Classification of Pneumonia and Tuberculosis from Chest X-rays

Submitted by

Muhammad Abubakar
(AUIC-FL16-BSCS-1406)

Hassan Ali
(AUIC-FL16-BSCS-1387)

Supervised by
Mr. Waqar Ali

Co-Supervised by
Mr. Farrukh Bashir

This project is submitted in partial fulfillment of requirement for the degree of Bachelor of Science in Computer Science (BSCS).

Department of Computing
Abasyn University Islamabad Campus (AUIC)

July 2020

Final Approval

This is certified that we have studied this project report, titled **"Classification of Pneumonia and Tuberculosis from Chest X-rays"**, submitted by **Muhammad Abubakar** and **Hassan Ali**, university Registration Number **AUIC-FL16-BSCS-1406** and **AUIC-FL16-BSCS-1387**, respectively. We conclude that this project report is of sufficient standard to warrant its acceptance by the Abasyn University Islamabad Campus (AUIC) for the award of the degree of Bachelor of Science in Computer Science (BSCS).

Committee

Supervisor

Name: Waqar Ali
Designation: Lecturer
Institute: Abasyn University Islamabad Campus (AUIC)
Signature: __________

Co-Supervisor

Name: Farrukh Bashir
Designation: Lecturer
Institute: Abasyn University Islamabad Campus (AUIC)
Signature: __________

Internal Examiner

Name: Abdul Basit
Designation: Assistant Professor
Institute: Abasyn University Islamabad Campus (AUIC)
Signature: __________

Internal Examiner

Name: Dr. Ali Arshad
Designation: Head of Department
Institute: Abasyn University Islamabad Campus (AUIC)
Signature: __________

Head of Department

Name: Dr. Ali Arshad

Signature: __________

Abstract

Artificial intelligence (AI) and specifically machine learning is making inroads into number of fields. Machine learning is replacing and/or complementing humans in a certain type of domain to make systems perform tasks more efficiently and independently. Healthcare is a worthy domain to merge with AI and Machine learning to get things to work smoother and efficiently. The X-ray based detection and classification of diseases related to chest is much needed in this modern era due to the low number of quality radiologists. This thesis focuses on the classification of Pneumonia and Tuberculosis – two major chest diseases – from the chest X-rays**.** This system provides an opinion to the user whether one is having a disease or not, thereby helping doctors and medical staff to make a quick and informed decision about the presence of disease.

Declaration

We certify that this project work titled "Classification of Pneumonia and Tuberculosis from Chest X-rays" is our own work. No portion of the work presented in this project has been submitted in support of another award or qualification either at this institution or elsewhere. Where material has been used from other sources, it has been properly acknowledged/referred. If any part of this system is proved to be copied or found to be a report of some other, we will stand by the consequences.

Muhammad Abubakar

AUIC-FL16-BSCS-1406

Hassan Ali

AUIC-FL16-BSCS-1387

Submission and Copyrights

This project report is being submitted to the Department of Computing, Abasyn University Islamabad Campus (AUIC) as a partial fulfillment of the requirements for the award of the degree of Bachelor of Science in Computer Science (BSCS).

The authors hereby grant permission to Abasyn University Islamabad Campus (AUIC), to reproduce and distribute publicly paper and electronic copies of this thesis and to grant others the right to do so.

Muhammad Abubakar

AUIC-FL16-BSCS-1406

Hassan Ali

AUIC-FL16-BSCS-1387

Farrukh Bashir

Co-supervisor

Waqar Ali

Supervisor

Acknowledgment

This project involved a lot of research work and dedication. If we did not have encouragement and help, it would not have been possible to complete this project. We are grateful to few individuals, who helped us from the very start till the ends of this research work especially our supervisor who have always been there to come us out of mental block. We are grateful and thankful to our families as well for their unconditional love, prayers, and support throughout the whole life and particularly in this BE degree whose encouragement helped us to complete it.

Abbreviations and Acronyms

AI	Artificial Intelligence
API	Application Programmable Interface
AWS	Amazon Web Services
CNN	Convolutional Neural Network
CT	Computed Tomography
CXR	Chest X-rays
DB	Database
GAN	General Adversarial Network
GUI	Graphical User Interface
ML	Machine Learning
MVC	Model View Controller
PA	Posterior Anterior
RGB	Red Green Blue
TB	Tuberculosis
UI	User Interface
UX	User Experience
WHO	World Health Organization

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Medical image classification systems are winning the market and has proved to be a game-changer in the medical treatments. Big four giants (Microsoft, Facebook, Google, Amazon) invested money in the health sector to improve the outcomes and results of health care systems. Siemens the Deutsch Company has a separate department of intelligent healthcare which are working on visualizations of the human internal body and classify diseases from it.

The discovery of X-rays in 1895 by Wilhelm Roentgen led to the first Nobel Prize in Physics. Computed Tomography ranks as one of the top five medical developments in the last 40 years, according to most medical surveys. It has proven as valuable as a medical diagnostic tool that the 1979 Nobel Prize in Medicine was awarded to the inventors of CT.

A classification system is trained enough to work in the future. Scientists and engineers are making it diverse to identify every type of scenario to give the best results. But in this case, it is always an issue for a patient to believe in a machine, so classification systems propose their results. But still, they are helping doctors and radiologists to save time. Nowadays, medical imaging techniques are used to identify different diseases like brain tumors from MRI, Cardiomegaly from echocardiography, and chest diseases form chest X-rays. Kaggle, one of the biggest data science competition platforms, is providing medical imaging data and tasks to train algorithms. The application of machine learning in the field of medical imaging is a game-changer. In developed countries, hospitals invest more on intelligent smart health equipment which helps them work faster convenient and efficient they also produce better results. Besides this, all medical image classification systems help doctors to;

- identify disease faster and save time to help them work on focused areas,
- determining which surgeries are necessary,
- improve the patient placement into appropriate areas of care, such as ICU.

