A Project Report

on

Covid-19 Detection using CNN with X-ray images

Submitted in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

by

A. BHAVYA SREE **(184G1A0507)**

B. GOWTHAM REDDY **(184G1A0519)**

A. JYOTHI **(184G1A0527)**

K. NANDINI **(184G1A0520)**

Under the Guidance of

Dr. M. Ranjit Reddy, M.Tech., Ph.D.

Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY:ANANTAPURAMU

(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE&CSE))

Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.

2021-2022

SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY:ANANTAPURAMU
(Affiliated to JNTUA, Accredited by NAAC with 'A' Grade, Approved by AICTE, New Delhi & Accredited by NBA (EEE, ECE & CSE))
Rotarypuram village, B K Samudram Mandal, Ananthapuramu-515701.

Certificate

This is to certify that the project report entitled **Covid-19 Detection using CNN with X-ray images** is the bonafide work carried out by **Bhavya sree A** bearing Roll Number **184G1A0507** in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science & Engineering** during the academic year 2021-2022.

Guide

Dr. M. Ranjit Reddy, M.Tech., Ph.D
Professor

Head of the Department

Mr. P. Veera Prakash M.Tech,(Ph.D)
Assistant Professor & HOD

Date:
Place: Ananthapuramu

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that we now have the opportunity to express my gratitude for all of them.

It is with immense pleasure that we would like to express my indebted gratitude to my Guide **Dr. M. Ranjit Reddy, M.Tech., Ph.D., Professor, Computer Science & Engineering**, who has guided me a lot and encouraged me in every step of the project work. We thank him for the stimulating guidance, constant encouragement and constructive criticism which have made possible to bring out this project work.

We express our deep-felt gratitude to **Mr. K.Venkatesh M.Tech., , Assistant Professor,** project coordinators valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We are very much thankful to **Mr. P. Veera Prakash M.Tech,(Ph.D) ,Assistant professor & Head of the Department, Computer Science & Engineering,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey my special thanks to **Dr.G.Bala Krishna,M.Tech,Ph.D Principal** of **Srinivasa Ramanujan Institute of Technology** for giving the required information in doing our project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported us in completing our project in time.

We also express our sincere thanks to the Management for providing excellent facilities.

Finally, we wish to convey our gratitude to our family who fostered all the requirements and facilities that we need.

Project Associates

DECLARATION

We, Ms A.Bhavya Sree with reg no: 184G1A0507 , Mr B.Gowtham Reddy with reg no: 184G1A0519, Ms A.Jyothi with reg no: 184G1A0527 , Ms. K.Nandini with reg no: 184G1A0551 students of SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY , Rotarypuram , hereby declare that the dissertation entitled "COVID-19 DETECTION USING CNN WITH X-RAY IMAGES" embodies the report of our project work carried out by us during IV year Bachelor of Technology under the guidance of Dr. M. Ranjit Reddy, M.Tech., Ph.D. Department of CSE, SRINIVASA RAMANUJAN INSTITUTE OF TECHNOLOGY, and this work has been submitted for the partial fulfillment of the requirements for the award of the Bachelor of Technology degree.

The results embodied in this project have not been submitted to any other University of Institute for the award of any Degree or Diploma.

A. BHAVYA SREE	Reg no: 184G1A0507
B. GOWTHAM REDDY	Reg no: 184G1A0519
A. JYOTHI	Reg no: 184G1A0527
K. NANDINI	Reg no: 184G1A0551

CONTENTS **Page No.**

List of Figures

List of Abbreviations

UML	Unified Modeling Language
CNN	Convolutional Neural Network
SVM	Support Vector Machine

LIST OF TABLES

ABSTRACT

The COVID-19 pandemic is causing a major outbreak in more than 150 countries around the world, having a severe impact on the health and life of many people globally. One of the crucial step in fighting COVID-19 is the ability to detect the infected patients early enough, and put them under special care. Detecting this disease from radiography and radiology images is perhaps one of the fastest ways to diagnose the patients. Some of the early studies showed specific abnormalities in the chest radiograms of patients infected with COVID-19. Inspired by earlier works, we study the application of deep learning models to detect COVID-19 patients from their chest radiography images. We have first considered the dataset of chest X-Ray images which were divided into training and testing. Once after considering the dataset, we have implemented CNN algorithm which is an deep learning algorithm that which is used to train the dataset that will be used to classify the Covid positive and negative classes. Post training we are using PyQT5 to create an application for results testing.

Keywords: Covid-19, Lung CT Scan images. Deep Learning, CNN, RNN, LSTM, SVM, Ensemble Algorithm.

CHAPTER - 1

INTRODUCTION

Covid-19 is an inflammatory condition of the lung primarily affecting the small air sacs known as alveoli. Symptoms typically include some combination of productive or dry cough, chest pain, fever, and difficulty breathing. The severity of the condition is variable.

Covid-19 is usually caused by infection with viruses, Identifying the responsible pathogen can be difficult. Diagnosis is often based on symptoms and physical examination. Lung CT Scans, blood tests, and RT-PCR test may help confirm the diagnosis. The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated Covid-19.

Risk factors for Covid-19 include cystic fibrosis, chronic obstructive pulmonary disease (COPD), sickle cell disease, asthma, diabetes, heart failure, a history of smoking, a poor ability to cough (such as following a stroke), and a weak immune system. Vaccines to prevent certain types of Covid-19 (such as those caused by C.1.2 variant, linked to hyper infection), or linked to Delta variants are available. Other methods of prevention include hand washing to prevent infection, not smoking, and social distancing.

Treatment depends on the underlying cause. Covid-19 believed to be due to virus which can't be treated with antibiotics. If the Covid-19 is severe, the affected person is generally hospitalized. Oxygen therapy may be used if oxygen levels are low.

Due to Covid-19 nearly more than about 4,800,330 deaths globally (2.05 % of the world population) reported around the world. With the introduction of antibiotics and vaccines in the 20th century, survival has greatly improved. Nevertheless, Covid-19 remains a leading cause of death in developing countries, and also among the very old, the very young, and the chronically ill. Covid-19 often shortens the period of

suffering among those already close to death and has thus been called "the old man's friend".

People with infectious Covid-19 often have a productive cough, fever accompanied by shaking chills, shortness of breath, sharp or stabbing chest pain during deep breaths, and an increased rate of breathing. In elderly people, confusion may be the most prominent sign.

The typical signs and symptoms in children under five are fever, cough, and fast or difficult breathing. Fever is not very specific, as it occurs in many other common illnesses and may be absent in those with severe disease, malnutrition or in the elderly. In addition, a cough is frequently absent in children less than 2 months old. More severe signs and symptoms in children may include blue-tinged skin, unwillingness to drink, convulsions, ongoing vomiting, extremes of temperature, or a decreased level of consciousness.

Bacterial and viral cases of Covid-19 usually result in similar symptoms. Some causes are associated with classic, but non-specific, clinical characteristics. Covid-19 caused by Legionella may occur with abdominal pain, diarrhea, or confusion. Covid-19 caused by Streptococcus Covid-19 is associated with rusty colored sputum. Covid-19 caused by Klebsiella may have bloody sputum often described as "currant jelly". Bloody sputum (known as hemoptysis) may also occur with tuberculosis, Gram-negative Covid-19, lung abscesses and more commonly acute bronchitis. Covid-19 caused by Mycoplasma Covid-19 may occur in association with swelling of the lymph nodes in the neck, joint pain, or a middle ear infection. Viral Covid-19 presents more commonly with wheezing than bacterial Covid-19. Covid-19 was historically divided into "typical" and "atypical" based on the belief that the presentation predicted the underlying cause. However, evidence has not supported this distinction, therefore it is no longer emphasized.

Covid-19 is due to infections caused primarily by bacteria or viruses and less commonly by fungi and parasites. Although more than 100 strains of infectious agents have been identified, only a few are responsible for the majority of cases. Mixed

infections with both viruses and bacteria may occur in roughly 45% of infections in children and 15% of infections in adults. A causative agent may not be isolated in about half of cases despite careful testing. In an active population-based surveillance for community-acquired Covid-19 requiring hospitalization in five hospitals in Chicago and Nashville from January 2010 through June 2012, 2259 patients were identified who had radiographic evidence of Covid-19 and specimens that could be tested for the responsible pathogen. Most patients (62%) had no detectable pathogens in their sample, and unexpectedly, respiratory viruses were detected more frequently than bacteria. Specifically, 23% had one or more viruses, 11% had one or more bacteria, 3% had both bacterial and viral pathogens, and 1% had a fungal or mycobacterial infection. "The most common pathogens were human rhinovirus (in 9% of patients), influenza virus (in 6%), and Streptococcus Covid-19 (in 5%)."

The term Covid-19 is sometimes more broadly applied to any condition resulting in inflammation of the lungs (caused for example by autoimmune diseases, chemical burns or drug reactions); however, this inflammation is more accurately referred to as pneumonitis.

Factors that predispose to Covid-19 include smoking, immunodeficiency, alcoholism, chronic obstructive pulmonary disease, sickle cell disease (SCD), asthma, chronic kidney disease, liver disease, and biological aging. Additional risks in children include not being breastfed, exposure to cigarette smoke and other air pollution, malnutrition, and poverty. The use of acid-suppressing medications – such as proton-pump inhibitors or H2 blockers – is associated with an increased risk of Covid-19. Approximately 10% of people who require mechanical ventilation develop ventilator-associated Covid-19, and people with a gastric feeding tube have an increased risk of developing aspiration Covid-19. For people with certain variants of the FER gene, the risk of death is reduced in sepsis caused by Covid-19. However, for those with TLR6 variants, the risk of getting Legionnaires' disease is increased.

