Deep learning assisted framework for chest infection detection

By: Mian Muhammad Umer Hassan Khan Qureshi

Supervised By:

Dr.Eva Navarro Lopez

A dissertation submitted in partial fulfillment of the requirement for the degree of

M.Sc. Data Science

(School of Engineering & Informatics)

The University of Wolverhampton

June 2023

ACKNOWLEDGEMENT

First of all, thanks to my supervisor, **Dr.Eva Navarro Lopez** for her patience, commitment and time for supporting me in this dissertation, It was a wonderful experience completing this dissertation under her supervision.I would also like to thanks my family who supported me in my work and giving me input to complete this dissertation.

ABSTRACT

Chest infection is a disease caused by virus or bacteria. Pneumonia and now-a-days COVID are the most common infections of lungs in human. Chest infection grows in a normal way in all humans through bacteria. COVID and Pneumonia are widely recognized forms of all other chest infections. Chest infection is life threatening and becoming one of the major causes of death in recent years. Correct recognition can affect the planning of treatment and increase the survival rate.

This project is about Convolutional Neural Networks (CNN) based model for the diagnosis of Corona virus, viral pneumonia and non-infected diseases. These infections influence and harm the human lungs. If a patient is infected with the virus, early diagnosis can help save the patient's life and stop the virus from spreading further. Because it is one of the most rapid and cost-effective methods for diagnosing the disease, the CNN model is utilized to aid in the early diagnosis of the virus using chest X-ray images. We proposed convolutional brain organizations (CNN) models, which were prepared utilizing datasets. With datasets that only included pneumonia cases, COVID-19 cases, and normal chest X-ray images, the model was trained for binary classification. An intelligent system to classify Chest infection by using state of the art image processing techniques. I made images fine and free from noise by applying pre-processing operations. In future work, remaining steps will be done. Two most common type of chest infection (COVID and Pneumonia) are included in the study. I intend making a system that enhance and optimize the chest infections diagnostic process. This will help radiologist for decision making and make diagnoses easily and more accurately.

Table of Content

List of Tables

LIST OF ABBREVIATIONS

IDE	Integrated Development Environment
CV	Computer Vision
HCI	Human Computer Interaction
GUI	Graphical User Interface
SRS	Software Requirements Specification
SDD	Software Design Description
DFD	Data Flow Diagram
CNN	Convolutional Neural Network
GoogleNet	Google Network

Chapter 1

Introduction

1.1 Introduction

COVID-19 is a respiratory infection that affects the human lungs and has recently been declared a global pandemic. As there were 532 million total cases, 507 million recoveries, and 6.3 million deaths reported to the World Health Organization [1] In December 2019, the first case of COVID-19 was detected in Wuhan, Hubei Province, China [2], after which it spread to other nations across the world. Because the COVID-19 virus is transmittable, early discovery is critical for both patients and those in their immediate vicinity, as the patient will receive proper care and others will be protected. The greatest strategy to fight the COVID-19 pandemic is to diagnose patients who have been infected with the virus as soon as possible. to provide special treatment and care COVID-19 is commonly diagnosed by reverse transcription-polymerase chain reaction (RT-PCR), although this test has low sensitivity in the early stages of the virus, which could lead to continued transmission. Because this test kit is expensive and in scarce, chest X-ray (CXR) images and computer tomography (CT) scans are the best options for diagnosing any patient who shows pneumonia symptoms.

viral pneumonia is also one of the top causes of death in both young and old people. According to the Centers for Disease Control and Prevention (CDC), over 1 million adult pneumonia patients are hospitalized and about 50,000 patients die from this disease in USA every year. Chest X-rays, according to the WHO [5]., are the best available method of diagnosing pneumonia. Pneumonia is a lung infection caused by bacteria, viruses, or fungi. It can be caused by bacteria, viruses, or fungi. Diagnosing pneumonia is considered a tedious task, even by expert radiologists, because its symptoms appear to be like other pathologies that affect the lungs.

In recent years, chest infection has become the leading cause of death and has become life-taking. There are many different forms of chest infections, each with its own structure, location in the chest, and nature. Computer-based methods can help radiologists and physicians improve their diagnostic abilities and accuracy. It can also reduce the time it takes to get correct Chest Infection classification findings. Correct COVID and Pneumonia treatment can influence treatment planning and improve survival rates. Radiologists still perform infection type analysis manually, which takes time and effort. As a result, both physicians and patients require the development of an automated system. The goal of this project is to provide an effective system for infection classification.

Radiologists and other health-care professionals can use this technology. There are numerous COVID and Pneumonia images to evaluate, as well as multiple angles of X-RAY scans to examine and diagnose. The goal of our research is to use many techniques including medical image processing and computer vision to properly detect and categories infections in the chest. Pre-processing of X-RAY scans taken from online infected imaging archives as well as scans obtained from numerous pathology labs are used in these procedures. We resize the images before applying the proposed classification techniques. The method is projected to improve the present COVID and Pneumonia screening procedures while also potentially lowering health-care expenditures by reducing the need for follow-up operations. For accurate characterization and analysis of biomedical imaging data, several processing steps are required. Modalities are the different sorts of X-RAY images that differ based on the method employed to collect them. In one modality, certain infection formations are more visible than in others. Furthermore, because X-RAY images do not emit radiation, they are safe.

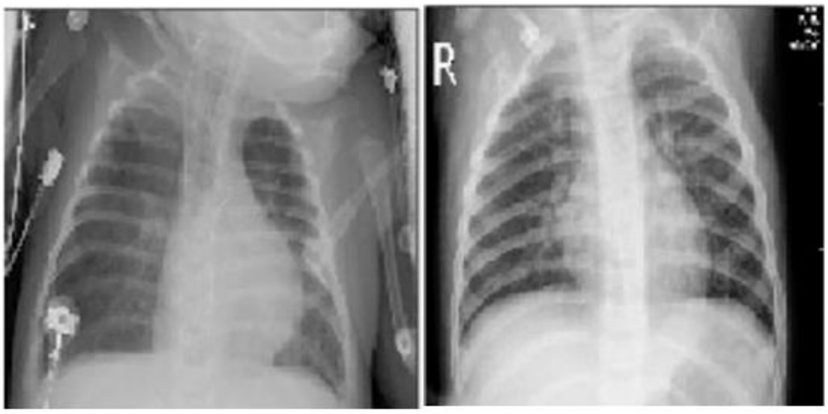

Figure 1. 1 Chest Infection

1.2 Brief

Chest Infection is caused by abnormalities in lungs. We have got multiple types of Chest Infections that exists, and each infection type has different size and severity. So, it is very difficult task to classify the infection part. COVID and Pneumonia treatments depend on doctor's decision. These decisions are done by seeing and analyzing the infection location in chest, infection shape and infection size. Doctors interpret the X-RAY of patient and decide that whether the patient has an infection or not. Human errors can mislead to wrong diagnosing of infection type and can be misleading for planning the treatment for patient.

Diagnosis of infection with the help of X-RAY image is mandatory but time-taking. This task is difficult for experts when they go for manual method. So, computer supported system are introduced for accurate is ease detection from X-RAY images. It is problematic for radiologists to detect the infection types manually and make decision due to variation in size, shape, and types of abnormal tissues. This system reduces the workload and avoids the mistakes that can be done by the radiologist.

Therefore, these systems are playing vital role in diagnosing and detection of infection automatically. Nowadays, the use of these systems supports the early detection of any type of infection in recent years. By studying the literature, we have analyzed that system's performance can be better by improving the computational speed and flexible systems for different chest data sets. The proposed system works on infection and non-infection types: pneumonia, COVID and no infection X-RAY images.

1.3 Relevance to Course Modules

In machine learning we have studied about different algorithms which are used to train models to perfume task automatically after getting the training data.

Report Writing Skill course is about learning how to write reports and other formal documentation, and, in our project, we need to write our documentation, so this course is helping a lot in this task.

Human Computer Interaction (HCI) helped us in understanding the ways in which a good design can be created.

1.4 Project Background

There are three categories of infections that we are dealing with i.e., COVID, Pneumonia and Normal images are also part of dataset. For examining the infection, the chest X-RAY are taken for detection of the infection and for diagnosing. Radiologist must examine these X-RAY image and they take a lot of time and energy for result. If a patient has advanced stage infection, then it is difficult to save the life because the manual method of detecting the exact infection whether it is COVID, or Pneumonia takes time. To get rid of this problem, an automated system is introduced which help radiologist in early detection of COVID and Pneumonia and to save the precious lives of patients.

1.5 Scope

The motivation we got from our environment. This is the era where people like to save time and perform their tasks in an effective manner. Meanwhile doctors are offering their services, but it takes a lot of time to diagnose a disease. If doctors want to get sure of whether a person has a disease or not? The doctor asks the person to perform certain tests to be sure of his/her ailment. Our application is for the doctors to facilitate them in the diagnoses. It would be helpful for the doctors to know the exact ailment of the patient in just few seconds. Our application has a wide scope in the field of medical like clinical laboratories, Hospitals and Clinics. This system will minimize doctor’s effort in diagnosis and the fatigue. They may face the issue of reduction in natural light and many other problems which may lead to wrong diagnosis.

1.6 Goals

Our goal is to develop well organized automatic Deep learning assisted framework for chest infection detection which has accuracy, low complexity and performance. The diagnosis elements are given as chest infection or no- chest infection images. The main objective of this project will be to build a system which will be helpful for the radiologists and physicians to classify the chest infection accurately and efficiently in X-ray image. We also intending to accomplish following sub-objectives:

- Use in Real time applications.
- Fast Computational time.
- Less prone to human error.
- Robust and flexible to large data amount.
- Compare Accuracy Results with other techniques.
- Cost effective.

Chapter 2

Literature Review

2.1 Literature Review

This heading contains the complete description of the past work and reports that have been completed in such manner.

2.2 Machine Learning Based Approaches

Machine Learning plays an important role in Computer Diagnosis systems and give promising results with improved accuracy. The machine learning also ensure the analysis of the different medical data and machine learning also provide the ability to make classy and good algorithms that help in diagnosing. For examining the bio-medical data, machine learning provides a worthy approach for making efficient algorithms. Following are the approaches that have been proposed by researchers.