1.1 Purpose

The proposed system will help users by providing easy and quick overview on their chest conditions by having their CXR images. Mostly people want to consult a radiologist to get an overview of their chest conditions which are visible in CXR. This procedure takes time and money.

In our project, we are providing this facility to a user they can easily upload the x-ray image in our application from home. Our machine learning-based model identifies whether the patient is having an infection or not.

1.2 Problem statement

According to WHO reports on Pakistan, 15% are children suffering from pneumonia [1]. There are 510,000 new cases of tuberculosis reported every year [1][2]. It is a huge number of patients suffering from these diseases due to not getting better healthcare facilities [3].

Pneumonia is a disease which can cause lungs inflammation by bacteria or viral infection in which air sacs of our body filled with as shown in Figure 1.1.

Figure 1.1 CXR of a patient having pneumonia

Tuberculosis is a disease which can be caused by bacteria (*Mycobacterium tuberculosis*) it spreads from person to person in Figure 1.2 you can see the x-ray of TB patient.

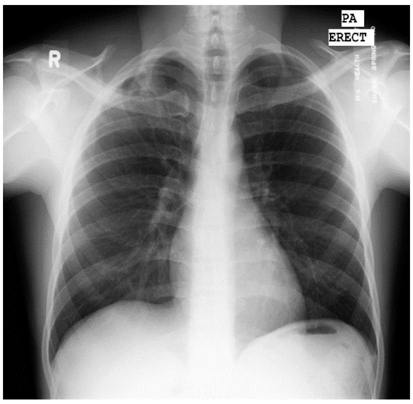

Figure 1.2 CXR of a patient having tuberculosis

These diseases can be easily detected in the CXR image of a patient, but we need a radiologist to consult for this purpose. According to the UK published a report on the most shortage of occupations the radiologists are in dire shortage hospitals having a low number of radiologists [4]. They may be in any emotion or stressed and may not identify the disease sometimes correctly.

Pakistan, being a developing country, has limited resources. Common men don't have access to good healthcare facilities due to the lack of funds. Government hospitals cannot employ a huge number of doctors they have limited vacancies. Pakistan has a large number of lower-class income families, so they have to choose government hospitals to get treated as it is affordable for them. Hence there are a large number of patients and a relatively small number of doctors, who are unable to perform their best. Then, there are chances of human error in checking the medical reports of a patient.

So, a machine that is trained to perform the same task as a human has many advantages. It will never get tired or need rest it is always unstoppable. So here this system will help doctors to work more efficiently by having all the reports generated by the system all they have to do is cross-check whether the system has classified the diseases correctly or not. So here our system is valuable and has the strength to manage tasks efficiently.

To provide people with easy and quick consultation on their chest conditions. In our project, we are providing this facility to a user that they can easily upload the x-ray image on our application from home. Our Machine learning-based model identifies whether the patient is having an infection on which our model is trained. So, it generates reports within seconds, so all doctor has to do is cross-check to check valid results our system is increasing productivity.

1.3 Objectives

Following are the main aims and objectives of our project:

- To provide people with easy and quick consultation on their chest conditions.
- To provide platform for examination of chest related diseases (Pneumonia, TB) more efferent and quickly.

1.4 Project scope

The project scope is limited to the detection of Pneumonia and Tuberculosis in the chest X-ray images.

1.5 External interface requirements

1.5.1 User interfaces

- **Html:** Web forms, main content of the website.
- **CSS:** Customize styling of website.
- **Bootstrap:** Responsive UI.
- **JavaScript:** Animation other UI based task**.**

1.5.2 Hardware interfaces

No dependency on hardware interfaces.

1.5.3 Software requirements

The software requirements of this system are as follows.

- Programming Language: Python is a programming language used in this system.
- Storage: X-rays are stored in local storage of PC.
- Framework: Framework used in the system is Flask. The web system is based on this framework.
- Interface: CSS library Bootstrap is used in the system to build a responsive website.

1.6 Significance of the system

CXR disease detector provides significance to the patient to check his/her CXR from home. It also helps the radiologist to reduce his/her burden of work. Our system is well trained on machine learning techniques which gives the result to users very quickly and generates users' reports.

1.7 Intended audience and reading suggestions

This proposed project is a prototype for the chest disease detector, and it is restricted under certain conditions**.** This project is implemented under the guidance of university professors. This project is useful for the Hospital’s Radiology Department and as well as for the patients.

1.8 User classes and characteristics

Users can upload the CXR image in the application and get generated reports from the system. He can also access the details of radiologist stored in our database for consultation. The patient user may do following functions

- Upload the CXR.
- Get the report from the system.

- Get info of radiologists.

1.9 Operating environment

Operating environment for the CXR disease detector is as follows.

- A web-browser that supports CGI, HTML & JavaScript e.g. Chrome, Firefox etc.
- Operating System (OS): Any OS supporting above mentioned class of browsers e.g. Windows, Mac, Linux (Ubuntu)
- Internet Connection

1.10 Design and implementation constraints

Frontend

- The system is Web based.
- The Web pages are created using HTML, CSS, and Bootstrap.

Backend

- The web-based system developed by using Flask Framework.
- The backend language is python.
- CNN model is created in python.
- Python Data structures are used such as lists, numpy arrays, python arrays
- Python libraries keras, opencv, sklearn are used.