1.1 EXISTING METHOD

This model emphasizes an existing method that which is designed using the some of the algorithms of deep learning and machine learning algorithms. Here the process is performed using the LSTM, RNN, Ensemble algorithm and SVM where these algorithms are unable to perform accurately and couldn't get the proper accuracy.

Disadvantages:

- Less feature compatibility
- Low accuracy
- Less performance

1.2 PROPOSED METHOD

In our proposed method we are performing the classification of either the person is infected with the Covid-19 or not using Convolution Neural Network (CNN) of deep learning. As Covid-19 causes pleural effusion, a condition in which fluids fill the lung, causing respiratory difficulty. Early diagnosis of Covid-19 is crucial to ensure curative treatment and increase survival rates. Hence, proper classification is important for the proper treatment that which will be possible by using our proposed method. Where we are implementing this process using the X-Ray images dataset and then training with the CNN dataset. Block diagram of proposed method is shown below.

Advantages:

- Cheaper to operate.
- It can be scaled up quickly.
- Time minimising.

APPLICATIONS:

- Hospitals.
- Airports.
- Private Organizations.

Deep Learning techniques have been on a rise since the last few years and have completely changed the scenario of many research fields. Especially, in medical field, image data set such as retina image, chest X-ray, and brain MRI provides promising results with an extended accuracy% by using the deep learning techniques As we know, X-ray machines provides inexpensive and faster results for scanning of various human organs in the hospitals. The interpretation of various X-rays images is usually performed manually by an expert radiologists.

As a data scientist, if we train those captured images with the significance of deep learning that will be a great aid to medical experts for detecting the COVID-19 patients. This will help the developing countries where the X-ray facility is available but the availability of an expert is still a dream. To this advantage, we also aim to develop a deep neural network named 'nCOVnet' that can analyze the X-ray images of lungs and detect whether the person tests positive for the virus or not. Among various deep learning classifiers, in particular, the Convolutional Neural Networks (CNN) have been immensely effective in computer vision and medical image analysis tasks. The results of CNN have proven its cogency in mapping of image data to a precise and expected output. Since the lungs are the primary target of the virus, analysing their changes can give an explicit result of presence of the virus.

The main contribution is to propose a CNN based model, which is able to train the images of corona virus infected lungs and those of healthy lungs.

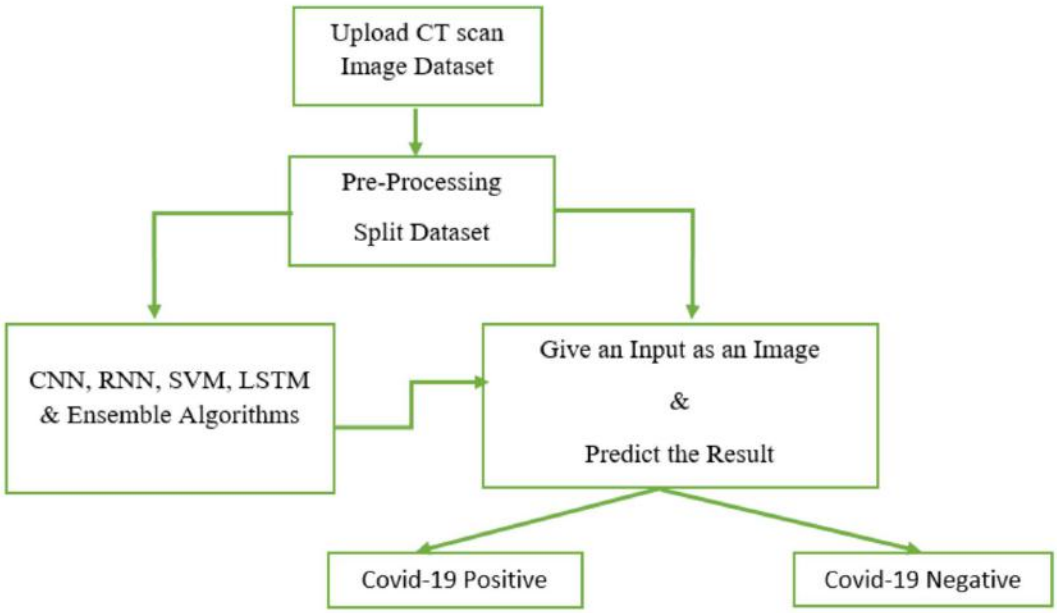

Fig.1.1:Block Diagram

CHAPTER – 2

LITERATURE SURVEY

[1] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, ''Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-19) in China: A report of 1014 cases,'' Radiology, vol. 296, no. 2, pp. E32–E40, Aug. 2020, doi: 10.1148/radiol. 2020200642.

Background Chest CT is used in the diagnosis of coronavirus disease 2019 (COVID-19) and is an important complement to reverse-transcription polymerase chain reaction (RT-PCR) tests. Purpose To investigate the diagnostic value and consistency of chest CT as compared with RT-PCR assay in COVID-19. Materials and Methods This study included 1014 patients in Wuhan, China, who underwent both chest CT and RT-PCR tests between January 6 and February 6, 2020. With use of RT-PCR as the reference standard, the performance of chest CT in the diagnosis of COVID-19 was assessed. In addition, for patients with multiple RT-PCR assays, the dynamic conversion of RT-PCR results (negative to positive, positive to negative) was analyzed as compared with serial chest CT scans for those with a time interval between RT-PCR tests of 4 days or more. Results Of the 1014 patients, 601 of 1014 (59%) had positive RT-PCR results and 888 of 1014 (88%) had positive chest CT scans. The sensitivity of chest CT in suggesting COVID-19 was 97% (95% confidence interval: 95%, 98%; 580 of 601 patients) based on positive RT-PCR results. In the 413 patients with negative RT-PCR results, 308 of 413 (75%) had positive chest CT findings. Of those 308 patients, 48% (103 of 308) were considered as highly likely cases and 33% (103 of 308) as probable cases. At analysis of serial RT-PCR assays and CT scans, the mean interval between the initial negative to positive RT-PCR results was 5.1 days ± 1.5; the mean interval between initial positive to subsequent negative RT-PCR results was 6.9 days ± 2.3. Of the 1014 patients, 60% (34 of 57) to 93% (14 of 15) had initial positive CT scans consistent with COVID-19 before (or parallel to) the initial positive RT-PCR results. Twenty-four of 57 patients (42%) showed improvement on follow-up chest CT scans before the RT-PCR results turned negative. Conclusion Chest CT has a high sensitivity for diagnosis of coronavirus disease 2019 (COVID-19). Chest CT may be considered as a primary tool for the current COVID-19 detection in epidemic areas.

[2] D. S. Kermany et al., ''Identifying medical diagnoses and treatable diseases by image-based deep learning,'' Cell, vol. 172, no. 5, pp. 1122–1131, Feb. 2018, doi: 10.1016/j.cell.2018.02.010.

The implementation of clinical-decision support algorithms for medical imaging faces challenges with reliability and interpretability. Here, we establish a diagnostic tool based on a deep-learning framework for the screening of patients with common treatable blinding retinal diseases. Our framework utilizes transfer learning, which trains a neural network with a fraction of the data of conventional approaches. Applying this approach to a dataset of optical coherence tomography images, we demonstrate performance comparable to that of human experts in classifying age-related macular degeneration and diabetic macular edema. We also provide a more transparent and interpretable diagnosis by highlighting the regions recognized by the neural network. We further demonstrate the general applicability of our AI system for diagnosis of pediatric pneumonia using chest X-ray images. This tool may ultimately aid in expediting the diagnosis and referral of these treatable conditions, thereby facilitating earlier treatment, resulting in improved clinical outcomes. VIDEO ABSTRACT.

[3] R. Kohavi, ''A study of cross-validation and bootstrap for accuracy estimation and model selection,'' in Proc. 14th Int. Joint Conf. Artif. Intell., vol. 2, Aug. 1995, pp. 1137–1143.

We review accuracy estimation methods and compare the two most common methods cross validation and bootstrap. Recent experimental results on artificial data and theoretical re cults in restricted settings have shown that for selecting a good classifier from a set of classifiers (model selection), ten-fold cross-validation may be better than the more expensive leaveone-out cross-validation. We report on a largescale experiment--over half a million runs of C4.5 and a Naive-Bayes algorithm--to estimate the effects of different parameters on these algorithms on real-world datasets. For cross validation we vary the number of folds and whether the folds are stratified or not, for bootstrap, we vary the number of bootstrap samples. Our results indicate that for real-word datasets similar to ours, The best method to use for model selection is ten fold stratified cross validation even if computation power allows using more folds.

[4] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, ''Deep feature extraction and classification of hyperspectral images based on convolutional neural networks,'' IEEE Trans. Geosci. Remote Sens., vol. 5

Due to the advantages of deep learning, in this paper, a regularized deep feature extraction (FE) method is presented for hyperspectral image (HSI) classification using a convolutional neural network (CNN). The proposed approach employs several convolutional and pooling layers to extract deep features from HSIs, which are nonlinear, discriminant, and invariant. These features are useful for image classification and target detection. Furthermore, in order to address the common issue of imbalance between high dimensionality and limited availability of training samples for the classification of HSI, a few strategies such as L2 regularization and dropout are investigated to avoid overfitting in class data modeling. More importantly, we propose a 3-D CNN-based FE model with combined regularization to extract effective spectral-spatial features of hyperspectral imagery. Finally, in order to further improve the performance, a virtual sample enhanced method is proposed. The proposed approaches are carried out on three widely used hyperspectral data sets: Indian Pines, University of Pavia, and Kennedy Space Center. The obtained results reveal that the proposed models with sparse constraints provide competitive results to state-of-the-art methods. In addition, the proposed deep FE opens a new window for further research.

