...	date	location	folder	
0	2	0.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 22, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	2020_01_28_23_51
1	2	3.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 25, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	2020_01_28_23_51
2	2	5.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 27, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	2020_01_28_23_51
3	2	6.0	M	65.0	Pneumonia/Viral/COVID-19	Y	Y	N	N	N	...	January 28, 2020	Cho Ray Hospital, Ho Chi Minh City, Vietnam	images	2020_01_28_23_51
4	4	0.0	F	52.0	Pneumonia/Viral/COVID-19	Y	NaN	N	N	N	...	January 25, 2020	Changhua Christian Hospital, Changhua City, Ta...	images	

5 rows × 30 columns

```
import os
import shutil

targetdir="NewDataset/Covid"
```

```
if not os.path.exists(targetdir):
    os.mkdir(targetdir)
    print("covid folder created")
```

covid folder created

```
for(i,row) in df.iterrows():
    print(i,row)
```

```
0 patientid                                                                  2
offset                                                                      0
sex                                                                         M
age                                                                        65
finding                                              Pneumonia/Viral/COVID-19
RT_PCR_positive                                                             Y
survival                                                                    Y
intubated                                                                   N
intubation_present                                                          N
went_icu                                                                    N
in_icu                                                                      N
needed_supplemental_O2                                                      Y
extubated                                                                 NaN
temperature                                                               NaN
pO2_saturation                                                            NaN
leukocyte_count                                                           NaN
neutrophil_count                                                          NaN
lymphocyte_count                                                          NaN
view                                                                       PA
modality                                                                X-ray
date                                                         January 22, 2020
location                           Cho Ray Hospital, Ho Chi Minh City, Vietnam
folder                                                                 images
filename                          auntminnie-a-2020_01_28_23_51_6665_2020_01_28_...
doi                                                        10.1056/nejmc2001272
url                               https://www.nejm.org/doi/full/10.1056/NEJMc200...
license                                                                   NaN
clinical_notes                    On January 22, 2020, a 65-year-old man with a ...
other_notes                                                               NaN
```

```
Unnamed: 29                                                   NaN
Name: 5, dtype: object
6 patientid                                                     5
offset                                                        NaN
sex                                                           NaN
age                                                           NaN
finding                                                 Pneumonia
RT_PCR_positive                                               NaN
survival                                                      NaN
intubated                                                       Y
intubation_present                                              Y
went_icu                                                        Y
in_icu                                                          Y
needed_supplemental_O2                                        NaN
extubated                                                     NaN
temperature                                                   NaN
pO2_saturation                                                NaN
leukocyte_count                                               NaN
neutrophil_count                                              NaN
lymphocyte_count                                              NaN
view                                                           PA
modality                                                    X-ray
date                                                         2017
location                                                      NaN
folder                                                     images
filename                                          ARDSSevere.png
doi                                                           NaN
url            https://en.wikipedia.org/wiki/File:ARDSSevere.png
```

```
count=0

for (i,row) in df.iterrows():
    if row["finding"]=="Pneumonia/Viral/COVID-19":
        count+=1
print("no of covid images ",count)
```

```
no of covid images  584
```

```
count=0
for (i,row) in df.iterrows():
    if row["finding"]=="Pneumonia/Viral/COVID-19" and row["view"]=="PA":
        count+=1
print("no of covid images that are PA ",count)
```

```
no of covid images that are PA  196
```

```
count=0
for (i,row) in df.iterrows():
    if row["finding"]=="Pneumonia/Viral/COVID-19" and row["view"]=="PA":
        filename=row["filename"]
        curr_imagepath=os.path.join(imagespath,filename)
        image_copy_path=os.path.join(targetdir,filename)
        shutil.copy2(curr_imagepath,image_copy_path)
        count+=1
print("no of covid images ",count)
```

```
no of covid images  196
```