1.11 Assumptions and dependencies

Following are the assumptions and dependencies of our system

- This is a web-based system. There may be traffic sometimes if it may take time to load it may be an issue with the server.
- As a human nature mostly, we do not believe in a machine so when a patient-user is uploading his CXR report to our portal. He is now emotional so to control his emotional state if his report has some serious symptoms and identified by the diseases then we make him calm and to make his doubts more clear we like to send his report automatically to the radiologist so he can consult him and make him calm and then further proceed. So, in some cases a patient is not having a report then this is the main reason.

Chapter 2. Literature Review

2.1 Introduction

Medical imaging uses the techniques and methods to generate the images of the human body for various clinical purposes such as classification of diseases or having an overview of the body. Over the last decade, the AI-based medical imaging systems have become popular in the industry. In terms of research and development, a number of grants for medical imaging research also catch the eye of researchers. Here are some startups and research groups which are working on intelligent medical imaging classification.

- Imagia [5]
- Canon Medical Research Europe [6]
- Viz [7]
- Siemens [8]

These intelligent healthcare systems were made to help save time by doctors and give detail about users about their health. It can examine the targeted users by classifying diseases on which it is trained. A medical imaging classification system is said to be a superior form of giving a personalized result.

2.2 Existing Systems

Review of some of the notable existing systems which provide proper diseases classification using medical images are as follows.

2.2.1 AI-Rad Companion

The AI-Rad Companion is AI-powered, cloud-based augmented workflow solutions, helps you to reduce the burden of basic repetitive tasks and may increase your diagnostic precision when interpreting medical images. Its solutions provide automatic post-processing of imaging datasets through our AI-powered algorithms. The automation of routine workflows with repetitive tasks and high case volumes helps you to ease your daily workflow – so that you can focus on more critical issues.

2.2.2 Chex Net

The Chex Net is an AI based model which is has capability to detect pneumonia from CXR which is equivalent to radiologist level. The team of chex-net trained and developed the model that can classify pneumonia from CXR at a level experienced radiologist. Chex Net can detect all 14 diseases from chest X-rays which can be only identified in chest X-rays and achieve state of the art results on all 14 diseases.

2.2.3 Xray4all

Xray4all is a web-based system for users to provide chest X-rays detailed results after diagnosing them. It is trained on Chex-Pert dataset provided by Stanford University. It supports multiple tasks like X-rays and histopathology slide analysis. This system analyzes uploaded images on a secure cloud backend and provides a probabilistic interpretation for different medical conditions.

2.2.4 Nora

Nora is a web-based framework for medical image analysis. It has been developed to bridge the gap between research and clinic, and to boost medical imaging research to the next level. It provides a high-level web-interface accessible from any web browser to visualize, organize, process and share data in a very customizable way. Depending on your needs, your Nora instance can run as a web-service in the cloud or as a local installation at your institution.

2.3 Classification techniques

2.3.1 Linear classifier

They are trained on a certain amount of data and after their training, they can perform classification on real-world objects. Support vector machine is a famous linear classification technique used to identify the classes on 2d-plane by drawing a line to separate the classes from each other to produce classification results.

2.3.3 Quadratic classifier

It is a statistical classifier which use quadratic decisions to produce classification results on the defined classes which our classifier is trained.

2.3.4 Decision tree

It is a tree like structure which use conditional statements to compare the nodes to decide the conditions are true or false and then provide the results on the basis of data we feed to it.

2.4 Related work

Here is some literature review of research articles related to our project.

Table 2.1 Journal review

S#	Proposed System	Results
1.	The author and his team used CNN model for classifying tuberculosis in chest X-rays. The dataset used in this is obtained from peruvian partners at "Scio's en Salud". The dataset contains 4701 images in which 453 are labeled as normal and 4258 labeled as abnormal. The final accuracy found after Alex net is about 85.68% a significant improvement from non-shuffle sampling which is 53.02% [9].	Accuracy is achieved 85% only on TB
2.	Do-un Jeoun author of this article used dataset of size 112,120 the images are transformed from 1024x1024 to 224x224 for extracting features from the author of the images used densenet-169 architecture and pass the output to support vector machines (SVM) to predict variables achieved accuracy is 80% [10].	They applied different techniques and highest accuracy achieved is 80%
3.	In this article, Wei dai used the JSRT dataset in this research and the proposed methodology is to identify the X-ray of the patient whether it is normal or not by using organ segmentation techniques and then identify the report using SCAN the main drawback of their work is they use a very small amount of data [11].	Dataset size is too small. Mostly used is synthetic data
4.	The researcher and his fellows used chest X-ray 14 dataset, containing over 100,000 frontal view X-ray images, published by the national institute of health. The proposed methodology by the researcher is that he used the Chex net algorithm; it is a state-of-the-art machine-learning algorithm to detect pneumonia at a level of a human practicing radiologist. It is a 121-layer convoluted neural network. The hex net algorithm can identify 14 pathologies from a chest X-ray. The performance of chex net reported an F1 score of 0.435. In this article, Wei dai used the JSRT dataset in this research and the proposed methodology is to identify the x-ray of the patient whether it is normal or not by using organ segmentation techniques and then identify the report using SCAN the main drawback of their work is they use a very small amount of data [12].	Testing score is quite low in this case

5.	Former apple engineer David W.Dai worked on detection of pneumonia and tuberculosis. He used JSRT and Montgomery chest set. He used very small dataset and generate synthetic data from existing dataset then trained model on it [13] [14].	The issue is in their dataset it is small and comprises mostly synthetic images.