[5] A. L. Maas, A. Y. Hannun, and A. Y. Ng, ''Rectifier nonlinearities improve neural network acoustic models,'' presented at the Workshop Deep Learn. Audio, Speech Lang. Process., Int. Conf. Mach. Learn. (ICML), Atlanta, GA, USA, vol. 30, Jun. 2013.

Deep neural network acoustic models produce substantial gains in large vocabulary continuous speech recognition systems. Emerging work with rectified linear (ReL) hidden units demonstrates additional gains in final system performance relative to more commonly used sigmoidal nonlinearities. In this work, we explore the use of deep rectifier networks as acoustic models for the 300 hour Switchboard conversational speech recognition task. Using simple training procedures without pretraining, networks with rectifier nonlinearities produce 2% absolute reductions in word error rates over their sigmoidal counterparts.

CHAPTER – 3

ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 SYSTEM REQUIREMENTS SPECIFICATION

3.2.1 Functional and non-functional requirements:

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements:

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional requirements:

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
2) The processing of each request should be done within 10 seconds
3) The site should load in 3 seconds whenever of simultaneous users are > 10000

3.2.2 HARDWARE & SOFTWARE REQUIREMENTS:

H/W Configuration:

- Processor : I3/Intel Processor
- Hard Disk : 160GB
- RAM : 8Gb

S/W Configuration:

- Operating System : Windows 7/8/10
- Server Side Script : Python,
- IDE : Pycharm
- Libraries Used : Sklearn , Pandas, Numpy, matplotlib, OpenCV, TensorFlow, Keras
- GUI : PyQT5
- Technology : Python 3.6+

3.3 SOFTWARE INSTALLATION:

Installing Python:

1. To download and install Python visit the official website of Python https://www.python.org/downloads/ and choose your version.

Fig 3.1:Python Download Window

2. Once the download is complete, run the exe for install Python. Now click on Install Now.

3. You can see Python installing at this point.

4. When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

Installing PyCharm:

1. To download PyCharm website https://www.jetbrains.com/pycharm/download/ and click the "DOWNLOAD" link under the Community Section.
2. Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.
3. On the next screen, Change the installation path if required. Click “Next”.
4. On the next screen, you can create a desktop shortcut if you want and click on “Next”.
5. Choose the start menu folder. Keep selected Jet Brains and click on “Install”.

6. Wait for the installation to finish.

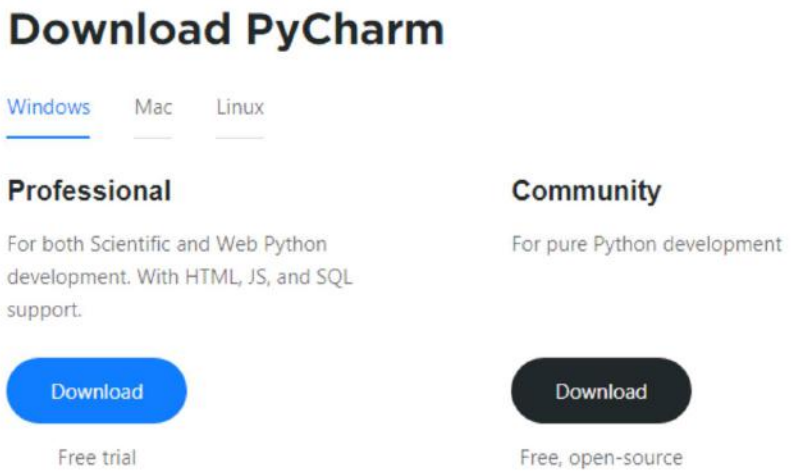

Fig 3.2:Downloading PyCharm

7. Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".
8. After you click on "Finish," the Following screen will appear.

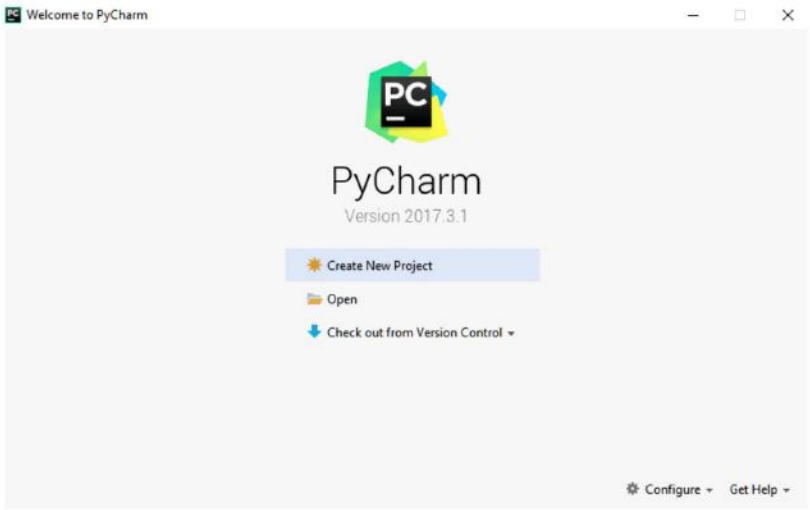

Fig 3.1:PyCharm Setup

9. You need to install some packages to execute your project in a proper way.

10. Open the command prompt/ anaconda prompt or terminal as administrator.

11. The prompt will get open, with specified path, type "pip install package name" which you want to install (like NumPy, pandas, sea born, scikit-learn, Matplotlib, Pyplot)

Ex: Pip install NumPy

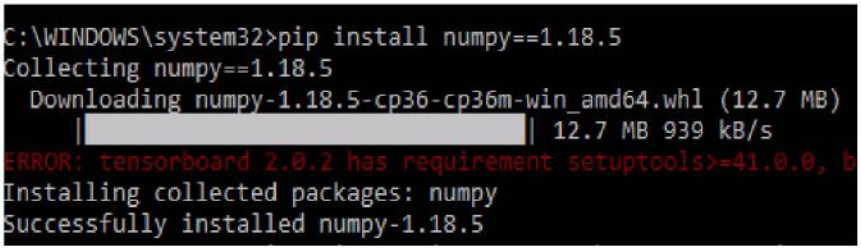

Fig 3.4:Installing Python Libraries

CHAPTER – 4

SOFTWARE DEVELOPMENT LIFE CYCLE – SDLC:

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.

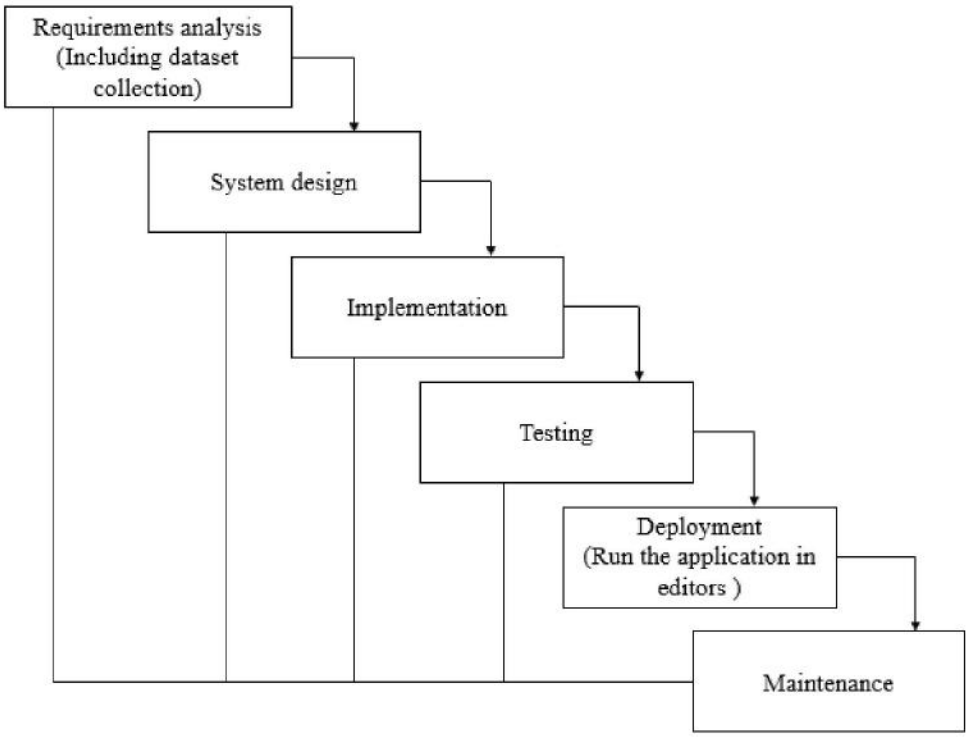

Fig 4:SDLC

- **Requirement Gathering and analysis** − all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − the requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − with inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

CHAPTER – 5

DESIGN

5.1 SYSTEM DESIGN:

Input Design:

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

- It should serve specific purpose effectively such as storing, recording, and retrieving the information.
- It ensures proper completion with accuracy.
- It should be easy to fill and straightforward.
- It should focus on user's attention, consistency, and simplicity.
- All these objectives are obtained using the knowledge of basic design principles regarding −
    - What are the inputs needed for the system?
    - How end users respond to different elements of forms and screens.

Objectives for Input Design:

The objectives of input design are

- To design data entry and input procedures
- To reduce input volume
- To design source documents for data capture or devise other data capture methods
- To design input data records, data entry screens, user interface screens, etc.
- To use validation checks and develop effective input controls.

Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Objectives of Output Design:

The objectives of input design are:

- To develop output design that serves the intended purpose and eliminates the production of unwanted output.
- To develop the output design that meets the end user's requirements.
- To deliver the appropriate quantity of output.
- To form the output in appropriate format and direct it to the right person.
- To make the output available on time for making good decisions.