Extracting the Normal Chest X-Ray images from GitHub repository

```
import random
kaggle_file_path="chest_xray_kaggle/train/NORMAL"
target_normal_dir="NewDataset/Normal"


image_names=os.listdir(kaggle_file_path)
random.shuffle(image_names)


for i in range(196):
    image_name=image_names[i]
    image_path=os.path.join(kaggle_file_path,image_name)
    target_path=os.path.join(target_normal_dir,image_name)

    shutil.copy2(image_path,target_path)
    print("copying image ",i)
```

Using Goole Colab to train the CNN Model:

The Dataset belonging to covid and normal chest x rays have been uploaded to the google drive and and the contents are fetched directly

```
from google.colab import drive
drive.mount('/content/drive')


TRAIN_PATH = "/content/drive/My Drive/updatedDataset/Train"
VAL_PATH = "/content/drive/My Drive/updatedDataset/Val"
```

```
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'])
```

APPENDIX-2

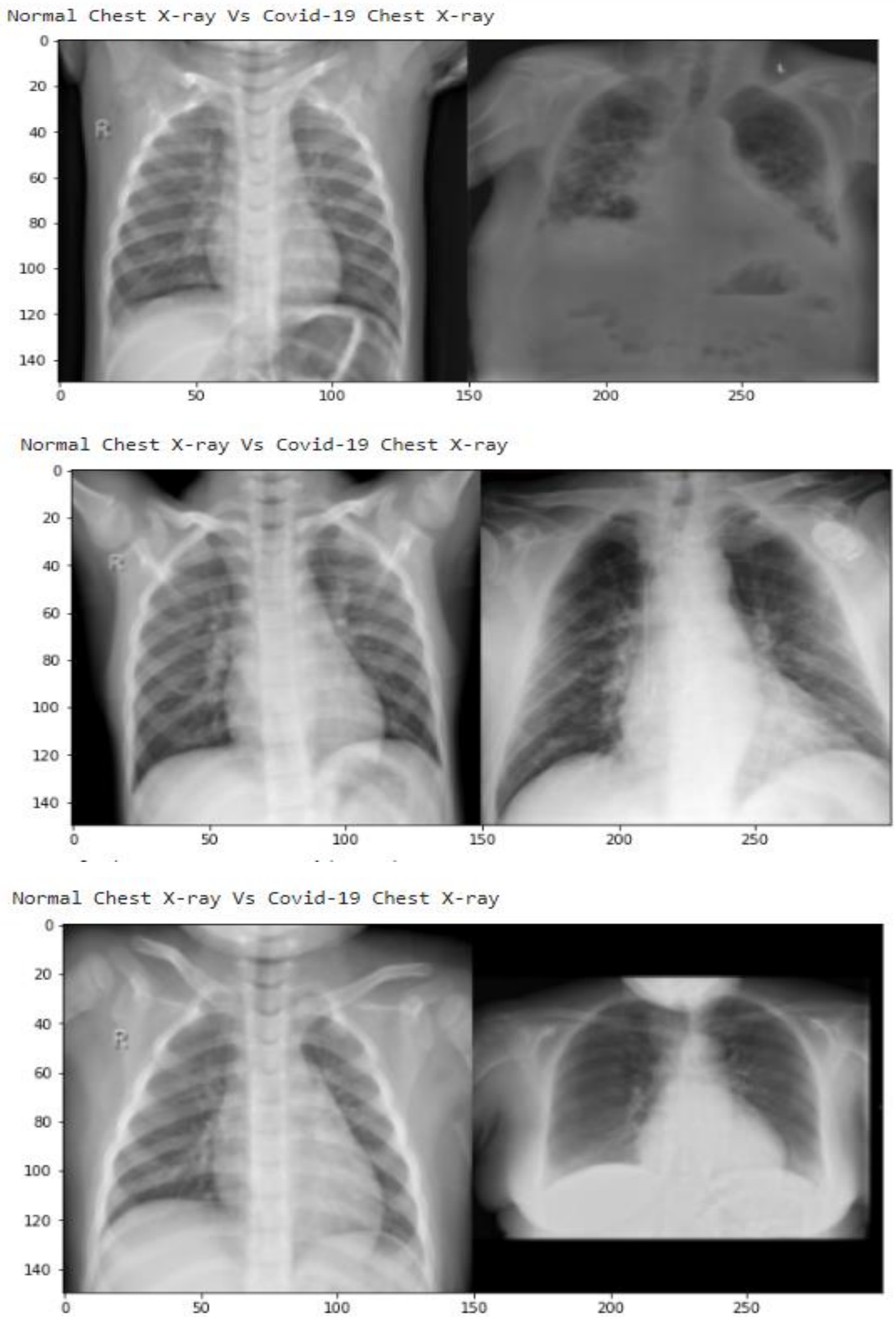


Fig 7.1 comaparison of Normal Chest X-Ray and Covid Chest X-Ray

model.summary()

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 222, 222, 32)      896
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 220, 220, 64)      18496
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 110, 110, 64)      0
_________________________________________________________________
dropout (Dropout)            (None, 110, 110, 64)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 108, 108, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 54, 54, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 54, 54, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 52, 52, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 26, 26, 128)       0
_________________________________________________________________
flatten (Flatten)            (None, 86528)             0
_________________________________________________________________
dense (Dense)                (None, 64)                5537856
_________________________________________________________________
dropout_3 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 65
=================================================================
Total params: 5,668,097
Trainable params: 5,668,097
Non-trainable params: 0
_________________________________________________________________
```

```
train_datagen = image.ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