Chapter 3. Software Requirement Specifications

This section outlines the low-level details of each system function of the pneumonia and tuberculosis classification system.

- Responsive UI and UX.
- Local storage of X-ray images.
- Report of provided X-ray.
- Radiologists recommendations.

3.1 Functional requirements

3.1.1 Responsive UI and UX

3.1.1.1 Description and priority

The user will be prompted to the home page, which is designed according to HCI 8 rules, which is easy to interact for every user.

3.1.1.2 Functional requirements

- REQ-1: The user will enter the website URL and can access the home page.
- REQ-2: To perform any prime functionality, the user must upload his chest X-ray.
- REQ-3: For that purpose, the user will click the upload button which is in the center.
- REQ-4: After uploading it the user will get results in seconds on the home page.
- REQ-5: The color scheme has to be cool and should look like a professional medical portal e.g. X-ray4All.

3.1.2 Local server storage of data

3.1.2.1 Description and priority

The user will be prompted to a home page. To perform any functionality user, have to upload the CXR.

3.1.2.2 Functional requirements

- REQ-1: User has to enter the URL of the website to open it.
- REQ-2: User will upload a chest X-ray by clicking the **upload** button.

- REQ-3: On clicking the **select** file button, user will select the X-ray image file (in PNG format) from local storage.
- REQ-4: On clicking the **upload** button, file is required to be uploaded on the server.
- REQ-5: On successful upload, the uploaded filename must be displayed on the left of the **select** button.

3.1.3 X-ray report

3.1.3.1 Description and priority

The user will get the X-ray report if he accesses the website and upload his chest X-rays on our system.

3.1.3.2 Functional requirements

- REQ-1: The trained model requires a PNG image of size of 90x90 pixels.
- REQ-2: The trained model requires the image to be in in either RGB or gray scale format.
- REQ-3: The image has to be normalized and centered.

3.1.4 Interacting with radiologists

3.1.4.1 Description and priority

The user will use this feature to get all the details of well-known radiologists. If they are not satisfied by their results.

3.1.4.2 Functional requirements

- REQ-1: There has to be some option of seeking opinion from professional radiologist on the home page (if the user is not satisfied with the generated report).
- REQ-2: User must be able to view the profiles of radiologists.

3.2 Nonfunctional requirements

3.2.1 Performance requirements

The developed system should be 24/7 available to our users. Users want to use our system must have an internet connection and reasonable internet speed.

3.2.2 Security requirements

The files uploaded to the system are fully encrypted and cannot be changed. Currently our system is working on localhost. The system will be deployed to Amazon web services (AWS) which is fully secure and trusted by users

3.2.3 Software quality attributes

3.2.3.1 Availability

The availability of the internet for our system is compulsory. Our system is internet dependent. If the internet connection is interrupted while sending or receiving information through the server, the response from our system will be a delay.

3.2.3.2 Usability

The graphical user interface (GUI) of our system will be designed in such a way to increase the usability for users. The presentation of information and choices should be clear and concise to increase usability. In short, it is easy for users to use it.

3.2.3.3 Reliability

The response time of our system should be good enough for timely response to users. The performance of our system should be good enough to increase the reliability of our system. System review will take place as per month. Any lack of performance will be addressed and improved on each review.

Chapter 4. Dataset

4.1 Dataset development

Dataset development is the first step towards training the ML algorithm. We need some data on which we need to train our algorithms to perform classification tasks. In our case, we need X-ray Images to train our algorithms because we are detecting the diseases from the X-rays. There are a few steps taken to develop the dataset.

- Data collection
- Pre-processing
- Generating labels
- Generating Pickle file

4.1.1 Data collection

It is very difficult to collect medical data because it is secure and needs lots of paperwork to be done. But there are many researchers and scientists publishing medical data for us to make use of that data by generating useful insights. So, we collected our data from different platforms related to Medics. The following are the platforms that provided us data in Table 4.1.

Table 4.1 Dataset count

Dataset Name	Total Image Count	Normal	Tuberculosis	Pneumonia
Montgomery county X-ray set [15]	138	80	58	N/A
Shenzhen china set [15]	662	326	336	N/A
Kaggle [16]	5856	1583	N/A	4273

4.1.2 Pre-processing

Some issues were faced during the pre-processing step. Some of these images were quite dark and some got unintended objects. Some problematic images are presented below.

In Figure 4.1 we have an X-ray of a female patient her backbone is tilted because of carrying child in her womb.

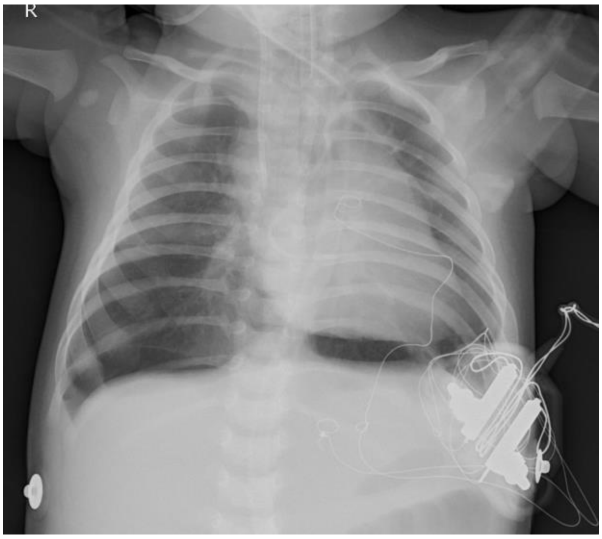

Figure 4.1 Backbone tilt in X-ray

In Figure 4.2 the clavicles are not attached to the shoulder joints. The clavicle can also be referred to as a collarbone which serves as a strut between the shoulder blade and the breastbone. The clips are used to attach them to prevent the shoulder dislocation. The 4th no rib of the patient is lying over the 5th rib it may be due to pregnancy.