5.2 UML INTRODUCTION

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

5.2.1 UML DIAGRAMS

USE CASE DIAGRAM

- A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
- Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
- The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

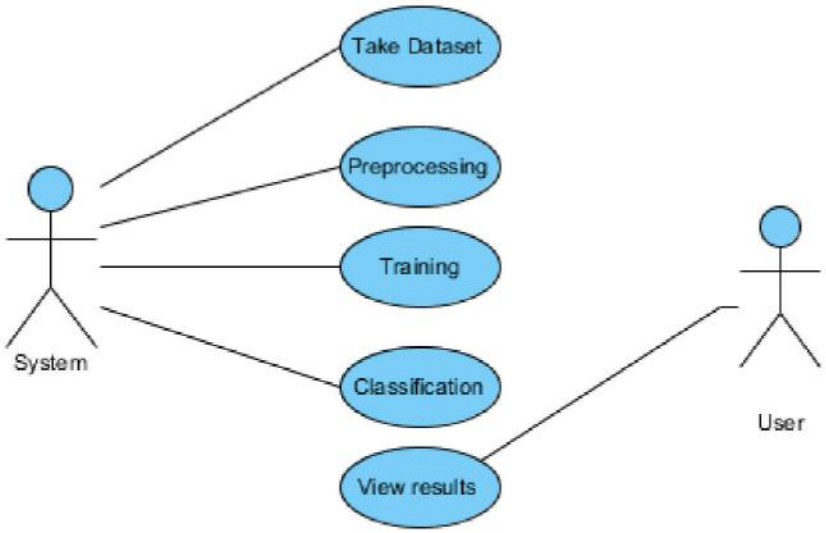

Fig 5.1:Use Case Diagram

CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information

Fig 5.2:Class Diagram

SEQUENCE DIAGRAM

- A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order.
- It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams

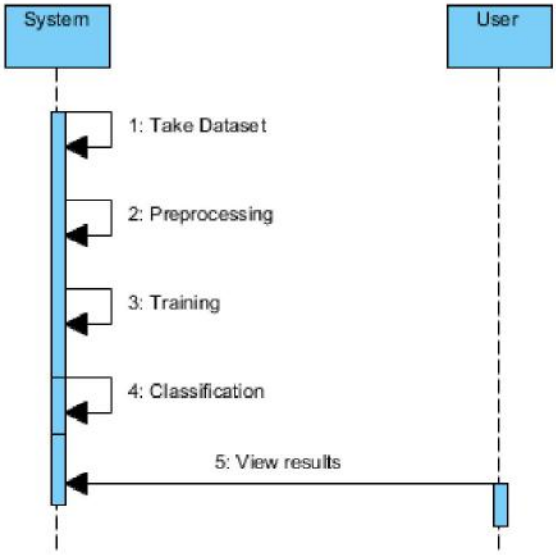

Fig 5.3:Sequence Diagram

COLLABORATION DIAGRAM:

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

Fig 5.4:Collaboration Diagram

DEPLOYMENT DIAGRAM

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware's used to deploy the application.

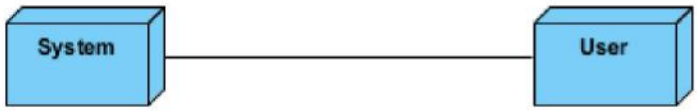

Fig 5.5:Deployment Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

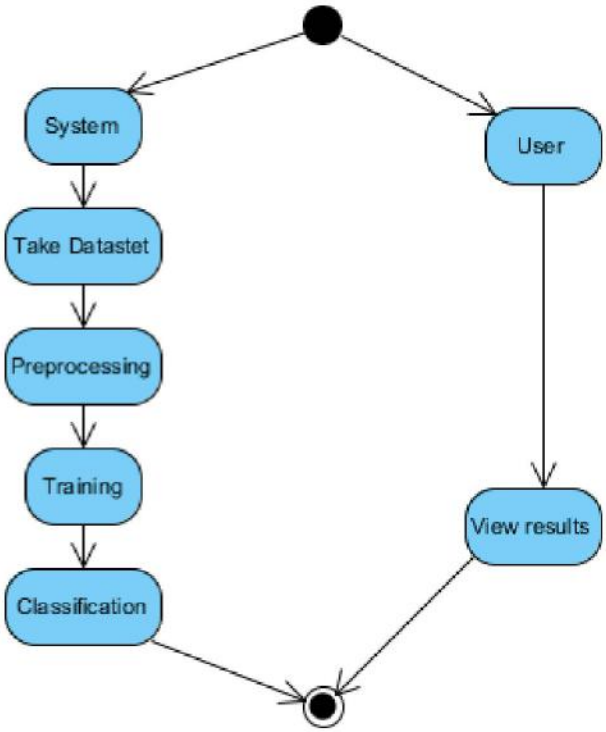

Fig 5.6:Activity Diagram

COMPONENT DIAGRAM:

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required function is covered by planned development.

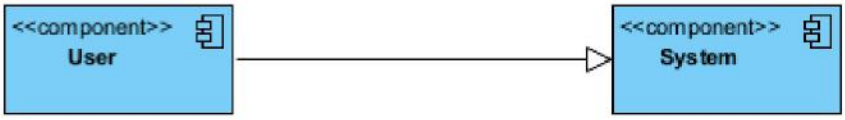

Fig 5.1:Component Diagram

ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.

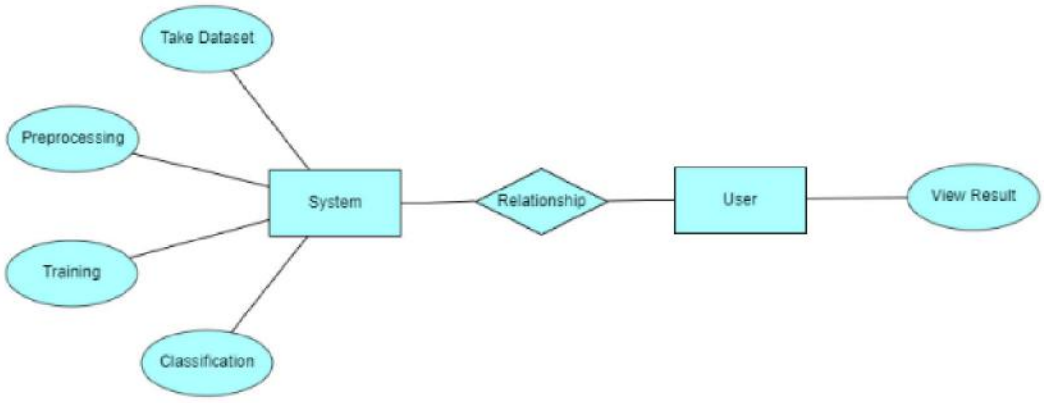

Fig 5.8:ER Diagram

DFD DIAGRAM:

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems

analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

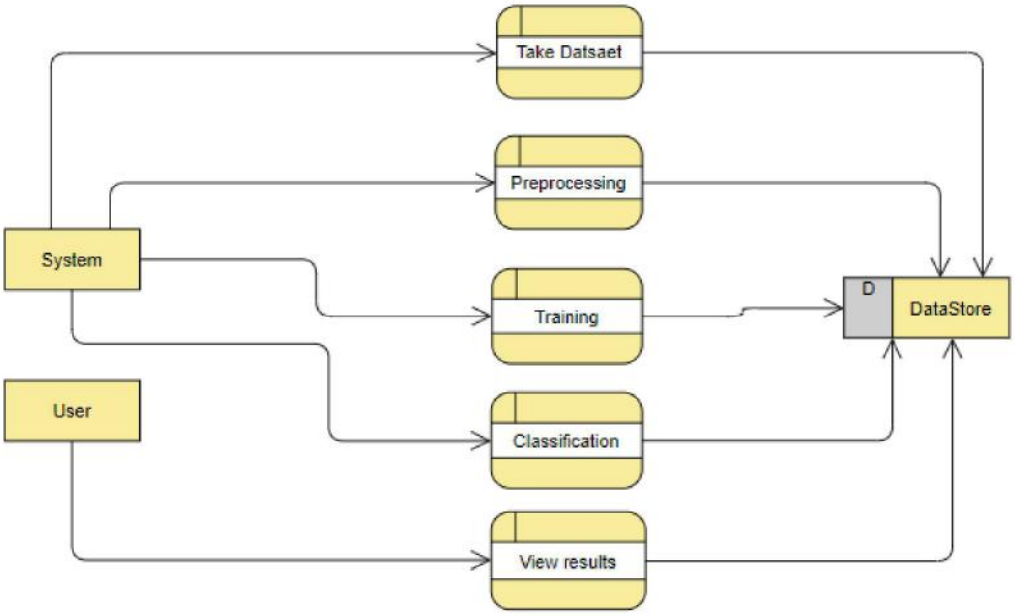

Fig 5.1:DFD Diagram

5.3 ARCHITECTURE:

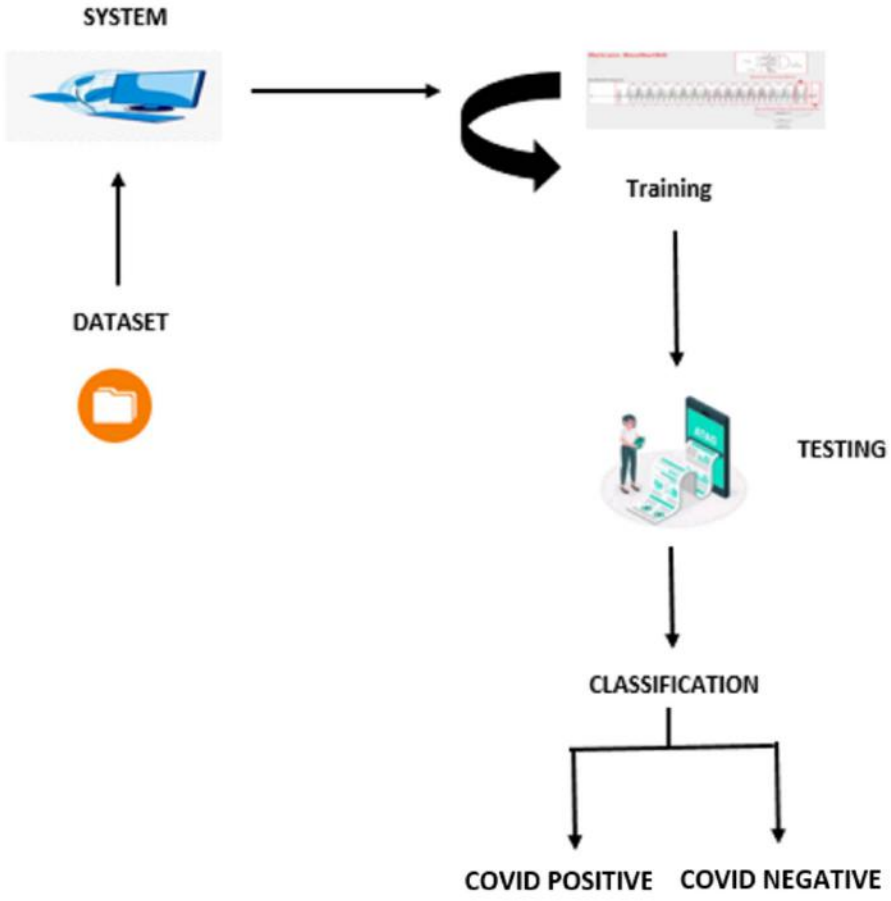

Fig 5.10:Architecture

CHAPTER – 6
IMPLEMENTATION

6.1 MODULES:

System:

1.1 Create Dataset:

The dataset containing images of the Lung X-Ray images with the Covid-19 affected and without Covid-19 i.e., normal are to be classified is split into training and testing dataset.

1.2 Pre-processing:

Resizing and reshaping the images into appropriate format to train our model.

1.3 Training:

Use the pre-processed training dataset is used to train our model using CNN.

1.4 Classification:

The results of our model is display of X –Ray images are either with Covid-19 or normal.

User:

2.1 Upload Image

The user has to upload an image which needs to be classified and this is implemented using PyQt5 GUI.

2.2 View Results

The classified image results are viewed by user.

6.2 PyQt5 Designer:

PyQt is a GUI widgets toolkit. It is a Python interface for **Qt**, one of the most powerful,

and popular cross-platform GUI library. PyQt was developed by River Bank

Computing Ltd. The latest version of PyQt can be downloaded from its official website – riverbankcomputing.com

PyQt API is a set of modules containing a large number of classes and functions. While **QtCore** module contains non-GUI functionality for working with file and

directory etc., **QtGui** module contains all the graphical controls. In addition, there are modules for working with XML **(QtXml)**, SVG **(QtSvg)**, and SQL **(QtSql)**, etc.

A list of frequently used modules is given below –

- **tCore** − Core non-GUI classes used by other modules
- **QtGui** − Graphical user interface components
- **QtMultimedia** − Classes for low-level multimedia programming
- **QtNetwork** − Classes for network programming
- **QtOpenGL** − OpenGL support classes
- **QtScript** − Classes for evaluating Qt Scripts
- **QtSql** − Classes for database integration using SQL
- **QtSvg** − Classes for displaying the contents of SVG files
- **QtWebKit** − Classes for rendering and editing HTML
- **QtXml** − Classes for handling XML

Supporting Environments:

PyQt is compatible with all the popular operating systems including Windows, Linux, and Mac OS. It is dual licensed, available under GPL as well as commercial license.The latest stable version is **PyQt5-5.13.2.**

Windows:

Wheels for 32-bit or 64-bit architecture are provided that are compatible with Python

version 3.5 or later. The recommended way to install is using **PIP** utility −

pip3 install PyQt5

To install development tools such as Qt Designer to support PyQt5 wheels, following is the command –

pip3 install pyqt5-tools

Here is the list of Widgets which we will discuss one by one in this chapter.

Sr.No	Widgets & Description
1	QLabel A QLabel object acts as a placeholder to display non-editable text or image, or a movie of animated GIF. It can also be used as a mnemonic key for other widgets.
2	QLineEdit QLineEdit object is the most commonly used input field. It provides a box in which one line of text can be entered. In order to enter multi-line text, QTextEdit object is required.
3	QPushButton In PyQt API, the QPushButton class object presents a button which when clicked can be programmed to invoke a certain function.
4	QRadioButton A QRadioButton class object presents a selectable button with a text label. The user can select one of many options presented on the form. This class is derived from QAbstractButton class.
5	QCheckBox A rectangular box before the text label appears when a QCheckBox object is added to the parent window. Just as QRadioButton, it is also a selectable button.
6	QComboBox A QComboBox object presents a dropdown list of items to select from. It takes minimum screen space on the form required to display only the currently selected item.
7	QSpinBox

	A QSpinBox object presents the user with a textbox which displays an integer with up/down button on its right.
8	QSlider Widget & Signal QSlider class object presents the user with a groove over which a handle can be moved. It is a classic widget to control a bounded value.
9	QMenuBar, QMenu & QAction A horizontal QMenuBar just below the title bar of a QMainWindow object is reserved for displaying QMenu objects.
10	QToolBar A QToolBar widget is a movable panel consisting of text buttons, buttons with icons or other widgets.

Table 1:Widgets and Description

Building a PyQt5 - Using Qt Designer

The PyQt installer comes with a GUI builder tool called **Qt Designer**. Using its simple drag and drop interface, a GUI interface can be quickly built without having to write the code. It is however, not an IDE such as Visual Studio. Hence, Qt Designer does not have the facility to debug and build the application.Start Qt Designer application which is a part of development tools and installed in scripts folder of the virtual environment.

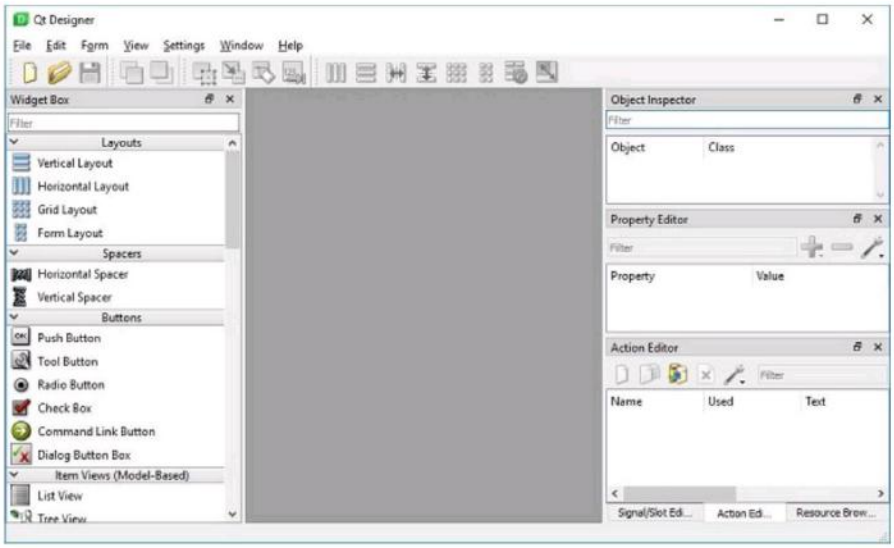

Fig 6.1:PyQt5 Designer

Start designing GUI interface by choosing File → New menu.

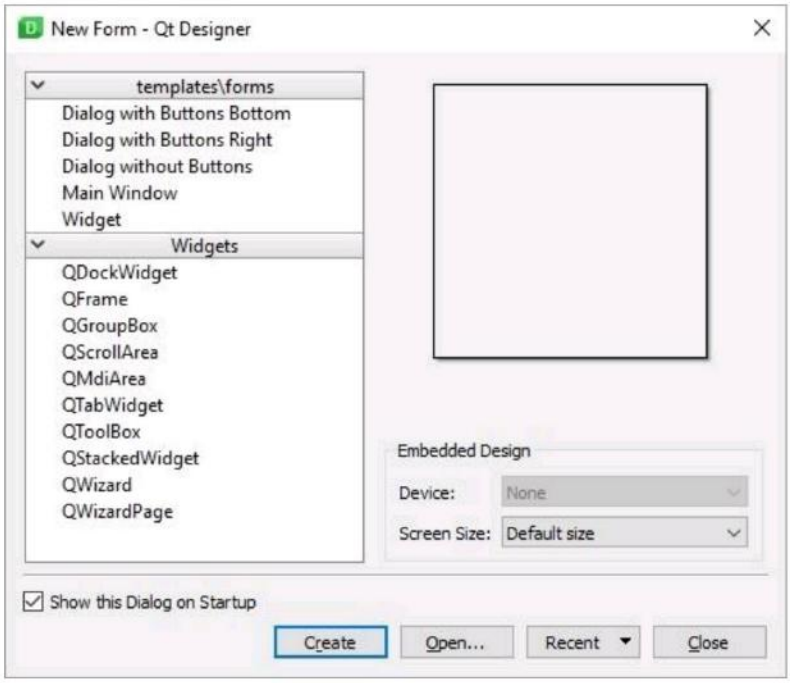

Fig 6.2:Creating Widgets

You can then drag and drop required widgets from the widget box on the left pane.