)


test_dataset = image.ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    '/content/drive/My Drive/updatedDataset/Train',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
```

```
Found 320 images belonging to 2 classes.
```

```
train_generator.class_indices
```

```
{'Covid': 0, 'Normal': 1}
```

```
validation_generator = test_dataset.flow_from_directory(
    '/content/drive/My Drive/updatedDataset/Val',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
```

```
Found 72 images belonging to 2 classes.
```

```
validation_generator.class_indices
```

```
{'Covid': 0, 'Normal': 1}
```

```
hist = model.fit_generator(
    train_generator,
    steps_per_epoch=6,
    epochs = 10,
    validation_data = validation_generator,
    validation_steps=2
)
```

WARNING:tensorflow:From <ipython-input-13-e660bfd9e6ac>:6: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.

Epoch 1/10
6/6 [==============================] - 61s 10s/step - loss: 1.4461 - accuracy: 0.5573 - val_loss: 0.6889 - val_accuracy: 0.5000

Epoch 2/10
6/6 [==============================] - 26s 4s/step - loss: 0.6383 - accuracy: 0.6562 - val_loss: 0.5808 - val_accuracy: 0.7812

Epoch 3/10
6/6 [==============================] - 13s 2s/step - loss: 0.5796 - accuracy: 0.7240 - val_loss: 0.4789 - val_accuracy: 0.8438

Epoch 4/10
6/6 [==============================] - 10s 2s/step - loss: 0.5156 - accuracy: 0.7135 - val_loss: 0.5086 - val_accuracy: 0.9531

Epoch 5/10
6/6 [==============================] - 8s 1s/step - loss: 0.4183 - accuracy: 0.8125 - val_loss: 0.2888 - val_accuracy: 0.9531

Epoch 6/10
6/6 [==============================] - 8s 1s/step - loss: 0.3329 - accuracy: 0.8802 - val_loss: 0.2056 - val_accuracy: 0.9688

Epoch 7/10
6/6 [==============================] - 8s 1s/step - loss: 0.2234 - accuracy: 0.9010 - val_loss: 0.1122 - val_accuracy: 0.9844

Epoch 8/10
6/6 [==============================] - 8s 1s/step - loss: 0.2004 - accuracy: 0.9479 - val_loss: 0.0741 - val_accuracy: 0.9844

Epoch 9/10
6/6 [==============================] - 8s 1s/step - loss: 0.1610 - accuracy: 0.9427 - val_loss: 0.0692 - val_accuracy: 0.9688

Epoch 10/10
6/6 [==============================] - 8s 1s/step - loss: 0.2108 - accuracy: 0.9323 - val_loss: 0.1292 - val_accuracy: 0.9844

model.save("/content/drive/My Drive/model/covid_cnn_new.h5")