Dipali M. Joshi [1] A neural network technique to classification is proposed. Matrix (GLCM). After then, the GLCM-based features were compared to previously stored features. After that, a Neuro Fuzzy classification system was created. Their classification method required a lot of image pre-processing and detection processes, which took a long time. High computing costs are also a disadvantage, as they require a lot of CPU and physical memory. The classification procedure is done using Support Vector Machines in the study paper proposed by Hari Babu Nandpuru [2] and Dr. S. S. Salankar (SVM). They presented chest image categorization based on feature extraction and two additional features such as symmetry and texture. They employed 50 patients' chest pictures divided into two categories: normal and abnormal. They effectively classified the infection type using the SVM classification technique.

2.3 Analysis from Literature Review /DP ERA

Mahmud et al. [10] proposed a deep learning-based technique for the classification of COVID-19 and pneumonia infection. Features are extracted using a deep CNN model named CovXNet. A public dataset is utilized for training containing 700 samples of non-COVID-19 pneumonia and 300 samples of COVID-19 pneumonia. The model successfully classified non-COVID-19 pneumonia and COVID-19 and pneumonia with an accuracy of 90.9%.

Kuo et al. [8] use features to detect pneumonia in 185 schizophrenia patients. They applied these features in many regression and classification models, such as decision trees, support vector machines, and logistic regression, and compared the results of the models. Deep learning has a classifier called Deep Neural Network (DNN) that classifies three different types of chest infections. Their classification method required a lot of image pre-processing and detection processes, which took a long time. The disadvantage was the high computing cost due to the extensive use of memory and resources. Also, low precision.

The Support Vector Machines (SVM) approach extracted features using grey scaling and two additional features such as symmetric and texture. This problem was solved with the K-Nearest Neighbour (KNN) classifier. The accuracy of this approach based on K-NN is 87.0 percent, although it has a significant computational cost.

2.4 Methodology and Software Life cycle for this project

Project methodology is an important phase of any project because it is a key element and set the overall tone. For this we must first understand the steps that are involved in project methodology. We use iterative approach to develop this application.

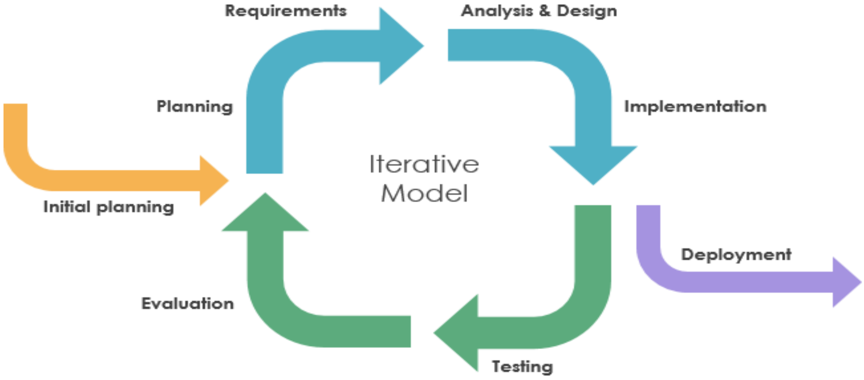

Figure 2. 1 Iterative model

2.5 Rationale behind selected methodology

Various features may be added, removed, or modified throughout the implementation phase. As a result, iterative development will be a good way to do this. Moreover, it permits:

- Working software produces quickly and accurately.
- This design is more adaptable– making changes to specifications and measurements is less expensive.
- Testing and debugging are not difficult or complex during iteration.
- Delivery costs are lowered.
- Because risks are identified during each stage, risk management is significantly simpler and easier.

Chapter 3

Requirement Analysis

1.1 Problem Definition

Our project deals with Deep learning assisted with chest infection detection. Our system aims to detect if the given X Ray image has chest infection and if found it then classifies the chest infection as COVID-19, Pneumonia and Normal

1.2 Problem Statement

Limited research has been done in multi class classification of chest infection and its detection. Our focus is to identify the problems related to the automatic diagnostics and classification of chest infection. It helps physicians to classify chest infection type. Very few existing techniques considered the problem of multi components pattern which are mostly observed in chest infection disease. There is a need to expand the different types of visual characteristics such as color, textural features etc. to distinguish the above-mentioned types of chest infection. This opens an area for researchers to propose innovative techniques that can improve the performance of chest infection detection, which will further help in diagnoses applications of chest infection.

1.3 Deliverables and Development Requirements

Following are the requirements to create our project:

- X-RAY images of infected chests by COVID, Pneumonia and Normal.
- X-RAY images of non-infection of chest.

The list of deliverable consists of following:

- Desktop application
- Documentation Development requirements include following software and hardware requirements:
- IDE: PyCharm
- Anaconda
- Dataset based work.
- PyTorch for Libraries
- 8 GB RAM system
- MS WORD
- Programming Language: Python
- Interface: PyQT Designer

1.4 Requirement Analysis

Also called requirement engineering is a process for a new or modified product. System requirement Specification (SRS) is an official statement to provide functional and operational requirements for the application. It plays a vital role as a contract between the developer and the customer. Customer is the one form whom the system is being developed. As the enlists and necessary environment specification for the system is enough, based on Specified requirements, the developer can easily build the System. System Requirement Specifications (SRS) is mainly functional based on functional and non-functional specifications which includes the proper working and responses from the system. Moreover, its proper dependence also includes make use of user's experience which are saved or run by software's.

1.5 Use Case Diagram

A use case diagram is a technique utilized as a part of framework investigation to perceive and arrange framework necessities. We design UML (Unified Modelling Language) figures to model a framework in the simple and efficient way.

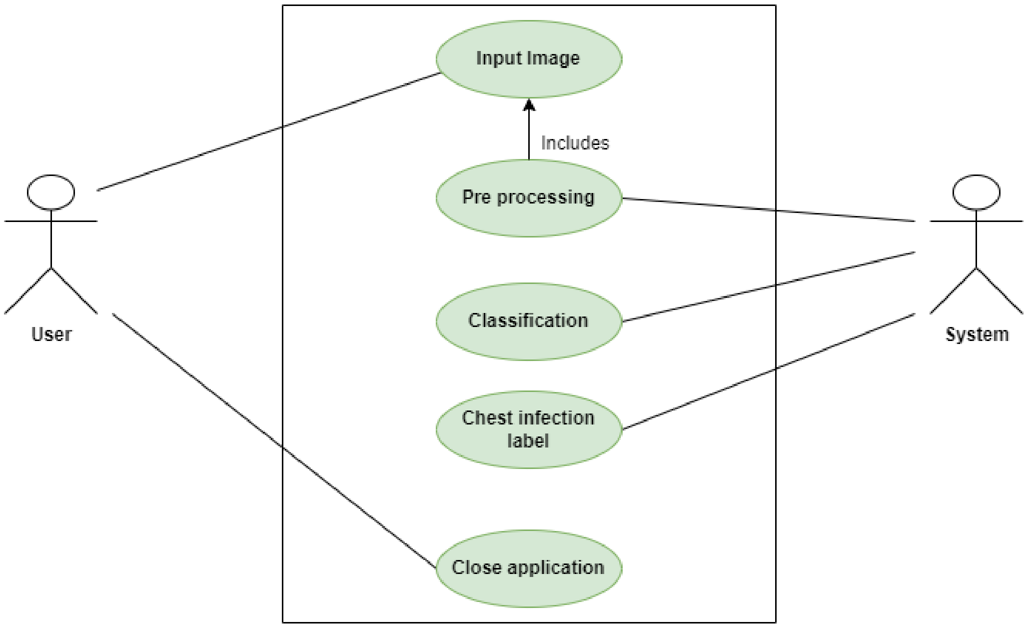

Figure 3. 1 Use case diagram

1.6 Detailed Use Case

1.6.1 Input X Ray Image

Table 3. 1 Input image (Use Case Description)

Use-case name Input X ray image	
Actor	User
Description	User will input X ray image from system directory to application as initial step to use this application.
Pre-condition	Users provide X ray Images must be present in system directory.
Post-condition	After giving an input to a system, system must start pre-processing process.

1.6.2 Pre-Processing

Table 3. 2 Pre-Processing (Use Case 2)

Use-case name Pre-Processing	
Actor	System
Description	User will input image for pre-processing.
Pre-condition	The User will upload chest infection image from directory.
Post-condition	An input image must be pre-processed.

1.6.3 Prediction

Table 3. 3 Prediction (Use Case 3)

Use-case name Prediction	
Actor	System
Description	A pre-processed image is sent for predicting the result.
Pre-condition	An image must be pre-processed.
Post-condition	After prediction, result is shown.

1.6.4 Results

Table 3. 4 Result (Use Case 4)

Identifier	FR-3
Title	Result
Requirements	The system should be able to generate result.
Dependencies	FR-3

1.7 Functional Requirements

Functional specifications provide us with an overview of calculation, technical work information, data collection and execution – some other specific features that describe the basic system structure and how is supposed to our system capable to do the diagnosing task in automated manner. In other words, a functional requirement will elaborate us a specific behaviour of function of the system when requirements that supposed to do are met. Following are some basic functional requirements of our project:

1.7.1 Functional Requirement 1

Table 3. 5 FR 1

Identifier	FR-1
Title	Input Image
Requirements	The system upload image from directory
Dependencies	None

1.7.2 Functional Requirement 2

Table 3. 6 FR2

Identifier	FR-2
Title	Pre-processing
Requirements	System will already upload in system and .jpeg form.
Dependencies	FR-1

1.7.3 Functional Requirement 3

Table 3. 7 FR-3 Classifier CNN

Identifier	FR-3
Title	classifier
Requirements	The system performs the classification and predict the chest infection detection label.
Dependencies	FR-2

1.7.4 Functional Requirement 4

Table 3. 8 FR-4 output

Identifier	FR-3
Title	Result
Requirements	The system should be able to generate result.
Dependencies	FR-3

1.8 Non-Functional requirements

Non-functional requirements show and specify how the system performs a certain functionality within scope of its requirements specified. In other words, behavior and limit of system based on its functionality describe that how a system will perform.