You can also assign value to properties of widget laid on the form.

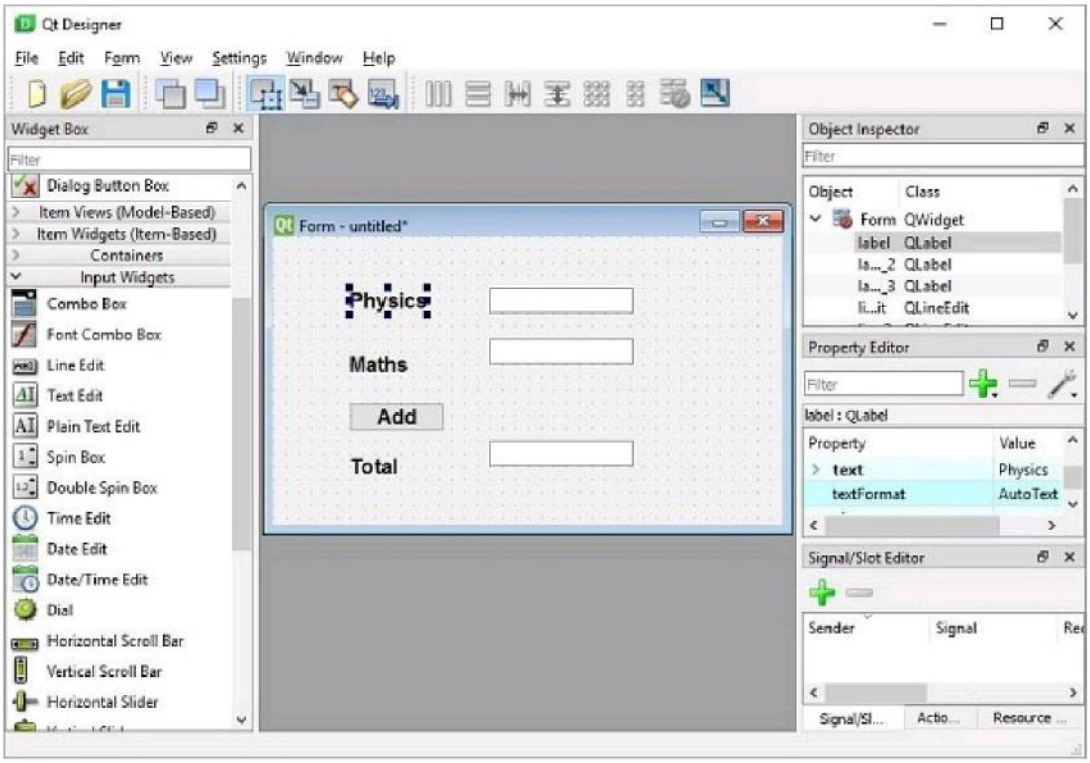

Fig 6.3:Widgets

Fig 6.4:GUI

6.3 METHODOLOGY :

Convolutional Neural Network:

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

Step1: convolutional operation

The first building block in our plan of attack is convolution operation. In this step, we will touch on feature detectors, which basically serve as the neural network's filters. We will also discuss feature maps, learning the parameters of such maps, how patterns are detected, the layers of detection, and how the findings are mapped out.

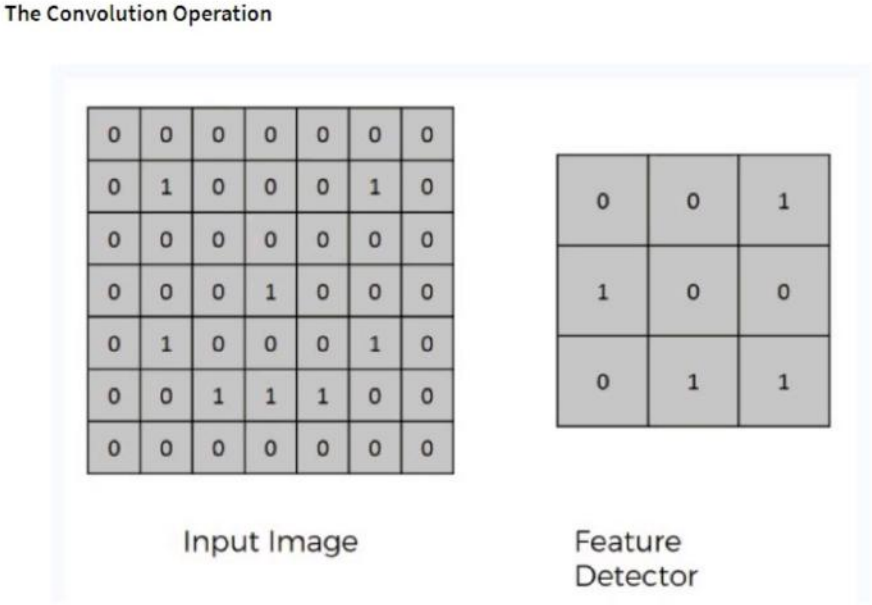

Fig 6.5: Detector

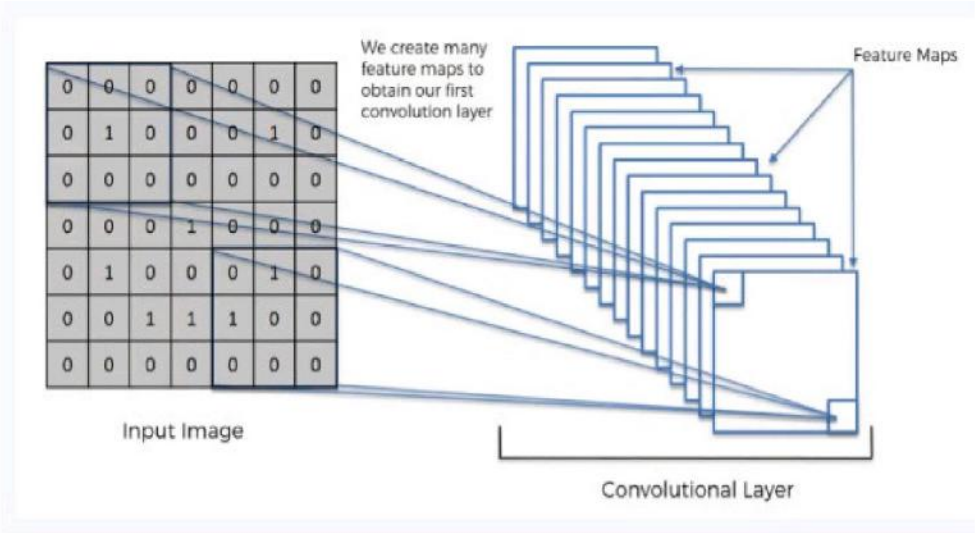

Fig 6.6:Feature maps

Step (1b): ReLU Layer

The second part of this step will involve the Rectified Linear Unit or Relook. We will cover Relook layers and explore how linearity functions in the context of Convolutional Neural Networks.

Not necessary for understanding CNN's, but there's no harm in a quick lesson to improve your skills.

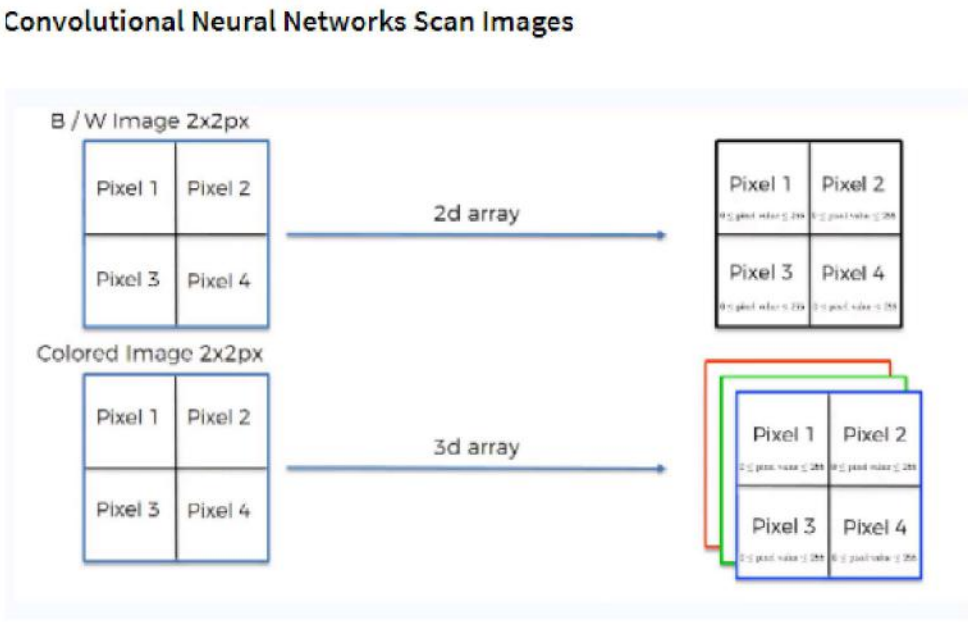

Fig 5.1:Scanning Images

Step 2: Pooling Layer

In this part, we'll cover pooling and will get to understand exactly how it generally works. Our nexus here, however, will be a specific type of pooling; max pooling. We'll cover various approaches, though, including mean (or sum) pooling. This part will end with a demonstration made using a visual interactive tool that will definitely sort the whole concept out for you.

Step 3: Flattening

This will be a brief breakdown of the flattening process and how we move from pooled to flattened layers when working with Convolutional Neural Networks.

Step 4: Full Connection

In this part, everything that we covered throughout the section will be merged together. By learning this, you'll get to envision a fuller picture of how Convolutional Neural Networks operate and how the "neurons" that are finally produced learn the classification of images.