```
model.evaluate_generator(train_generator)
```

```
WARNING:tensorflow:From <ipython-input-15-e4ade065aa26>:1: Model.evaluate_generator (from tensorflow.python.keras.engine.training)
Instructions for updating:
Please use Model.evaluate, which supports generators.
[0.2067764699459076, 0.934374988079071]
```

Fig 7.2 Training Accuracy

```
model.evaluate_generator(validation_generator)
```

```
[0.12972643971443176, 0.9861111044883728]
```

```
[ ] model=load_model('/content/drive/My Drive/model/covid_cnn_new.h5')
```

Fig 7.3 Validation Accuracy

```
model.evaluate_generator(train_generator)
```

```
WARNING:tensorflow:From <ipython-input-13-e4ade065aa26>:1: Model.evaluate_generator
Instructions for updating:
Please use Model.evaluate, which supports generators.
[0.23002904653549194, 0.9280821681022644]
```

```
[ ] model.evaluate_generator(validation_generator)
```

```
[0.16180285811424255, 0.9700000286102295]
```

```
[ ] model=load_model('/content/drive/My Drive/model/covid_cnn_new1.h5')
```

```
[ ] import os
    train_generator.class_indices
```

```
{'Covid': 0, 'Normal': 1}
```

```
[ ] model=load_model('/content/drive/My Drive/model/covid_cnn_new.h5')
```

```
import os
train_generator.class_indices
```

```
{'Covid': 0, 'Normal': 1}
```

```
[ ] y_actual=[]
    y_test=[]
```

```
[ ] for i in os.listdir("/content/drive/My Drive/updatedDataset/Val/Normal/"):
      img=image.load_img("/content/drive/My Drive/updatedDataset/Val/Normal/"+i,target_size=(224,224))
      img=image.img_to_array(img)
      img=np.expand_dims(img,axis=0)
      p=model.predict_classes(img)
      y_test.append(p[0,0])
      y_actual.append(1)
```

```
[ ] for i in os.listdir("/content/drive/My Drive/updatedDataset/Val/Covid/"):
      img=image.load_img("/content/drive/My Drive/updatedDataset/Val/Covid/"+i,target_size=(224,224))
      img=image.img_to_array(img)
      img=np.expand_dims(img,axis=0)
      p=model.predict_classes(img)
      y_test.append(p[0,0])
      y_actual.append(0)
```


```
      y_test.append(p[0,0])
      y_actual.append(0)
```


```
[ ] y_actual=np.array(y_actual)
    y_test=np.array(y_test)
```

```
[ ] from sklearn.metrics import confusion_matrix
    cm=confusion_matrix(y_actual,y_test)
```

```
[ ] import seaborn as sns
    sns.heatmap(cm,cmap="plasma",annot=True)
```

```python
import matplotlib.pyplot as plt
```

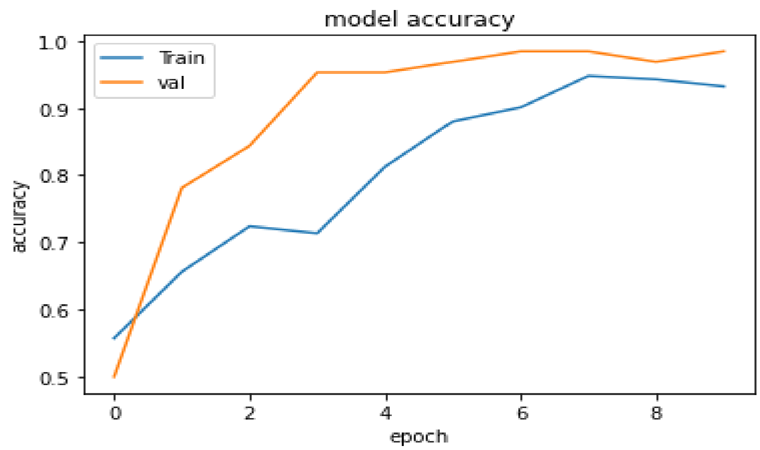
```python
[ ] hist.history
```