Non-functional requirements generally specify the system quality attributes or characteristics. Following are some non-functional requirements to ensure our system efficiency.

1.8.1 Usability

The presentational highlights of the application will be structured easy to use with least preparing required to utilize the application.

1.8.2 Reliability

The framework will be accessible 100% of the time.

1.8.3 Integrity

Integrity requirements define the security features of framework, limit access to features or information to a specific user and ensure the protection of information in the software.

1.8.4 Flexibility

This system required less effort to improve any operational program.

1.8.5 Maintainability

The effort required to locate and fix the error will be maintained by software development team.

1.8.6 Accuracy

The software has definite elegance and appropriate flow with an obvious presence.

1.8.7 Robustness

The system will not allow the inappropriate user interaction and any wrong input. The system interface provides guidance message in case of user make an error.

1.8.8 Predictability

The framework can never crash. The framework must create unsurprising outcomes.

1.8.9 Aesthetics

The framework won't allow to the user enter any invalid information.

1.8.10 Human Factors

GUI is convenient & captivating.

Chapter 4

Design and Architecture

4.1 Design and Architecture

Following the collection of all requirements, the second process is to begin planning how we will create our project, including how many resources, costs, time, benefits, and other items will be necessary. Following the planning phase, we go on to the designing and architectural phase to determine which approaches and methods we can employ and how we will proceed with our project. This is the most difficult stage of project development.

4.1.1 System Architecture

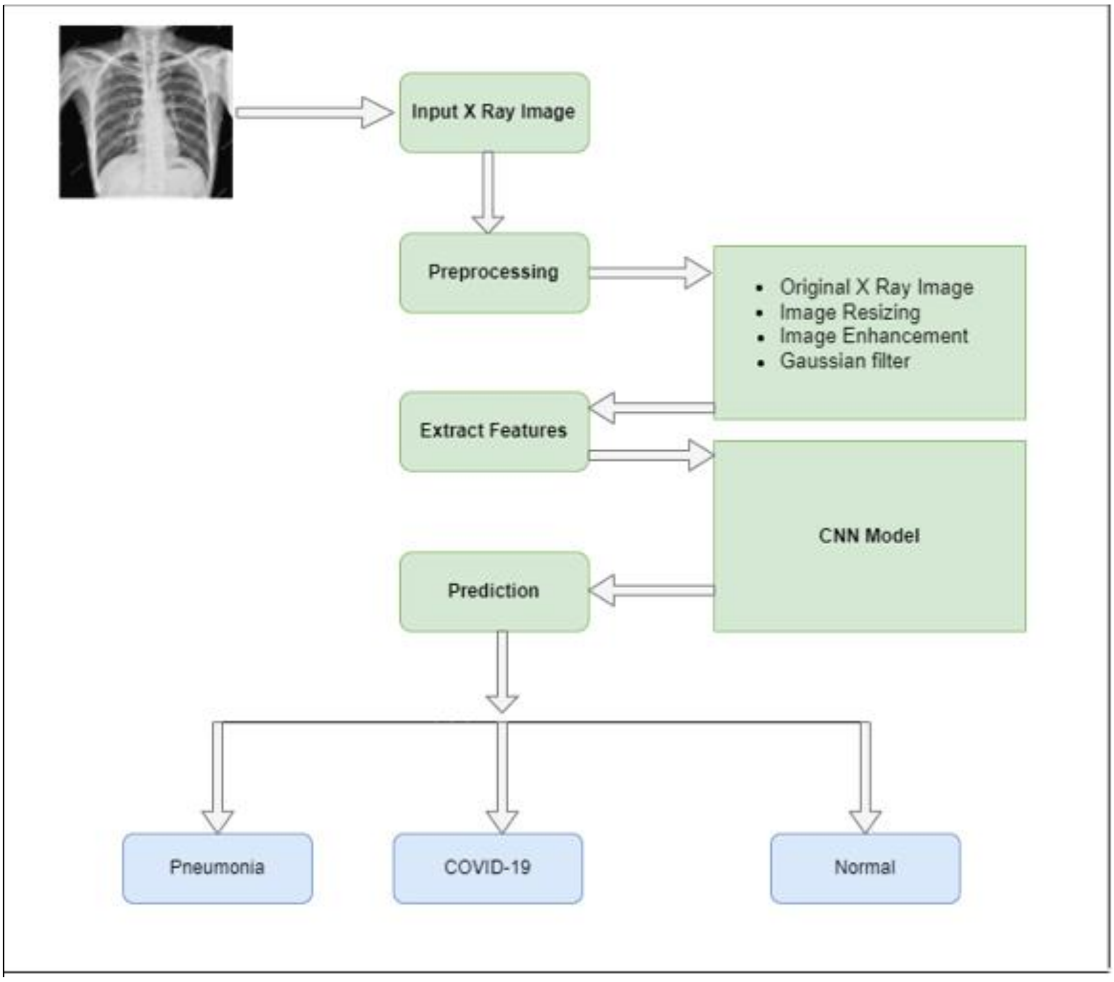

Figure 4. 1 System Architecture

4.2 Data Representation

A data flow diagram is graphical interpretation of information move from a data framework is called data flow diagram. A DFD is used for basic step to create an overview of the system without going into great aspect, which can later be elaborated.

4.2.1 Data Flow Diagram (Level 0):

Level 0 tells us about the least detail about a system and shows the external view of the system that how a system seems to be from the outside. In this diagram, it is shown that in this application there are two basic entities a user and the system. When the user provides an image to the system, system return output.

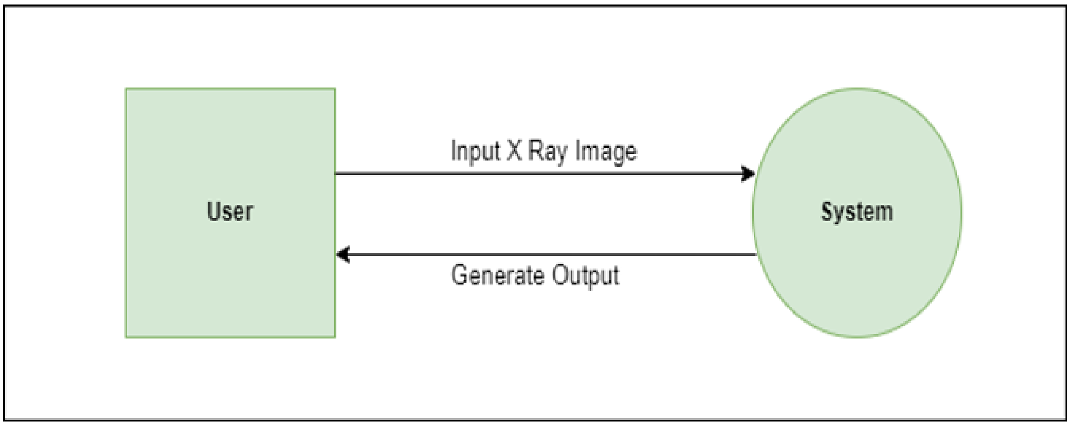

Figure 4. 2 DFD level 0

4.2.2 Data Flow Diagram (Level 1):

Level 1 show internal and more detail of a system comparatively to the level 0

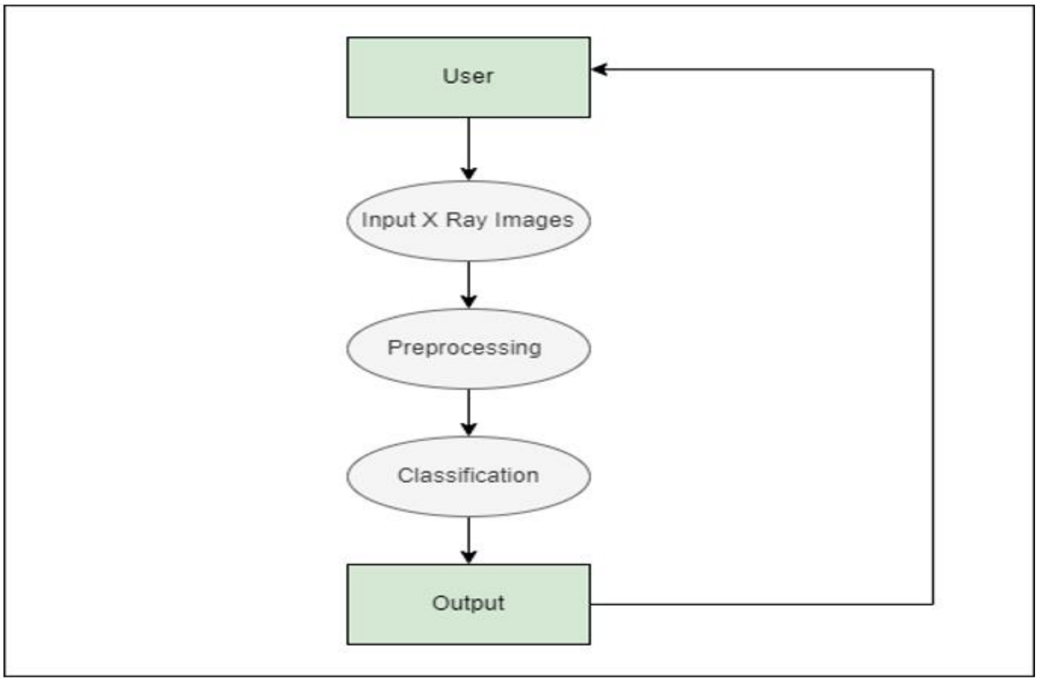

Figure 4. 3 DFD Level 1

4.3 Process Flow Representation

4.3.1 Activity Diagram:

Activity diagram shows the behavior of a system that how the process flows from one activity to another.