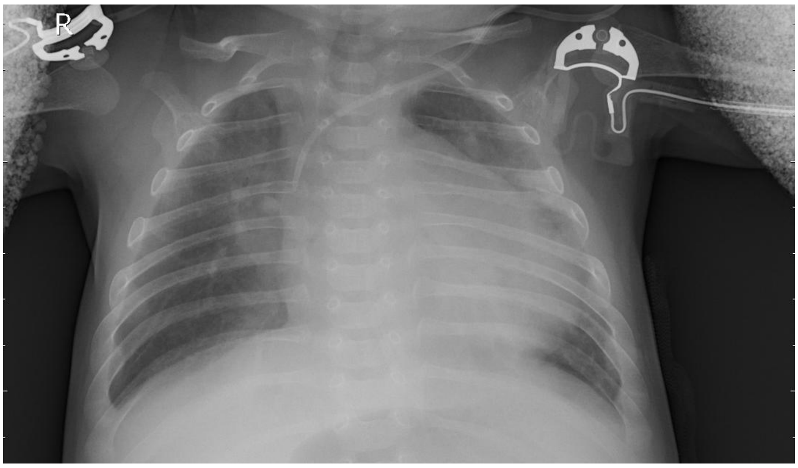

Figure 4.2 Patient having clips in shoulder joint

In Figure 4.3 we have a drainage tube shown in our X-ray. The drainage pipe is used to treat the pneumothorax. In pneumothorax, water is stuck inside our lungs and drainage pipe is used to suck the water out from the lungs.

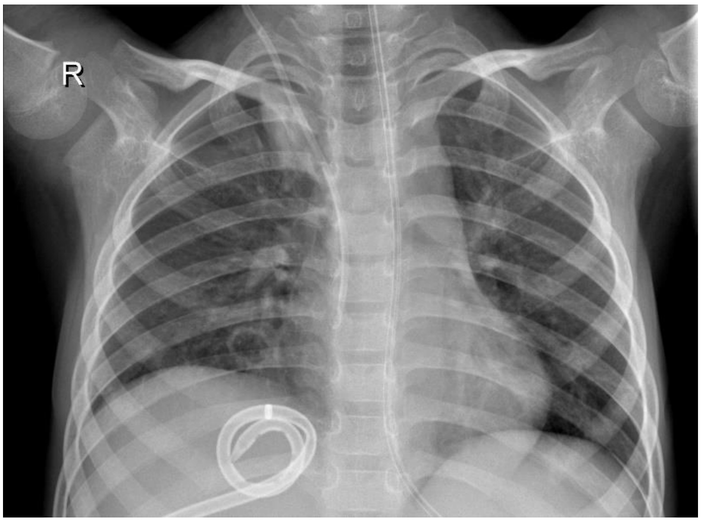

Figure 4.3 Drainage pipe inside X-ray

We cannot further pre-process it because it can remove the important factors through which disease can be identified from an X-ray, so we have to use it. There are many examples like this many patients have discs on their shoulder area. Some patients have pacemakers on their hearts etc.

For pre-processing we use cropping because it is the only technique suitable for our task. After this, we resize images to 90 X 90 and convert the dimensions of the image to 2D Grayscale. We used both grayscale and RGB images to train our algorithm both have different results. The flow of pre-processing is given in Figure 4.4.

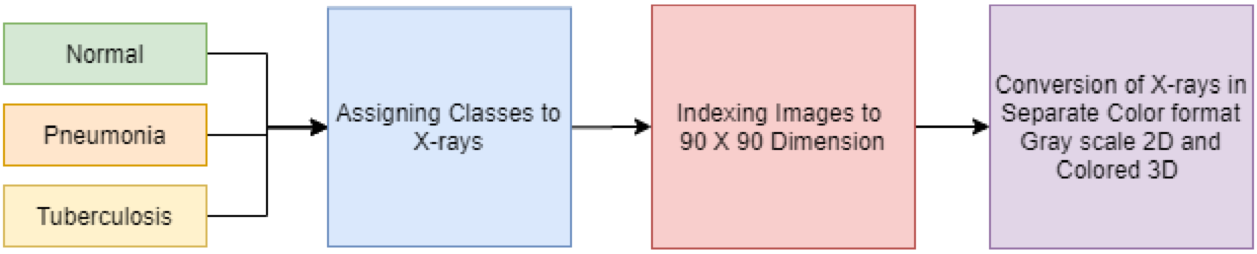

Figure 4.4 Flow of preprocessing of the dataset

4.1.3 Generating labels

After having all the X-ray images, we need to label them for our classifier to understand which image belongs to the defined class, so we have three classes normal, pneumonia, and tuberculosis and we label them as shown in Table 4.2. We used the python list in which we are adding an image with its indexed folder using python function and then convert those lists into NumPy array.

Table 4.2 Label distribution

X-ray Type	Class Assigned
Normal	0
Pneumonia	1
Tuberculosis	2

4.1.4 Dataset generation

There are many approaches to do a certain task there are two approaches which can be used to do this task.

4.1.4.1 Loading files from local storage on runtime

We can load Images in our IDE without generating a dataset pickle file, but it takes too much time and power by CPU and is not recommended.

4.1.4.2 Pickle file generation

We adopt this approach of pickle file generation. So, pickle is a file which stores all the images of our dataset and we have separate label file we use this approach because our system has not to use extra power to load all the images to the IDE in the pickle it is already done.