```
{'accuracy': [0.5572916865348816,
  0.65625,
  0.7239583134651184,
  0.7135416865348816,
  0.8125,
  0.8802083134651184,
  0.9010416865348816,
  0.9479166865348816,
  0.9427083134651184,
  0.9322916865348816],
 'loss': [1.4460757970809937,
  0.6383071541786194,
  0.5796326398849487,
  0.5155718922615051,
  0.418318782764435,
  0.33294251561164856,
  0.22337965667247772,
  0.20043669641017914,
  0.16096900403499603,
  0.2107647806406021],
```

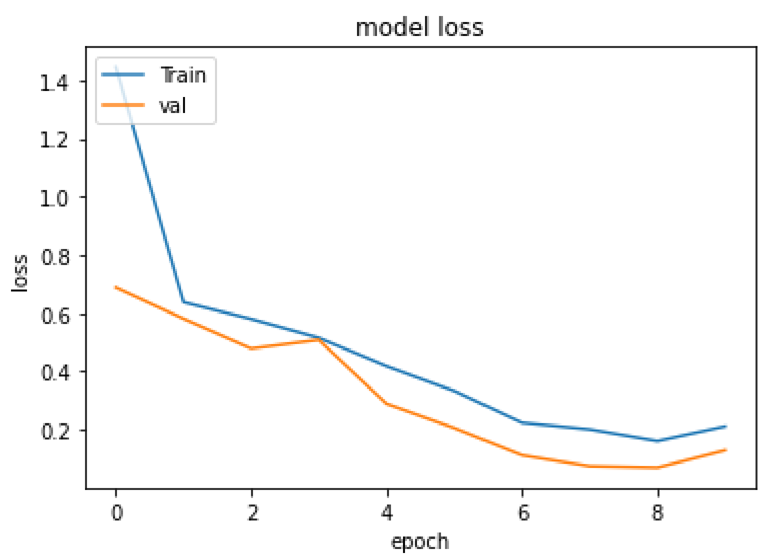
0.2107647806406021],

```
 'val_accuracy': [0.5,
  0.78125,
  0.84375,
  0.953125,
  0.953125,
  0.96875,
  0.984375,
  0.984375,
  0.96875,
  0.984375],
 'val_loss': [0.6888501048088074,
  0.5807641744613647,
  0.478858083486557,
  0.5086323022842407,
  0.28875482082366943,
  0.20562082529067993,
  0.11216838657855988,
  0.0741429328918457,
  0.06924157589673996,
  0.12917011976242065]}
```

```
[ ] plt.plot(hist.history['accuracy'])
    plt.plot(hist.history['val_accuracy'])
    plt.title("model accuracy")
    plt.ylabel("accuracy")
    plt.xlabel("epoch")
    plt.legend(['Train','val'],loc='upper left')
    plt.show()
```




```
[ ] plt.plot(hist.history['loss'])
    plt.plot(hist.history['val_loss'])
    plt.title("model loss")
    plt.ylabel("loss")
    plt.xlabel("epoch")
    plt.legend(['Train','val'],loc='upper left')
    plt.show()
```



```
[ ] from sklearn.metrics import classification_report
    print(classification_report(y_actual,y_test))
```

```
              precision    recall  f1-score   support

           0       1.00      0.97      0.99        36
           1       0.97      1.00      0.99        36

    accuracy                           0.99        72
   macro avg       0.99      0.99      0.99        72
weighted avg       0.99      0.99      0.99        72
```