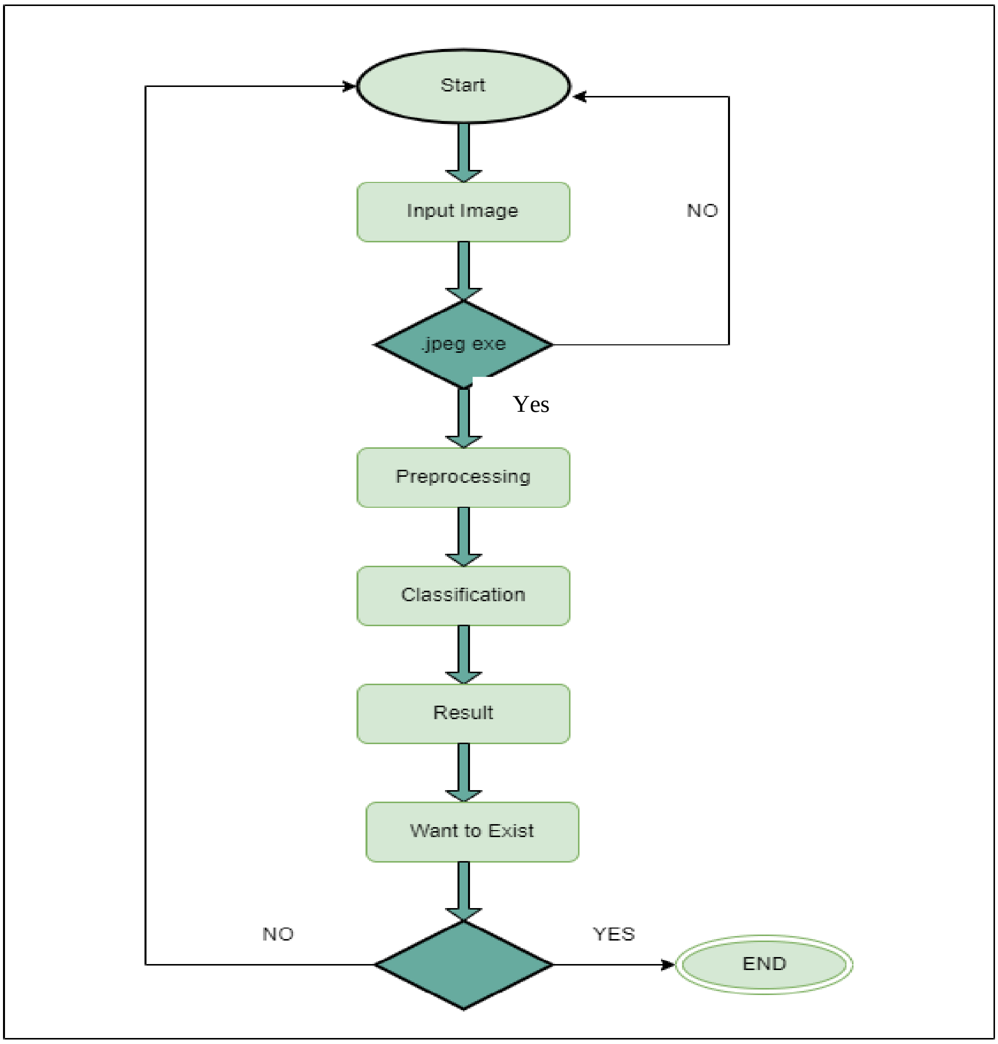

Figure 4. 4 Activity Diagram

4.4 Design Models

4.4.1 Sequence Diagram

Sequence diagram shows interaction between objects in a system and the messages that pass between them and shows the order in which they work together. In the given diagram.

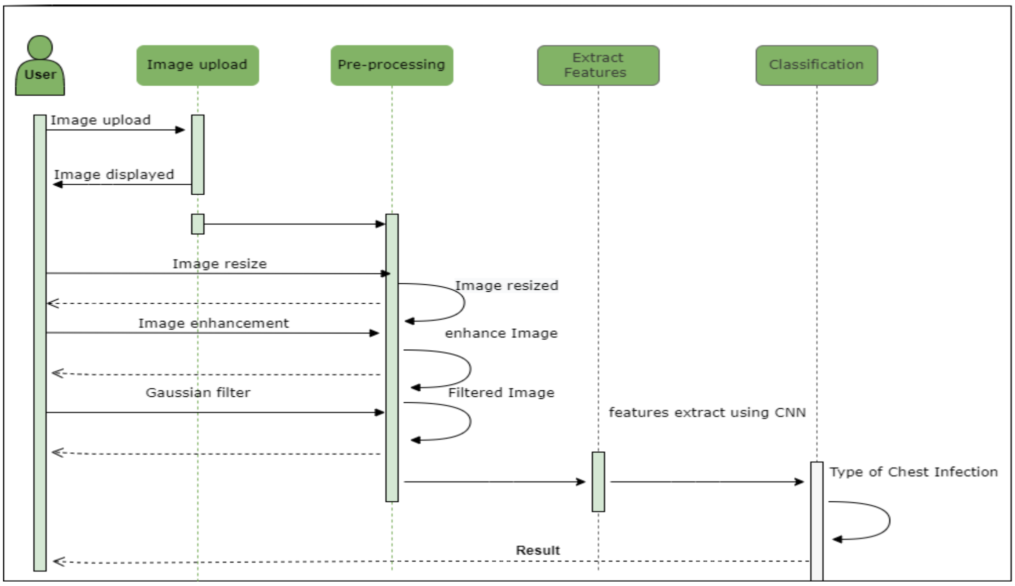

Figure 4. 5 Sequence Diagram

4.5 Class Diagram

In Unified Modeling Language (UML), class diagram is a static diagrammatic structure representation in which we explain classes, functions, attributes and relationship between them

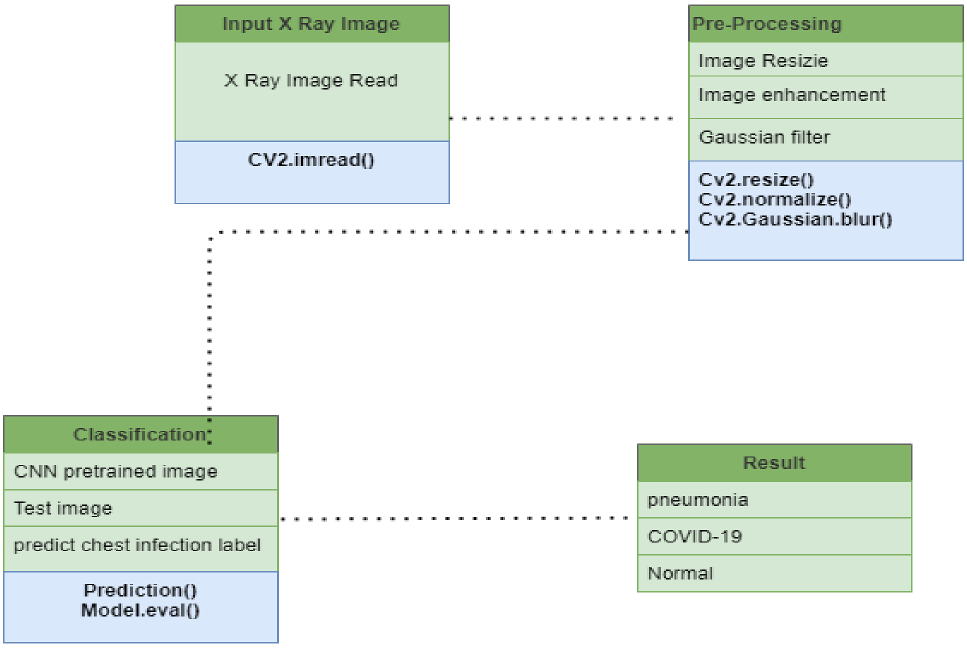

Figure 4. 6 Class Diagram

Chapter 5

Implementation

5.1 Implementation

To run a system properly, the back-end coding, front end interface, and their connectivity must all be implemented. This section covers all the back-end coding required to build that project. It also comprises the system's graphical user interface (GUI). At this point, all the features have been merged to provide a meaningful image. Deep learning application in the field of chest infection detection could be the most appropriate technique to categories and recognize chest infection images, which could be highly useful in the field of medicine for early diagnosis and improved accuracy.

5.2 Algorithm

The algorithms that are integrated are discussed here.

5.2.1 Dataset

Three chest infection classes are included in the dataset which are Covid, Viral pneumonia and Normal. It contains COVID (500 images), viral pneumonia (850 images) and Normal (561 images) The size of CXR image is 512 x 512 dimensions All the images are in (.jpeg) format.

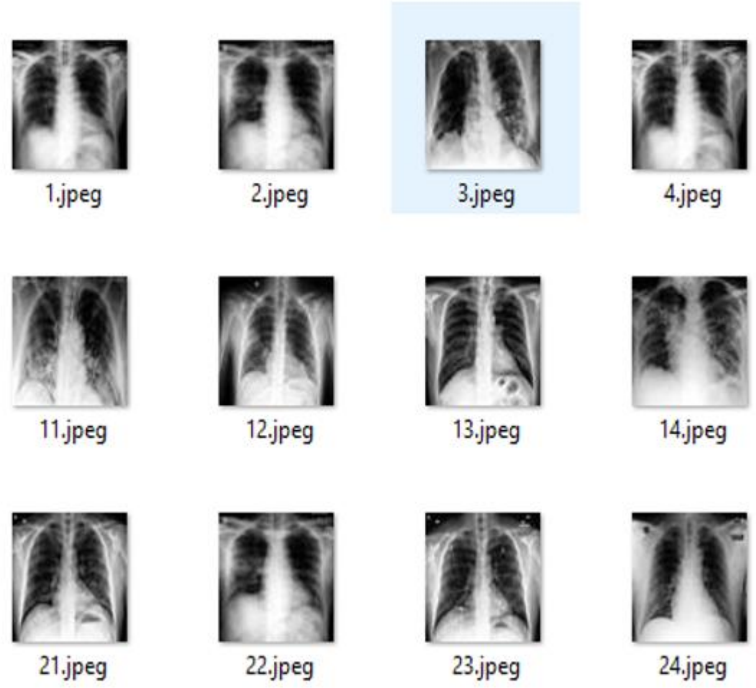

Figure 5. 1 Dataset

5.2.2 Preprocessing

Preprocessing is the framework's first and most important step. It's necessary for removing undesired noisy data from the CXR image. When it comes to image pre-processing, data must be clean and consistent. To produce a noise-free image for subsequent processing, raw data (image) with irrelevant artefact is filtered. The image is converted to a grey level
scale.