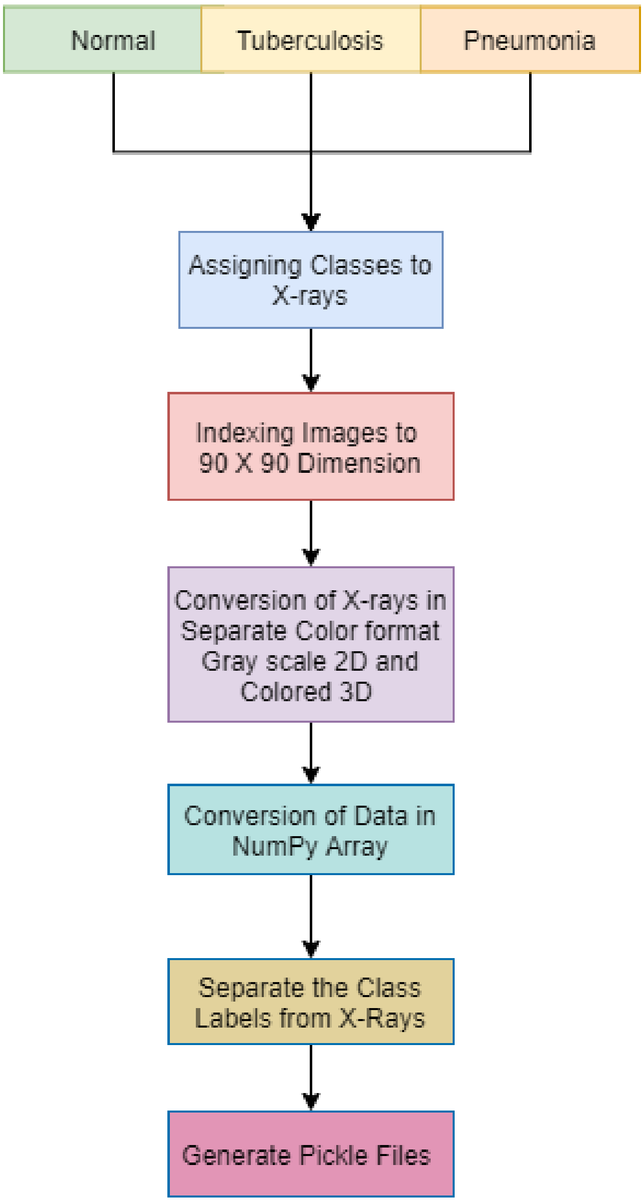

Figure 4.5 Generation of dataset

Chapter 5. Proposed Methodology

Our system is classifying the type of X-ray from the three of the following categories.

1. Normal
2. Pneumonia
3. Tuberculosis

In the next section, we discuss the popular classification techniques used in machine learning and data science.

5.1 Classification techniques

5.1.1 Linear classification

It is the most famous classification technique model used. The types of the linear classifier are given below.

1. Naive Bayes
2. Logistic regression
3. Support Vector Machines

The linear classifiers are trained on a certain amount of data and after their training, they can perform classification on real-world objects. For example, the SVM is trained to identify humans so it is trained by giving images of humans to classify humans.

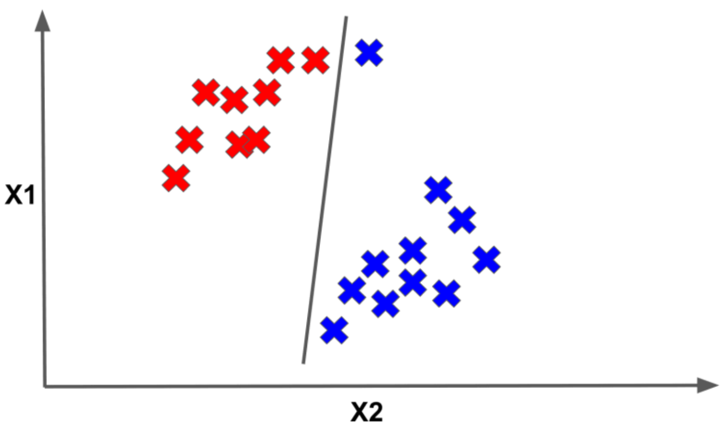

Figure 5.1 Support vector machines [17]

5.1.2 Nearest neighbor

There is a popular K-NN algorithm which is also a classification supervised algorithm. It works by finding the distances between surrounding it and if there are samples of classes in it the higher samples of that class are classified.

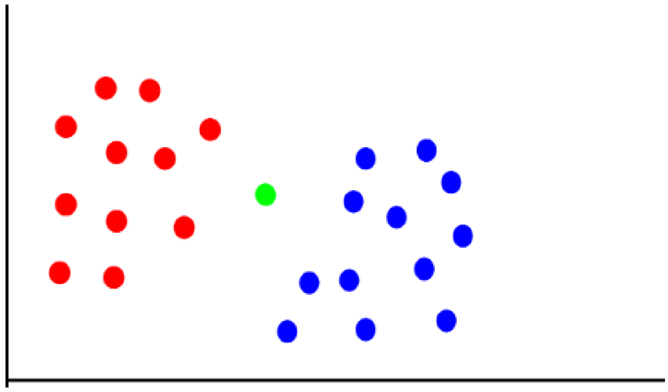

Figure 5.2 K-NN Classification

5.1.3 Neural networks

Neural networks also perform classification tasks. The neural network is a network that is inspired by the design of the human brain. Scientists design them after studying the human brain.