```
[ ] s=input()
```

```
/content/drive/MyDrive/Dataset/Covid/0a7faa2a.jpg
```

```
[ ] img=image.load_img(s,target_size=(224,224))
    img=image.img_to_array(img)
    img=np.expand_dims(img,axis=0)
    p=model.predict_classes(img)
    p[0,0]
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:450: UserWarning: `model.predict_classes()`
  warnings.warn('`model.predict_classes()` is deprecated and '
0
```

REFERENCES

[1] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "COVID-CAPS: A capsule network-based framework for identification of COVID-19 cases from X-ray images," *Pattern Recognit. Lett.*, vol. 138, pp. 638–643, 2020, doi: 10.1016/j.patrec.2020.09.010.

[2] E. Tartaglione, C. A. Barbano, C. Berzovini, M. Calandri, and M. Grangetto, "Unveiling COVID-19 from chest x-ray with deep learning: A hurdles race with small data," *Int. J. Environ. Res. Public Health*, vol. 17, no. 18, pp. 1–17, 2020, doi: 10.3390/ijerph17186933.

[3] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network," *Appl. Intell.*, 2020, doi: 10.1007/s10489-020-01829-7.

[4] G. Maguolo and L. Nanni, "A Critic Evaluation of Methods for COVID-19 Automatic Detection from X-Ray Images," *arXiv*, pp. 1–10, 2020.

[5] I. D. Apostolopoulos and T. A. Mpesiana, "Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Phys. Eng. Sci. Med.*, vol. 43, no. 2, pp. 635–640, 2020, doi: 10.1007/s13246-020-00865-4.

[6] M. Turkoglu, "COVIDetectioNet: COVID-19 diagnosis system based on X-ray images using features selected from pre-learned deep features ensemble," *Appl. Intell.*, 2020, doi: 10.1007/s10489-020-01888-w.

[7] D. Das, K. C. Santosh, and U. Pal, "Truncated inception net: COVID-19 outbreak screening using chest X-rays," *Phys. Eng. Sci. Med.*, vol. 43, no. 3, pp. 915–925, 2020, doi: 10.1007/s13246-020-00888-x.

[8] Z. Wang, Q. Liu, and Q. Dou, "Contrastive Cross-Site Learning with Redesigned Net for COVID-19 CT Classification," *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 10, pp. 2806–2813, 2020, doi: 10.1109/JBHI.2020.3023246.

[9] Hall L O, Paul R, Goldgof D B, Goldgof G M (2020) Finding covid-19 from chest x-rays using deep learning on a small dataset

[10] Das D, Santosh K C, Pal U (2020) Truncated inception net: Covid-19 outbreak screening using chest x-rays. Research Square pp 1–11

[11] Khalifa N E M, Taha M H N, Hassanien A E, Elghamrawy S (April 2020) Detection of

Coronavirus (COVID-19) Associated Pneumonia based on Generative Adversarial Networks and a Fine-Tuned Deep Transfer Learning Model using Chest X-ray Dataset.

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

[13] Loey M, Smarandache F, Khalifa N E M (2020) A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images. Preprints

[14] Wu Y-H, Gao S-H, Mei J, Xu J, Fan D-P, Zhao C-W, Cheng M-M (2020) Jcs: An explainable covid-19 diagnosis system by joint classification and segmentation

[15] Zhang J, Xie Y, Li Y, Shen C, Xia Y (2020) Covid-19 screening on chest x-ray images using deep learning based anomaly detection 2003.12338

[16] Tang Z, Zhao W, Xie X, Zhong Z, Shi F, Liu J, Shen D (2020) Severity assessment of coronavirus disease 2019 (covid-19) using quantitative features from chest ct images

[17] Rajinikanth V, Dey N, Raj A N J, Hassanien A E, Santosh K C et al (2020) Harmony-search and otsu based system for coronavirus disease (covid-19) detection using lung ct scan images 2004.03431

[18] Xu X, Jiang X, Ma C, Du P, Li X, Lv S, Yu L, Chen Y, Su J, Lang G, Li Y, Zhao H, Xu K, Ruan L, Wu W (2020) Deep learning system to screen coronavirus disease 2019 pneumonia