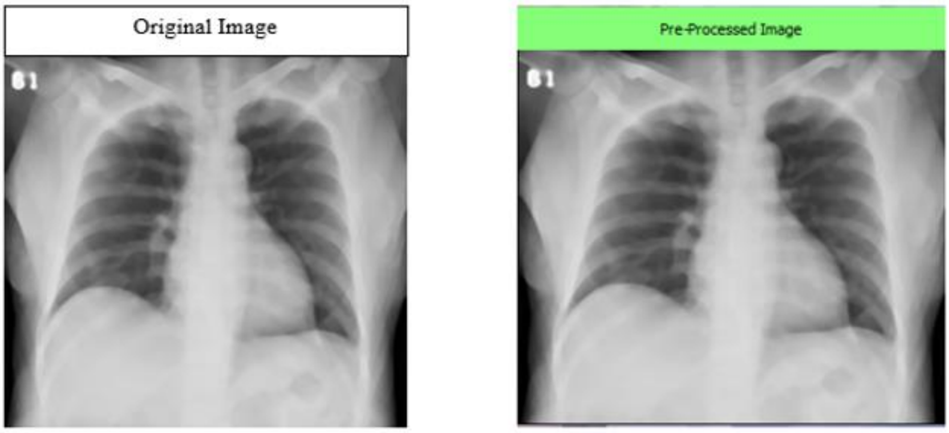

Figure 5. 2 Preprocessing

Following techniques are applied to clean noise from images:

5.2.2.1 Re-sizing

It refers to resizing an image in terms of pixels. When the image dimensions are changed from original to standard, all images are processed at the same period. Each image was scaled to 64 X 64 pixels. It's critical to have all input test images the same size so that they all have the same features.

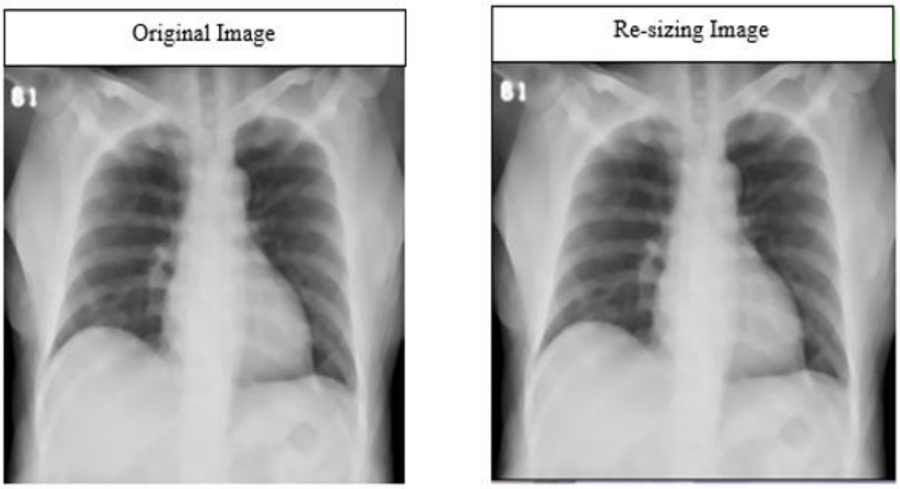

Figure 5. 3 Resizing Image

5.2.2.2 Image Enhancement

Image enhancement is used to make the image clearer and sharper than previously. Image normalizing, which changes the digital value of images, is used for this purpose. To make the image more normal to the senses, the range of pixel intensity values is increased. Its goal is to enhance image contrast to remove noise.

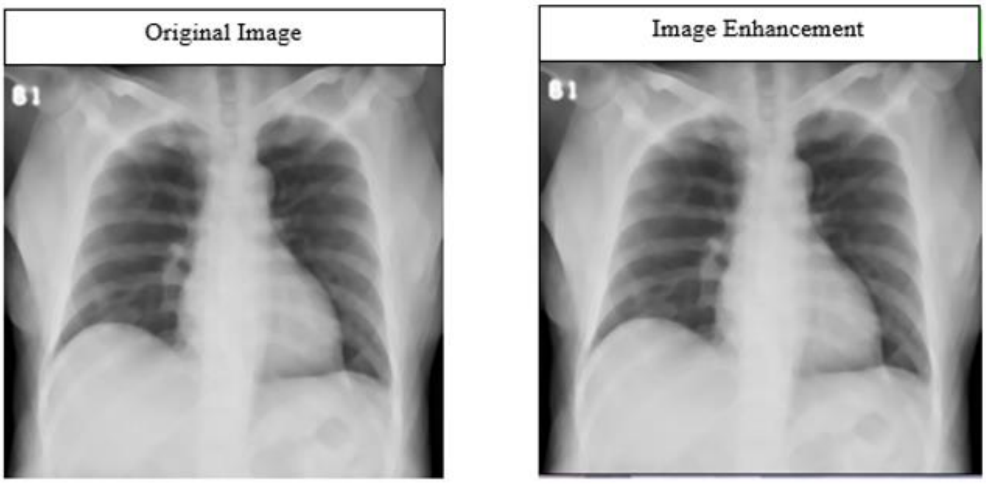

Figure 5. 4 Enhancement

5.2.2.3 Gaussian Filter

It used to produce blur in the image by Gaussian function for removing Gaussian noise. It does not preserve the edges sharp like median filter.

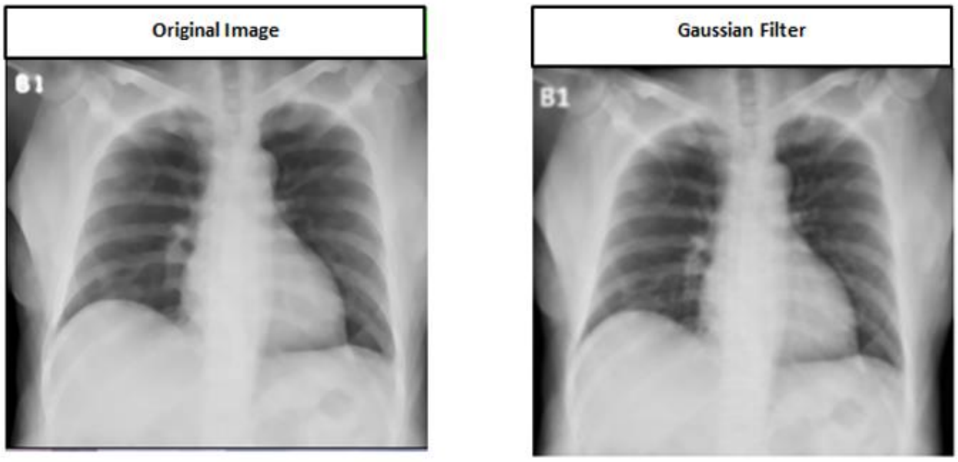

Figure 5. 5 Gaussian filter

5.2.3 Feature Extraction

The properties of X-ray images can be divided into edges and color. Feature extraction is a significant step in any object recognition algorithm. It refers towards the way of extracting useful information which is then referred to as 'feature' from an input image. The extricated features should be delegate in nature, conveying critical and unique attributes of the image. We used Convolutional Neural Network that automatically performs feature extraction and pass those features for classification.

5.2.4 Classification

We used Convolutional Neural Network (CNN) classifier to classify chest infection detection into three types. Our model is trained on dataset and then test image is matched with pretrained model to identify the class of chest infection.

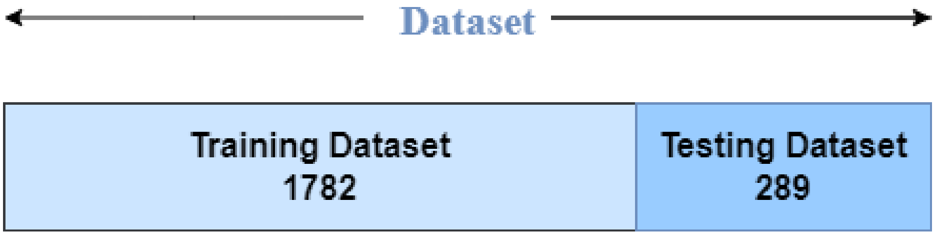

Figure 5. 6 Classification

5.2.5 Prediction

5.2.5.1 CNN Architecture

We implemented simple architecture proposed by Nyoman Abiwinanda [5].The feature extraction is done by CNN from the pixels of Chest X Ray images using three main operations:

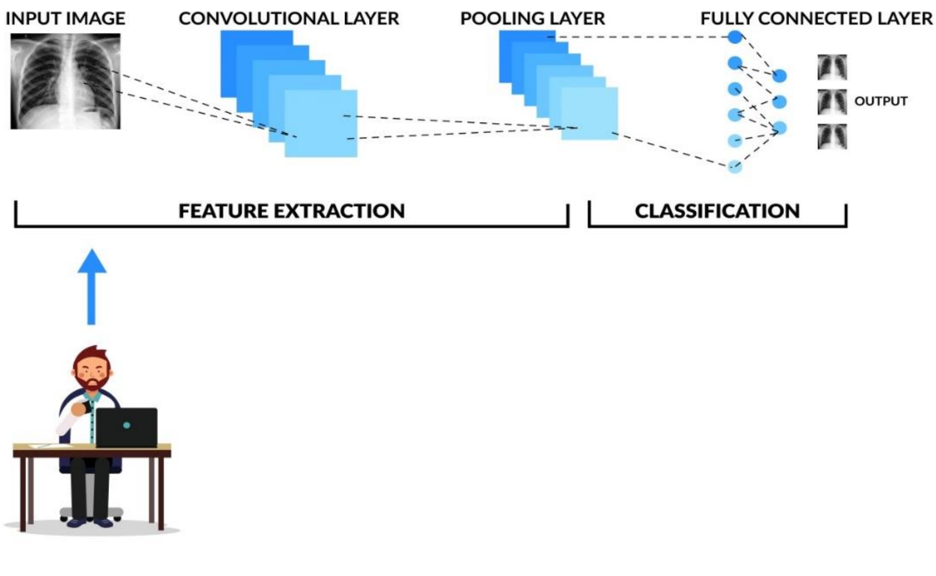

Figure 5. 7 CNN Architecture

The following is how CNN's architecture is implemented:

5.2.5.1.1 Convolution

Each input image is passed to series of convolution layers with filters. We use 2 convolutions. In first convolution layer we use 32 channels with 3x3 kernel size;

And in 2nd convolution layer we use 64 channels with 4x4 kernel size.