Summary

In the end, we'll wrap everything up and give a quick recap of the concept covered in the section. If you feel like it will do you any benefit (and it probably will), you should check out the extra tutorial in which Softmax and Cross-Entropy are covered. It's not mandatory for the course, but you will likely come across these concepts when working with Convolutional Neural Networks and it will do you a lot of good to be familiar with them.

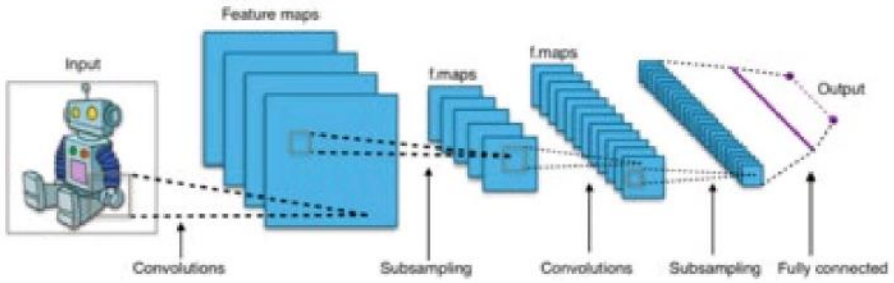

Fig 6.8:Steps in neural network

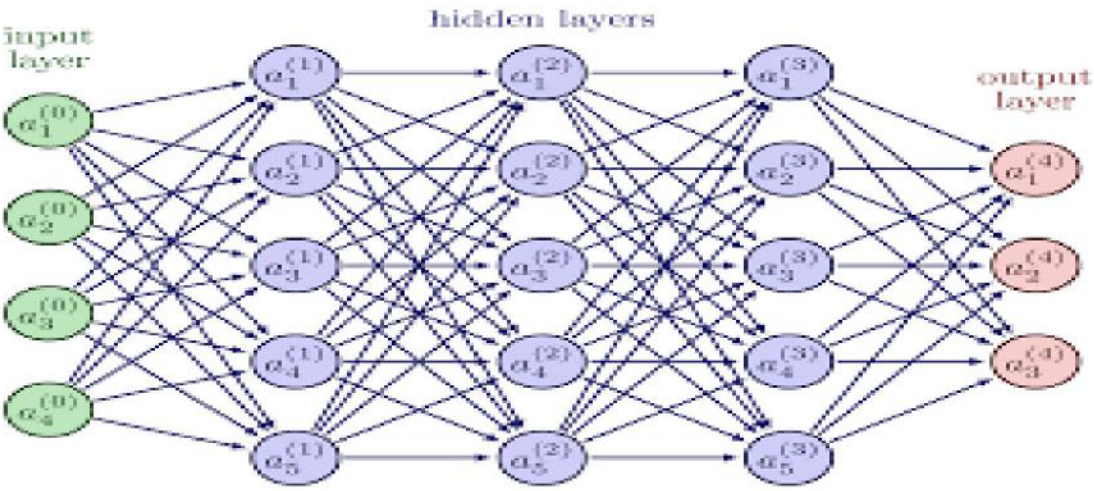

Fig 6.9:Neural Network

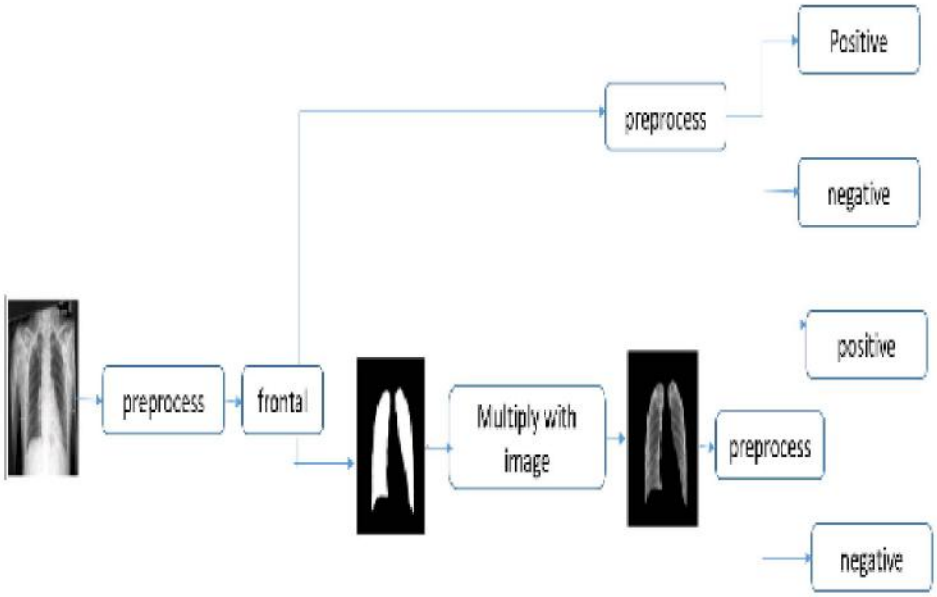

Fig 6.10:Flow Diagram

CHAPTER – 7

RESULT & OUTPUT

GUI Home page:

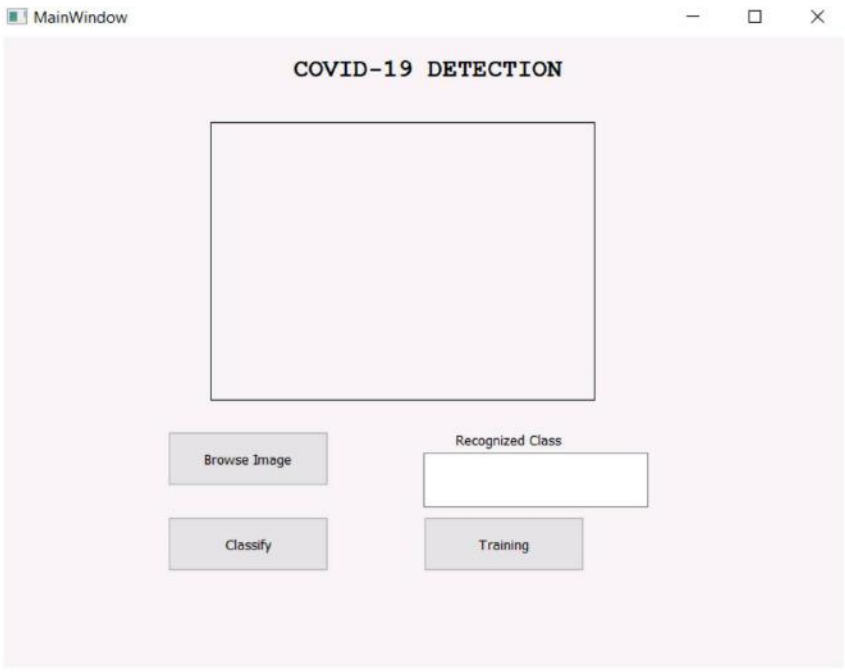

Fig 7.1:Home Page

Uploaded Input Image:

Fig 7.2:GUI Interface

Classified Output:

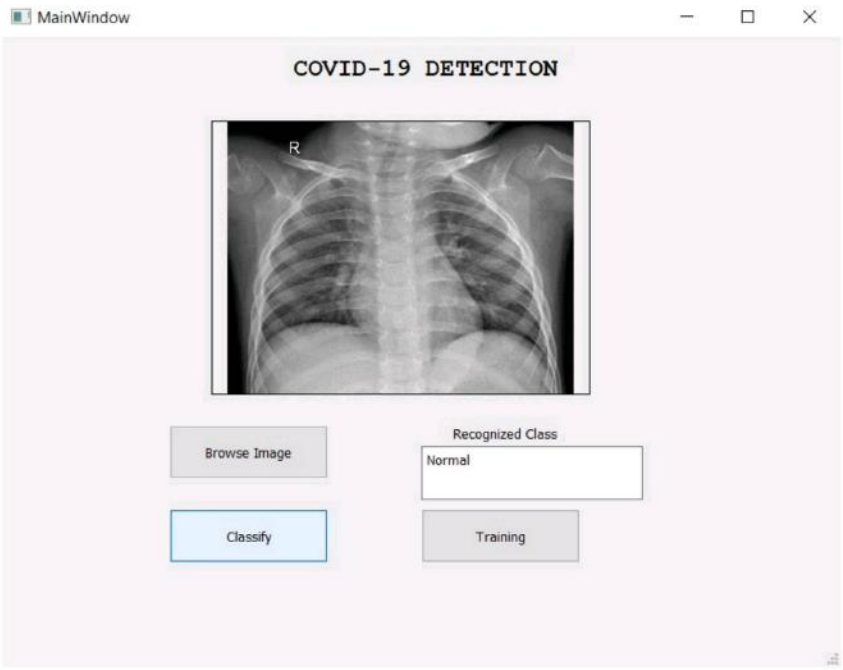

Fig 7.3:GUI Interface with Normal

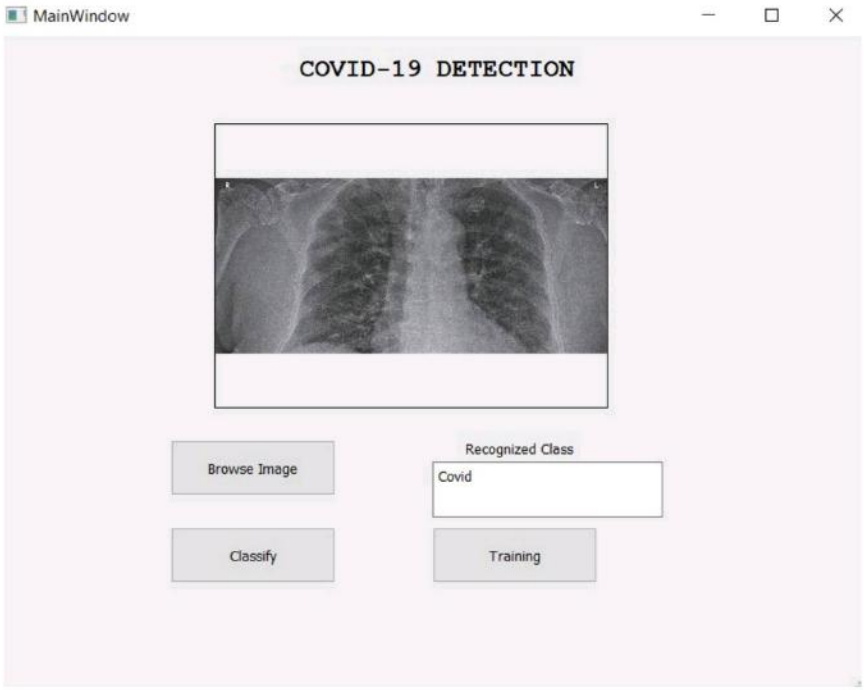

Fig 7.4:GUI Interface with Covid Positive

TEST CASES:

Input	Output	Result
Input image	Tested for the classification of person affected with Covid-19 or not from the X-Rayimages	Success

CHAPTER-8
TESTING

5.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the.