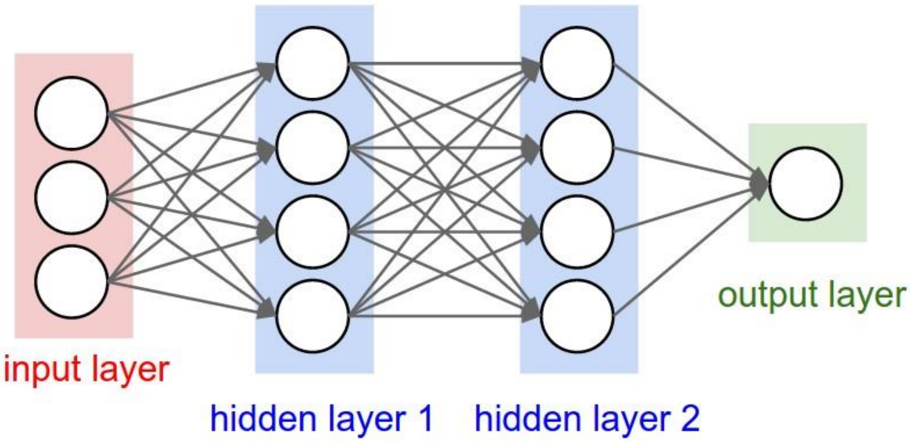

Figure 5.3 Structure of neural network [18]

Neural networks work differently from other classifiers the input is multiplied by the weights attached and then it is passed through a function to generate the output.

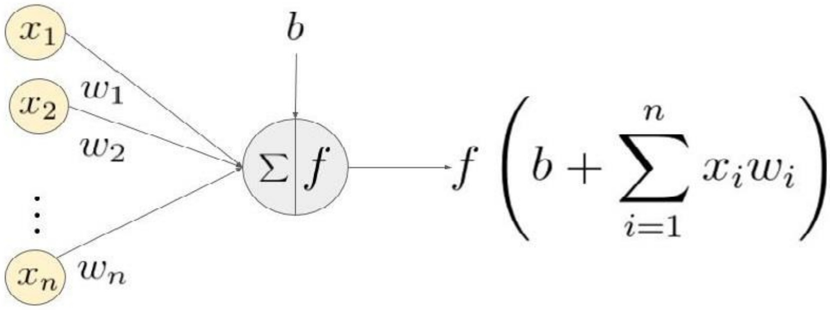

Figure 5.4 Formation of equation for neural networks [19]

5.2 Methodology implemented

5.2.1 Convolutional neural network

Convolution neural networks is one of the best approaches to classify from images. CNN is a type of Deep neural networks mostly used to analyze visual imagery. For example, CNN takes an image to classify it whether it is a dog or a cat. So, in case we have 2 classes Dog and cat with pictures representing each class we have to resize them all with fixed dimensions like (90 X 90 X 3) where 90 is representing height and width and 3 is a dimension which means it is 3d image. In CNN different layers are connected the input image is first of all pass through the convolution layer. In the convolution layer, there is a set of filters applied on the image which decide the portion of the image to use then that portion with activation functions and then passed through the pooling layer. In the pooling layer, there are further filters to extract data. So, there is a choice to define the architecture by yourself by adding further layers and at the end the flatten layer is added which is passed through the last convolution layer having several classes with softmax activation function.

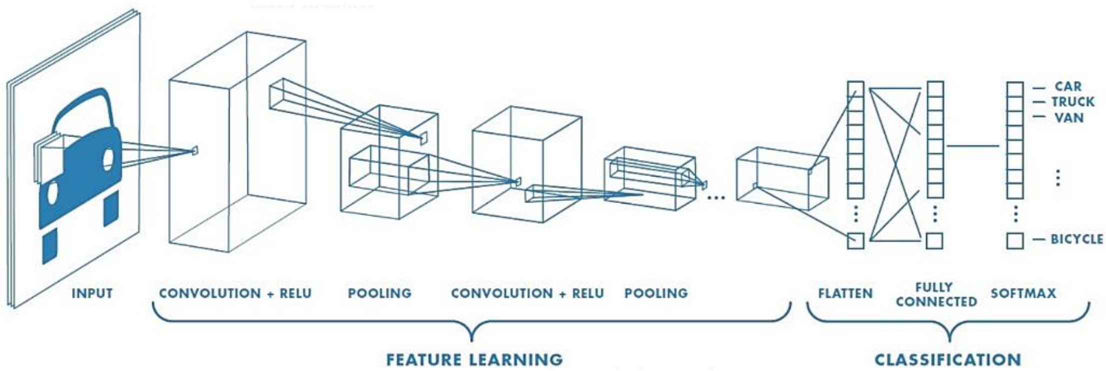

Figure 5.5 General CNN architecture [20]