5.2.5.1.2 Pooling

Max pooling is used for down sampling the image. Down sampling operation reduce the dimension and preserve the features of tumor image. We use 1 max pooling layers.

5.2.5.1.3 Fully connected.

In our model, we used 2 fully connected layers.

5.2.5.2 Implementation

5.2.5.2.1 Input Dataset

The PyTorch library is used to load the dataset into the algorithm.

5.2.5.2.2 CNN Model

Convolutional layers, max pooling and fully connected layers are used.

5.2.5.2.3 Training

The images are separated into two categories: training and testing . The data was trained using 1,783 images. All aspects of the training phase have been completed, including weight update, loss error function optimization, and algorithm optimization. This process went on. The model is saved with the pyTorch (.PTH) file extension after training.

5.2.5.2.4 Testing

For testing, we used 289 images from each chest infection class. Test images are reviewed and classified using a pre-trained model.

5.2.5.2.5 Output

When all of the steps are completed, the accuracy of each individual class is displayed on the GUI display.

5.2.5.2.6 Accuracies

Table 5. 1 Accuracies

Chest Infection type	Overall Accuracy
Accuracy	96%
Testing Accuracy	84%

5.2.5.2.7 Graphical Representation

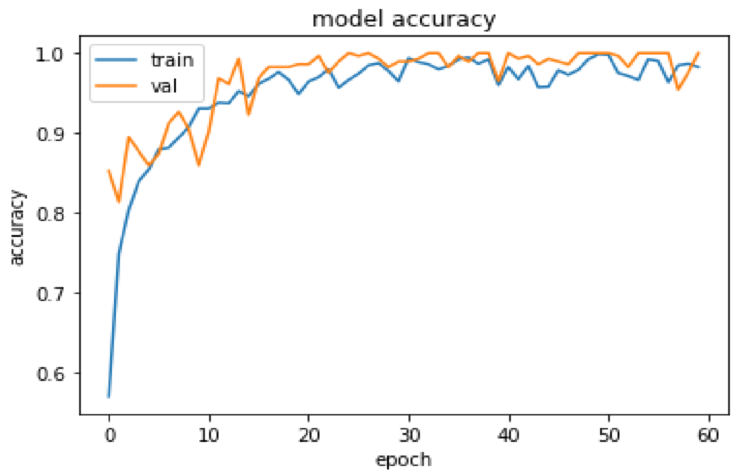

Figure 5. 8 Accuracy Graph

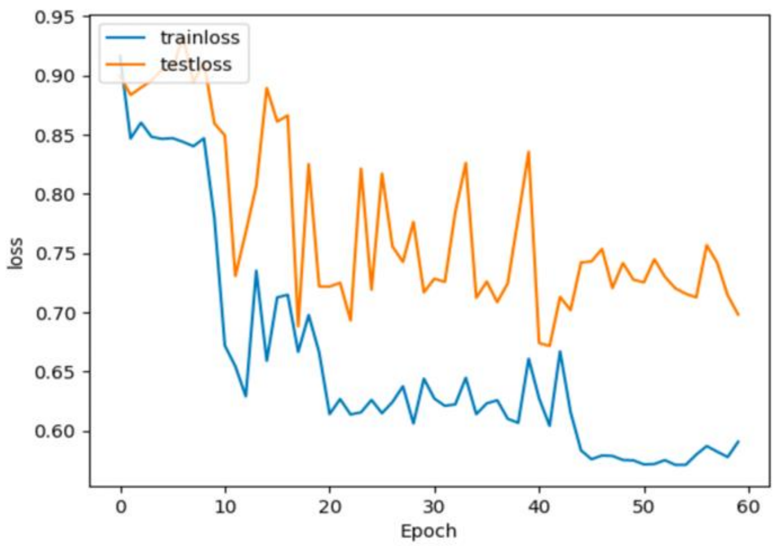

Figure 5. 9 Loss Graph

5.3 Interface

The design of application is the major part because; it should be interactive and design in such a way that user did not find any problem while using. Our front end is designed in PyQt Designer in python. PyQt installer comes with a GUI tool (Qt Designer). It is interactive drag and drop sort of interface provide facility for making attractive GUIs

All windows of interface are shown in order:

5.3.1 Main Window

This is the main window of our project in which the image will be uploaded as an input to the system and can be changed or deleted.

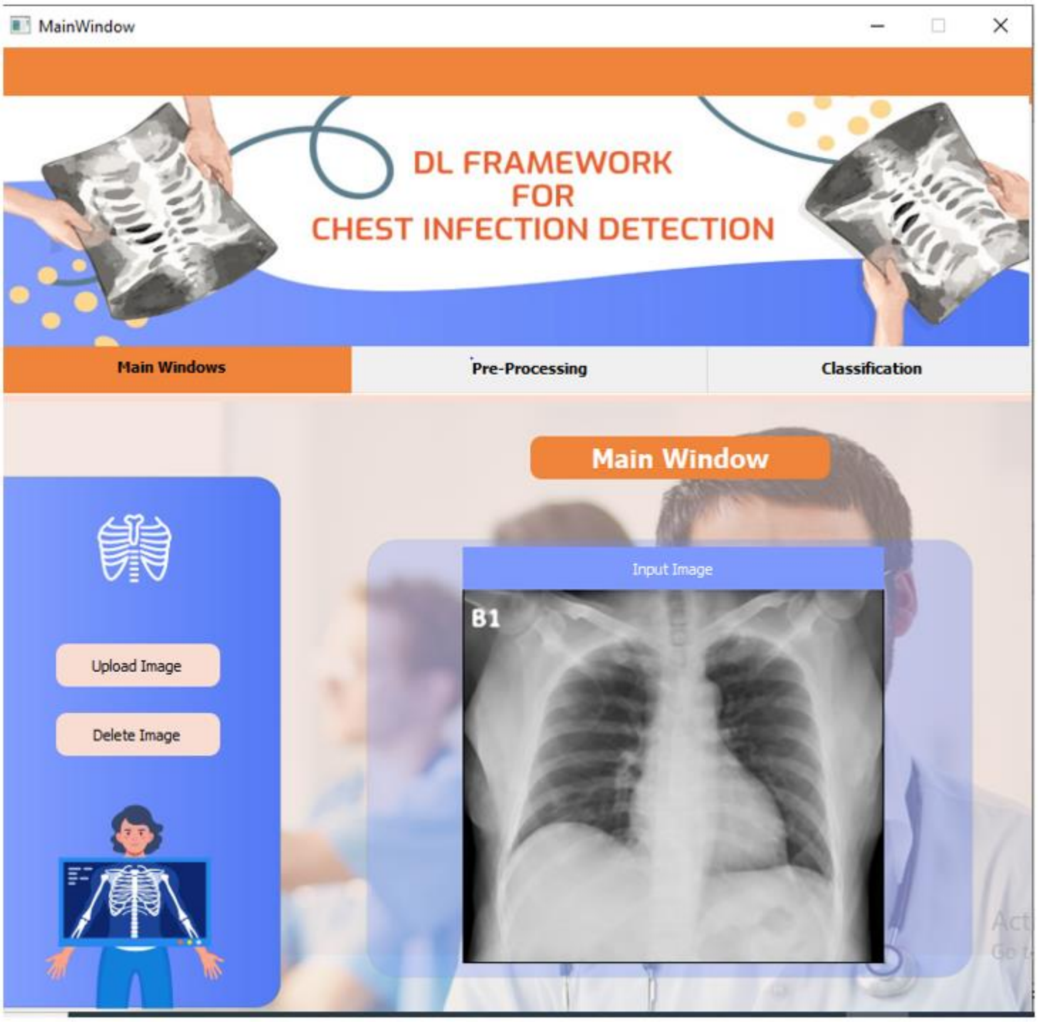

Figure 5. 10 Main Window

5.3.2 Preprocessing

In this window, all the operations of pre-processing will take place. For example: Resizing the image, apply gray scale filter, enhancing the image, applying median filter and then the clear all option. We do pre-process on an image to make it sharp and to remove any kind of noise it has.