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM_TEST

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or

requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results **:** All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results : All the test cases mentioned above passed successfully. No defects encountered.

CONCLUSION

In this project we have successfully classified the images of lung X-ray images images of a person, is either affected with the Covid-19 or normal using the deep learning algorithm. Here, we have considered the dataset of Lung X-Ray images which will be of 2 different types (Covid-19 affected and normal) and trained using CNN. After the training we have tested by uploading the image and classified it using PyQt5 GUI.

FUTURE SCOPE:

In future this application can be extended to a real time model, where CT scan images of people are taken and the results are generated immediately which can be very useful in airports when people travels to different countries.

REFERENCES

[1] S. Karanam, R. Li, F. Yang, W. Hu, T. Chen, and Z. Wu, ''Towards contactless patient positioning,'' IEEE Trans. Med. Imag., vol. 39, no. 8, pp. 2701–2710, Aug. 2020.

[2] World Health Organization (WHO). WHO Coronavirus Disease (COVID-19) Dashboard. Accessed: Jul. 15, 2020. [Online]. Available: https://covid19.who.int/

[3] W. Wang, Y. Xu, R. Gao, R. Lu, K. Han, G. Wu, and W. Tan, ''Detection of SARS-CoV-2 in different types of clinical specimens,'' J. Amer. Med. Assoc., vol. 323, no. 18, pp. 1843–1844, May 2020, doi: 10.1001/ jama.2020.3786.

[4] G. D. Rubin et al., ''The role of chest imaging in patient management during the COVID-19 pandemic: A multinational consensus statement from the Fleischner society,'' Radiology, vol. 296, no. 1, pp. 172–180, Jul. 2020, doi: 10.1148/radiol.2020201365.

[5] M. Hosseiny, S. Kooraki, A. Gholamrezanezhad, S. Reddy, and L. Myers, ''Radiology perspective of coronavirus disease 2019 (COVID-19): Lessons from severe acute respiratory syndrome and middle east respiratory syndrome,'' Amer. J. Roentgenol., vol. 214, no. 5, pp. 1078–1082, May 2020, doi: 10.2214/AJR.20.22969.

[6] B. Udugama, P. Kadhiresan, H. N. Kozlowski, A. Malekjahani, M. Osborne, V. Y. C. Li, H. Chen, S. Mubareka, J. B. Gubbay, and W. C. W. Chan, ''Diagnosing COVID-19: The disease and tools for detection,'' ACS Nano, vol. 14, no. 4, pp. 3822–3835, Mar. 2020, doi: 10. 1021/acsnano.0c02624.

[7] American College of Radiology (ACR). ACR Recommendations for the Use of Chest Radiography and Computed Tomography (CT) for Suspected COVID-19 Infection. Accessed: Jul. 15, 2020. [Online]. Available:

[8] T. Ai, Z. Yang, H. Hou, C. Zhan, C. Chen, W. Lv, Q. Tao, Z. Sun, and L. Xia, ''Correlation of chest CT and RT-PCR testing for coronavirus disease 2019 (COVID-

19) in China: A report of 1014 cases,'' Radiology, vol. 296, no. 2, pp. E32–E40, Aug. 2020, doi: 10.1148/radiol. 2020200642.

[9] A. Dangis, C. Gieraerts, Y. D. Bruecker, L. Janssen, H. Valgaeren, D. Obbels, M. Gillis, M. V. Ranst, J. Frans, A. Demeyere, and R. Symons, ''Accuracy and reproducibility of low-dose submillisievert chest CT for the diagnosis of COVID-19,'' Radiol., Cardiothoracic Imag., vol. 2, no. 2, Apr. 2020, Art. no. e200196, doi: 10.1148/ryct. 2020200196.

[10] A. Jacobi, M. Chung, A. Bernheim, and C. Eber, ''Portable chest X-ray in coronavirus disease-19 (COVID-19): A pictorial review,'' Clin. Imag., vol. 64, pp. 35–42, Aug. 2020, doi: 10.1016/j.clinimag.2020.04.001.

[11] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, ''A survey of the recent architectures of deep convolutional neural networks,'' Artif. Intell. Rev., early access, Apr. 21, 2020, doi: 10.1007/s10462-020-09825-6.

[12] M. Mossa-Basha, C. C. Meltzer, D. C. Kim, M. J. Tuite, K. P. Kolli, and B. S. Tan, ''Radiology department preparedness for COVID-19: Radiology scientific expert review panel,'' Radiology, vol. 296, no. 2, pp. E106–E112, Aug. 2020, doi: 10.1148/radiol.2020200988.

[13] Imaging Technology News. How Does COVID-19 Appear in the Lungs? Accessed: Jul. 15, 2020. [Online]. Available: https://www. itnonline.com/content/how-does-COVID-19-appear-lungs

[14] R. Vaishya, M. Javaid, I. H. Khan, and A. Haleem, ''Artificial intelligence (AI) applications for COVID-19 pandemic,'' Diabetes Metabolic Syndrome, Clin. Res. Rev., vol. 14, no. 4, pp. 337–339, Jul. 2020, doi: 10.1016/ j.dsx.2020.04.012.

[15] J. Zhang, Y. Xie, Y. Li, C. Shen, and Y. Xia, ''COVID-19 screening on chest X-ray images using deep learning based anomaly detection,'' 2020, arXiv:2003.12338. [Online]. Available: http://arxiv.org/abs/2003.12338

[16] A. Narin, C. Kaya, and Z. Pamuk, ''Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks,'' 2020, arXiv:2003.10849. [Online]. Available: http://arxiv.org/abs/2003.10849

[17] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, ''Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network,'' 2020, arXiv:2003.13815. [Online]. Available: https://arxiv.org/abs/2003.13815

[18] L. Wang and A. Wong, ''COVID-net: A tailored deep convolutional neural network design for detection of COVID-19 cases from chest X-ray images,'' 2020, arXiv:2003.09871. [Online]. Available: http://arxiv.org/abs/2003.09871

[19] T. Ozturk, M. Talo, E. A. Yildirim, U. B. Baloglu, O. Yildirim, and U. R. Acharya, ''Automated detection of COVID-19 cases using deep neural networks with X-ray images,'' Comput. Biol. Med., vol. 121, Jun. 2020, Art. no. 103792, doi: 10.1016/j.compbiomed.2020.103792.

[20] X. Xu, X. Jiang, C. Ma, P. Du, X. Li, S. Lv, L. Yu, Y. Chen, J. Su, G. Lang, Y. Li, H. Zhao, K. Xu, L. Ruan, and W. Wu, ''Deep learning system to screen coronavirus disease 2019 pneumonia,'' 2020, arXiv:2002.09334. [Online]. Available: http://arxiv.org/abs/2002.09334 [21] O. Gozes, M. Frid-Adar, H. Greenspan, P. D. Browning, H. Zhang, W. Ji, A. Bernheim, and E. Siegel, ''Rapid AI development cycle for the coronavirus (COVID-19) pandemic: Initial results for automated detection & patient monitoring using deep learning CT image analysis,'' 2020, arXiv:2003.05037. [Online]. Available: http://arxiv.org/abs/2003.05037

[22] A. A. Ardakani, A. R. Kanafi, U. R. Acharya, N. Khadem, and A. Mohammadi, ''Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks,'' Comput. Biol. Med., vol. 121, Jun. 2020, Art. no. 103795, doi: 10.1016/j.compbiomed.2020.103795.

[23] T. Iqbal and H. Ali, ''Generative adversarial network for medical images (MI-GAN),'' J. Med. Syst., vol. 42, no. 11, p. 231, Oct. 2018, doi: 10.1007/ s10916-018-1072-9.

[24] X. Yi, E. Walia, and P. Babyn, ‘‘Generative adversarial network in medical imaging: A review,’’ Med. Image Anal., vol. 58, Dec. 2019, Art. no. 101552, doi: 10.1016/j.media.2019.101552.

[25] M. Chung, A. Bernheim, X. Mei, N. Zhang, M. Huang, X. Zeng, J. Cui, W. Xu, Y. Yang, Z. A. Fayad, A. Jacobi, K. Li, S. Li, and H. Shan, ‘‘CT imaging features of 2019 novel coronavirus (2019-nCoV),’’ Radiology, vol. 295, no. 1, pp. 202–207, Apr. 2020, doi: 10.1148/radiol.2020200230.