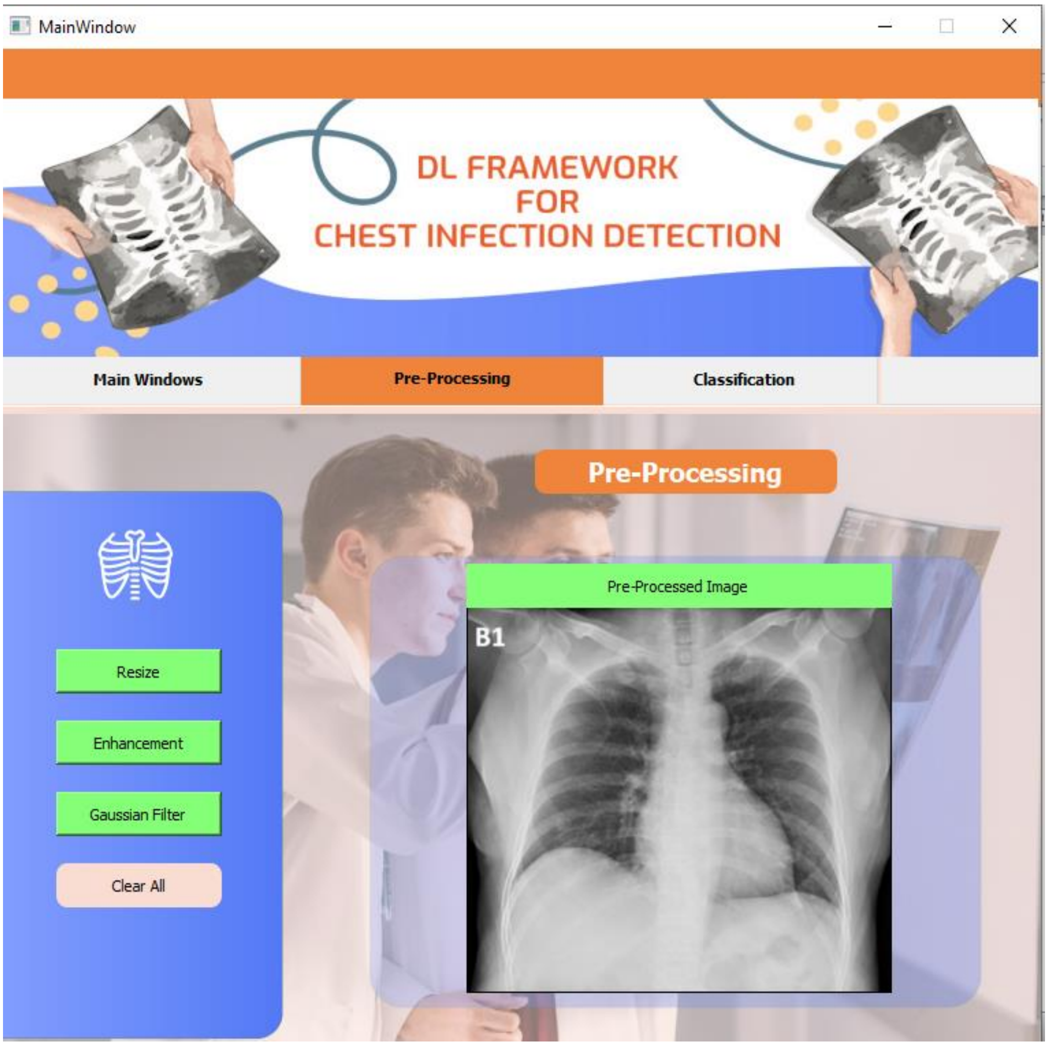

Figure 5. 11 Preprocessing

5.3.3 Classifier

The main function of classifier is to predict the real/accurate result for the given input data. It classifies and predicts about the occurrence of disease using CXR images based on feature extraction techniques.

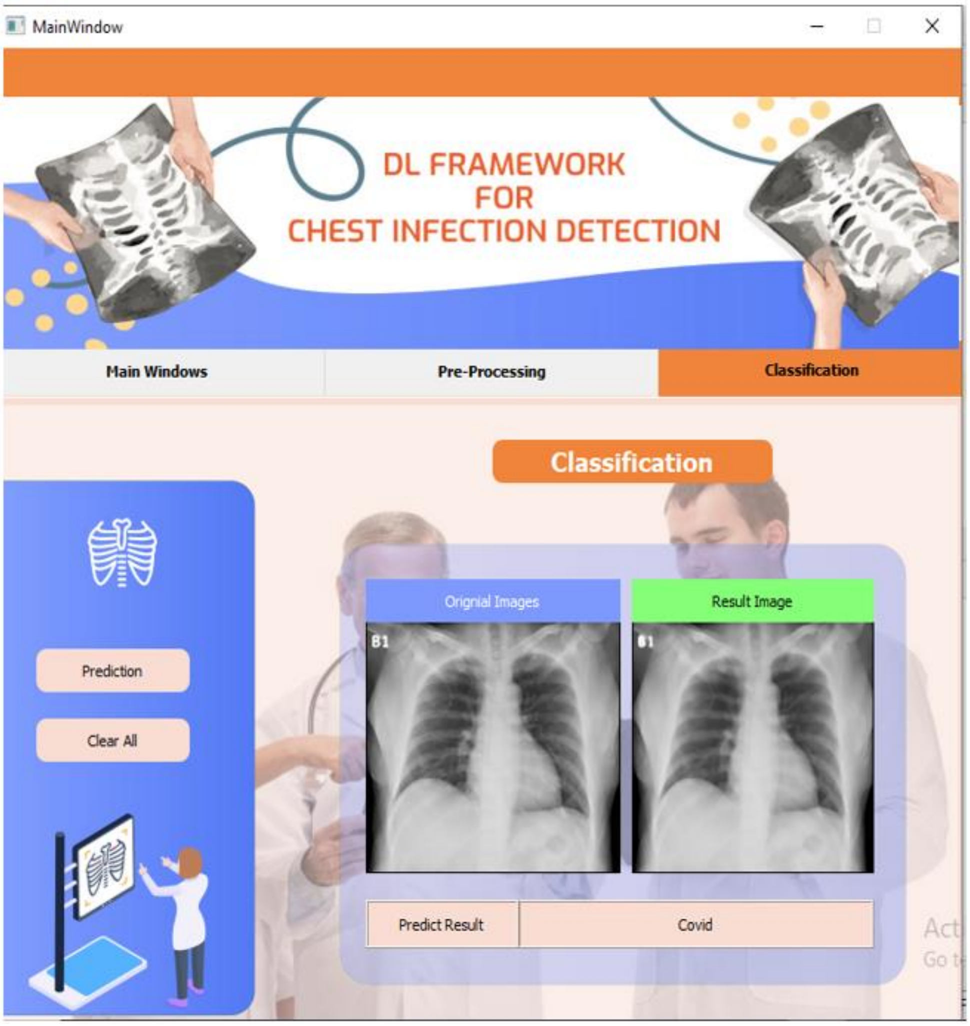

Figure 5. 12 Classifier

Chapter 6

Testing and Evaluation

5.1 Testing and Evaluations

The system is tested using a set of tests to check that the project is effective properly and to determine its capabilities and limitations. The evaluation stage is necessary because it is here that we evaluate well how our system is working (predicting). Debugging faults and errors is necessary for each module.

5.2 Manual Testing

This form of testing involves manually testing the entire system without the use of any tools. It's the first time the system has been tested since its implementation. All lines of code are checked, and if any errors or bugs are identified, they are fixed. The needs of radiologists are noted at the outset, and during manual testing, all the requirements are cross matched to see if the system is producing the desired results.

5.2.1 System Testing

This type of testing is performed to verify all the functions of system. It starts from browsing an image from directory to interface and by going through several algorithms, our system displays an output at the end i.e., chest infection type whether it is viral pneumonia, Covid-19 and Normal.

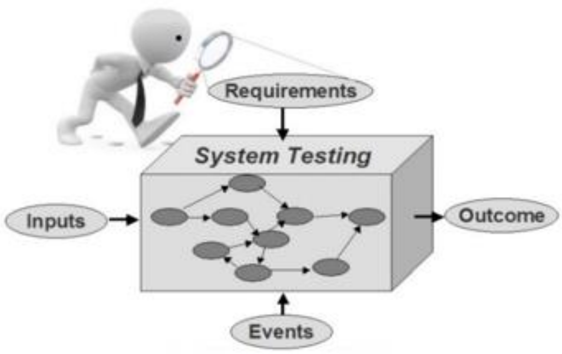

Figure 6. 1 System Testing

5.2.2 Unit Testing

In unit testing, all units of our system are tested and checked whether the system is according to the user needs or not.

Figure 6. 2 Unit testing

5.2.3 Input image

When the button ‘Input image’ is clicked then it opens the image directory. Any image from directory can be selected and uploaded on graphical user interface. The format of image must be (.jpeg) format.

5.2.3.1 Preprocessing

Three buttons named as *‘image resizing’*, ‘*image enhancement’* and ‘*gaussian filter*’ are clicked in any order and image is free from noise. There are three classes of chest infection selected for this project. All the images in dataset have numeric label with them. Before preprocessing, all images in (.jpeg) format are available in the directory and following algorithms are applied on images:

- Image resizing.
- Contrast enhancement.
- Gaussian filter

5.2.3.2 Feature Extraction

We used Convolutional Neural Network that perform feature extraction automatically.

5.2.3.3 Prediction

Then ‘*prediction*’ button is clicked, test image is compared with the trained model and result is predicted. The output image is in .jpeg format

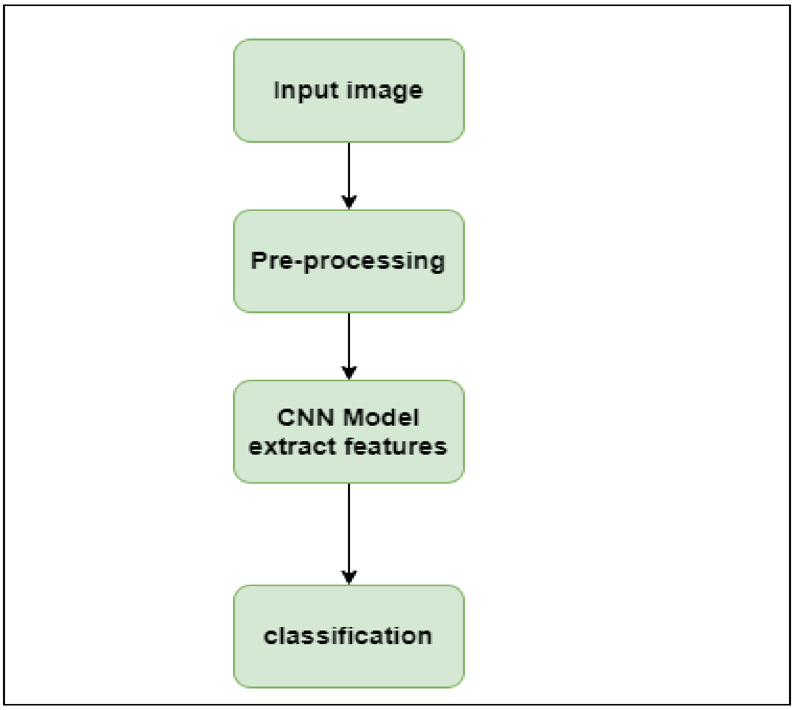

Figure 6. 3 Units of project

5.2.4 Functional Testing

In functional testing, all the buttons are tested that whether the attached functionality works according to demands of user or not. All the working of algorithms is working at the backend of buttons present on the graphical user interface.

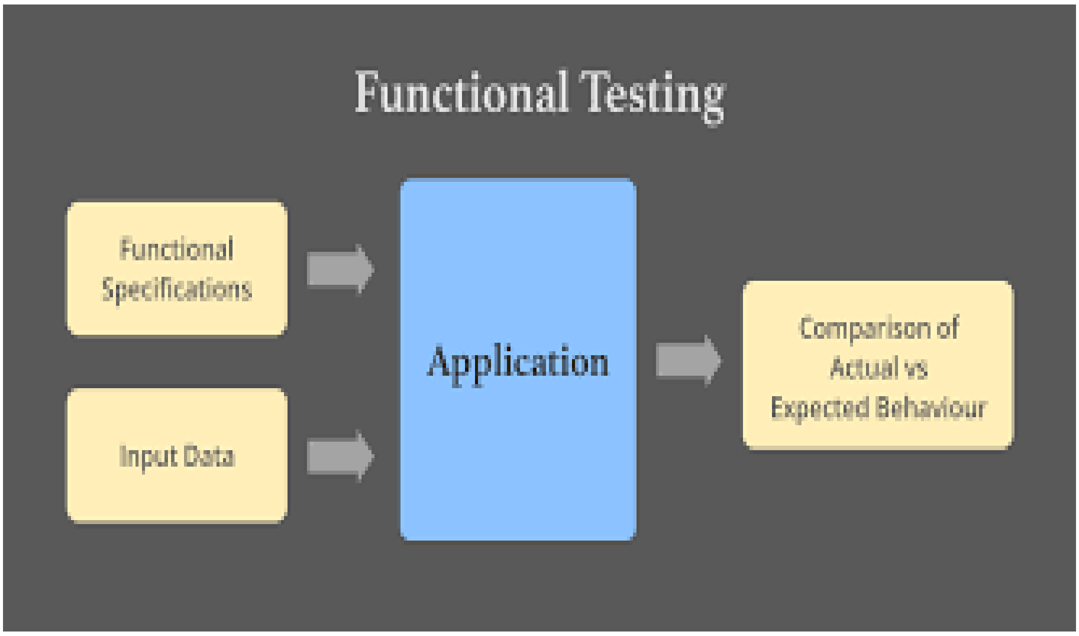

Figure 6. 4 Functional test

5.2.4.1 Image upload

Table 6. 1 Input image

No.	**Test case**	**Attribute and value**	**Expected result**	**Result**
1	Input image	User input image in" jpeg" format from folder.	Image should be displayed on screen	Image displayed on screen

5.2.4.2 No image uploads

Table 6. 2 No image uploads

No.	**Test case**	**Attribute and value**	**Expected result**	**Result**
1	No input image upload	If button pressed without image upload	Image should not be displayed on screen	Image should not displayed on screen

5.2.4.3 Pre-processing

Table 6. 3 pre-processing

No.	Test case	Attribute and value	Expected result	Result
1	Image Resizing	Image must be present in jpeg format in folder	On button click, output image should be resized and displayed.	Output image scaled and displayed.
2	Image Enhancement	Image must be present in jpeg format in folder.	On button click, output image should be contrast enhanced and displayed.	Output image contrast enhanced and displayed.
3	Gaussian Filter	Test image present in folder	Image displayed	Image noise removed and Displayed

5.2.4.4 Classifier

Table 6. 4 Classification

No.	Test case	Attribute and value	Expected result	Result
1	Prediction	Input image matched with trained load model.	Predicted the Chest infection disease label	Image label predicted displayed on screen.

5.2.4.5 No Chest Infection

Table 6. 5 No chest infection

No.	Test case	Attribute and value	Expected result	Result
1	No chest infection	Any image presented	No chest infection label predicted	Prediction not successful.

5.2.5 Integration Testing

- Interfacing of various modules can be difficult.
- Data loss can occur across an interface, one module may affect the other, and individually acceptable imprecision may be magnified when combined.

Table 6. 6 Integration Testing

No.	Test case	Attribute and value	Expected result	Result
1	Input image	User input image of CXR in jpeg format from folder.	Image is ready to be displayed on screen.	Image displayed on screen.
2	No input image	If button pressed without image upload	Image should not be displayed on screen	Image should not displayed on screen
3	Pre-processing .	Result image present in folder.	All operation buttons are pressed	Image pre-processed successfully.
4	Prediction	Match test image with pretrained load model.	On button click, label and other details should be displayed on screen.	Image label and other details displayed on screen.
5	No chest infection	Any image presented	No chest infection label predicted	Prediction not successful.

5.3 Automated Testing

This testing can be performed by any automated tool

5.4 Tools

Table 6. 7 Tools

Tool Name	Tool Description	Applied on [list of related tests cases / FR / NFR]	Results
PyCharm	JetBrains PyCharm Community Edition 2019.2.1	All functionalities.	Code
Anaconda	Anaconda3-2019.10-Windows-x86_64	All functionalities.	Supports libraries.
PyQt Designer	Library (PyQt5) of python.	Designing of interface.	Frontend
Microsoft Word	Word 2016	Creation of documentation.	Documentation.
Microsoft PowerPoint	PowerPoint 2016	Creation of presentation slides.	Presentation slides.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

After study and implementation of deep learning in the field of chest infection disease it is concluded that deep leaning while using Convolutional Neural Network (CNN) provide very competitive results. First these images are taken from “Kaggle”. Then the model is trained for the test of three different chest infection classes. After training the model was able to classify the chest infection disease images into different classes. We're working on a user-friendly graphical user interface. Input image, preprocessing ('image scaling’, ‘image enhancement',' Gaussian filter') and prediction are the six processes. We implemented a Convolutional Neural Network to train our model, and then the testing image was matched with the trained model to predict the result. Because CNN does not require region segmentation, it produces good results and accuracy even without it. There are numerous methods for detecting chest infections in CXR images. However, this procedure is quick and effective.

Figure 7. 1 Conclusion

For this current work 96% accuracy was achieved with a CNN architecture. The high percentage is largely due to the usage of CNN and epochs. Compared to previous works, this work uses a greater number of images to achieve better performance. So nowadays deep learning is the best solution of chest infection diseases classification and recognition of chest infection diseases.

7.2 Future Work

The project will be fully implemented in the future, with all components being programmed, tested, and deployed at an industrial level. There are several chest infection diseases around the world, thus new chest infection classes might be added, and the system's general accuracy could be enhanced. The overall classification accuracy can be increased with a larger data collection. Graphical comparison of abnormal and normal CXR can be incorporated to the system because there are so many algorithms available for prediction and image processing. This technology can be used in clinics to replace manual CXR image reading for the detection of chest infections. It can save time and is less prone to human mistake, which can assist doctors in making better decisions about patient treatment.

Chapter 8

References

Appendix A

8.1 References

[1] R. H. Abiyev and M. K. S. Ma'aitah, "Deep convolutional neural networks for chest diseases detection," *Journal of Healthcare Engineering*, vol. 2018, Article ID 4168538, 11 pages, 2018.View at: Publisher Site | Google Scholar

[2] R. M. Pereira, L. O. Teixeira, D. Bertolini, C. N. Silla, and Y. M. G. Costa, "Covid-19 identification in chest X-ray images on flat and hierarchical classification scenarios," *Computer Methods and Program in Biomedicine*, vol. 194, 2020.View at: Google Scholar

[3] R. Abiyev, M. Arslan, J. B. Idoko, B. Sekeroglu, and A. Ilhan, "Identification of epileptic eeg signals using convolutional neural networks," *Applied Science*, vol. 10, no. 12, 2020.View at: Publisher Site | Google Scholar

[4] R. H. Abiyev and M. Arslan, "Head mouse control system for people with disabilities," *Expert System*, vol. 37, no. 1, pp. 1–14, 2020.View at: Publisher Site | Google Scholar

[5] J. Bush, R. Abiyev, and M. Arslan, "Impact of machine learning techniques on hand gesture recognition," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 3, pp. 4241–4252, 2019.View at: Publisher Site | Google Scholar

[6] K. Fukushima, "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.View at: Publisher Site | Google Scholar

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.View at: Publisher Site | Google Scholar

[8] H. Gunraj, L. Wang, and A. Wong, "COVIDNet-CT: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest CT images," *Frontiers in Medicine*, vol. 7, pp. 1–12, 2020.View at: Google Scholar

[9] Z. Han, B. Wei, Y. Hong et al., "Accurate screening of COVID-19 using attention-based deep 3D multiple instance learning," *IEEE Transactions on*

Medical Imaging, vol. 39, no. 8, pp. 2584–2594, 2020.View at: Publisher Site | Google Scholar

[10] Rajaraman, J. Siegelman, P. O. Alderson, L. S. Folio, L. R. Folio, and S. K. Antani, "Iteratively pruned deep learning ensembles for COVID-19 detection in chest X-Rays," *IEEE Access*, vol. 8, pp. 115041–115050, 2020.View at: Publisher Site | Google Scholar

[11] K. Hammoudi, H. Benhabiles, M. Melkemi et al., "Deep learning on chest X-ray images to detect and evaluate pneumonia cases at the era of COVID-19," *Journal of Medical Systems*, pp. 1–10, 2021.View at: Google Scholar

[12] H. Ko, H. Chung, W. S. Kang et al., "COVID-19 pneumonia diagnosis using a simple 2d deep learning framework with a single chest CT image: model development and validation," *Journal of Medical Internet Research*, vol. 22, no. 6, pp. 1–13, 2020.View at: Publisher Site | Google Scholar