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 86, 86, 128)       9728
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 83, 83, 128)       262272
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 20, 20, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 17, 17, 128)       262272
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 14, 14, 128)       262272
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
batch_normalization_1 (Batch (None, 1152)              4608
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 3459
=================================================================
```

Figure 5.6 Architecture of the implemented CNN

5.2.2 VGG-16

Vgg-16 is a transfer learning algorithm. It is a convolution neural network which is differ in architecture having 16 convolution layers. The Vgg-16 is created in image net competition because of having its best results it got popular in the deep learning. Now here is the difference. The CNN uses back propagation technique to maintain the weights to achieve good accuracy, but it uses predefined Image Net Weights.

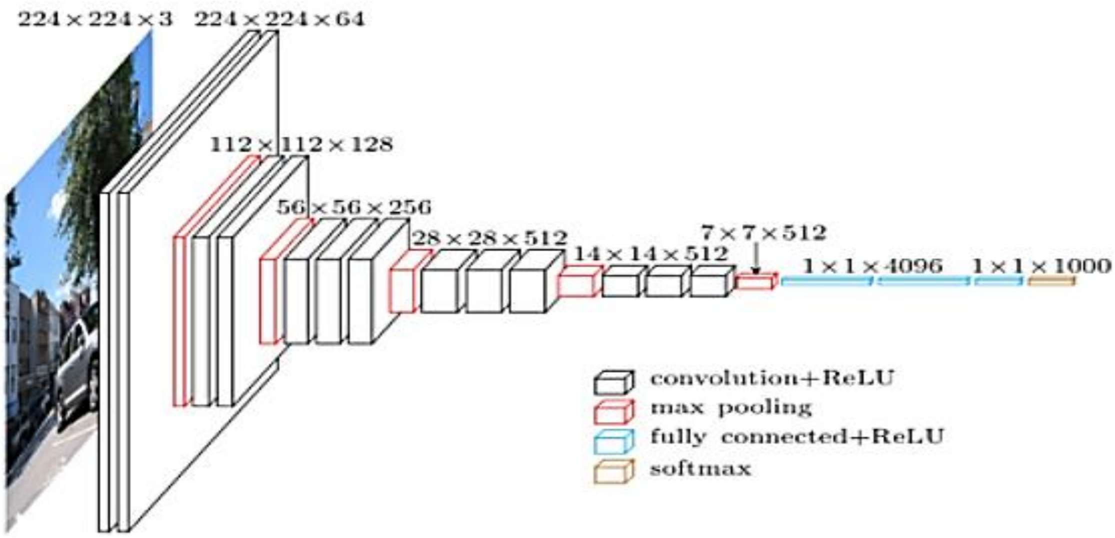

Figure 5.7 Vgg-16 architecture [21]

5.2.3 Inception network

Inception layers or inception network is the state of art deep learning architecture. They are used to allow researchers to perform efficient computation and deeper networks through a dimensionality reduction. It solves many problems like overfitting. The working of inception layer is different it has 27 layers like a neural network but in this network the convolution is performed on input using 3 types of convolution 1x1, 3x3, 5x5. Also, max-pooling is performed and then outputs are concatenated and sent to next layer.

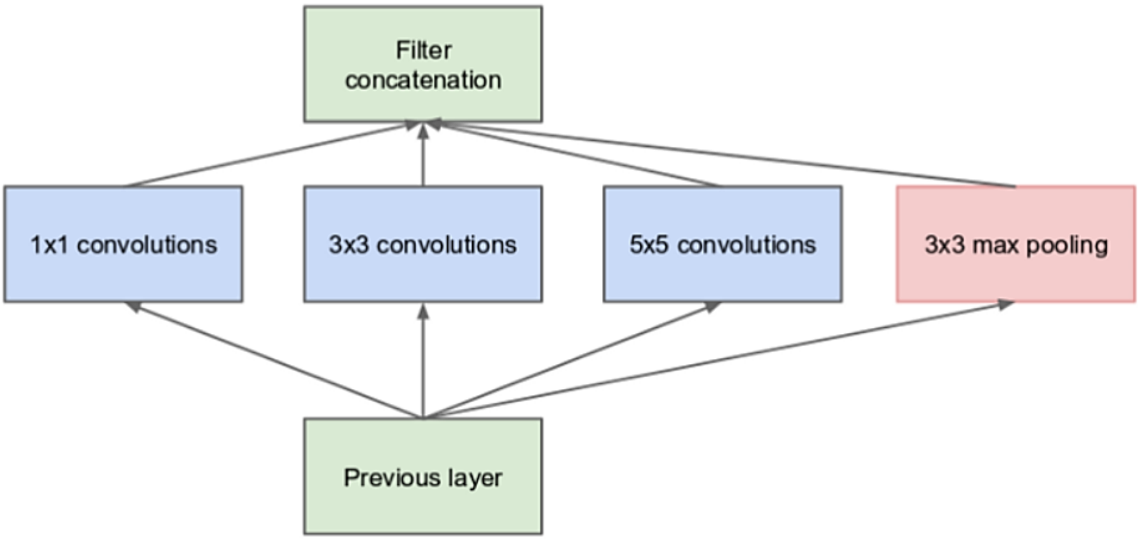

Figure 5.8 Inception-net architecture [22]

Chapter 6. System Design and Architecture

6.1 Description of the system

User visit on our website to get details related to his chest X-ray. When users access our website, he is redirected to the home page. On the home page in the center, he can easily find the prime function of the system. He can easily click on chose file to upload the X-ray in our system. Our system will classify it if the person has pneumonia or tuberculosis whether it is normal. Then details are given to the user.

6.1.1 System design

System design is the specification for business requirements identified in the system analysis. It gives the overall plan of the system consisting of all specifications that give the system its form and structure.

6.1.2 Use case diagrams

A use case diagram is a graphical representation of relations among the elements of a system. The use case is a methodology used to clarify and organize system requirements. The Figure 6.1 given below is the use case diagram of our system. In this diagram we describe the actors of our system who are going to use the system which are patient users and system itself is acting as an actor to perform some prime functionalities to get the results. We have six prime functionalities which can be performed to get the results of an X-ray of the patient. First of all, user is accessing the main page of our system. Then user can upload the X-ray image to the system and then submit it. Then our system performs functionalities by storing the X-ray in our local storage system and then resizing and dimension reduction of the X-ray is done. After performing these tasks, the image is feed to our neural network which provide us the results of our X-ray image.

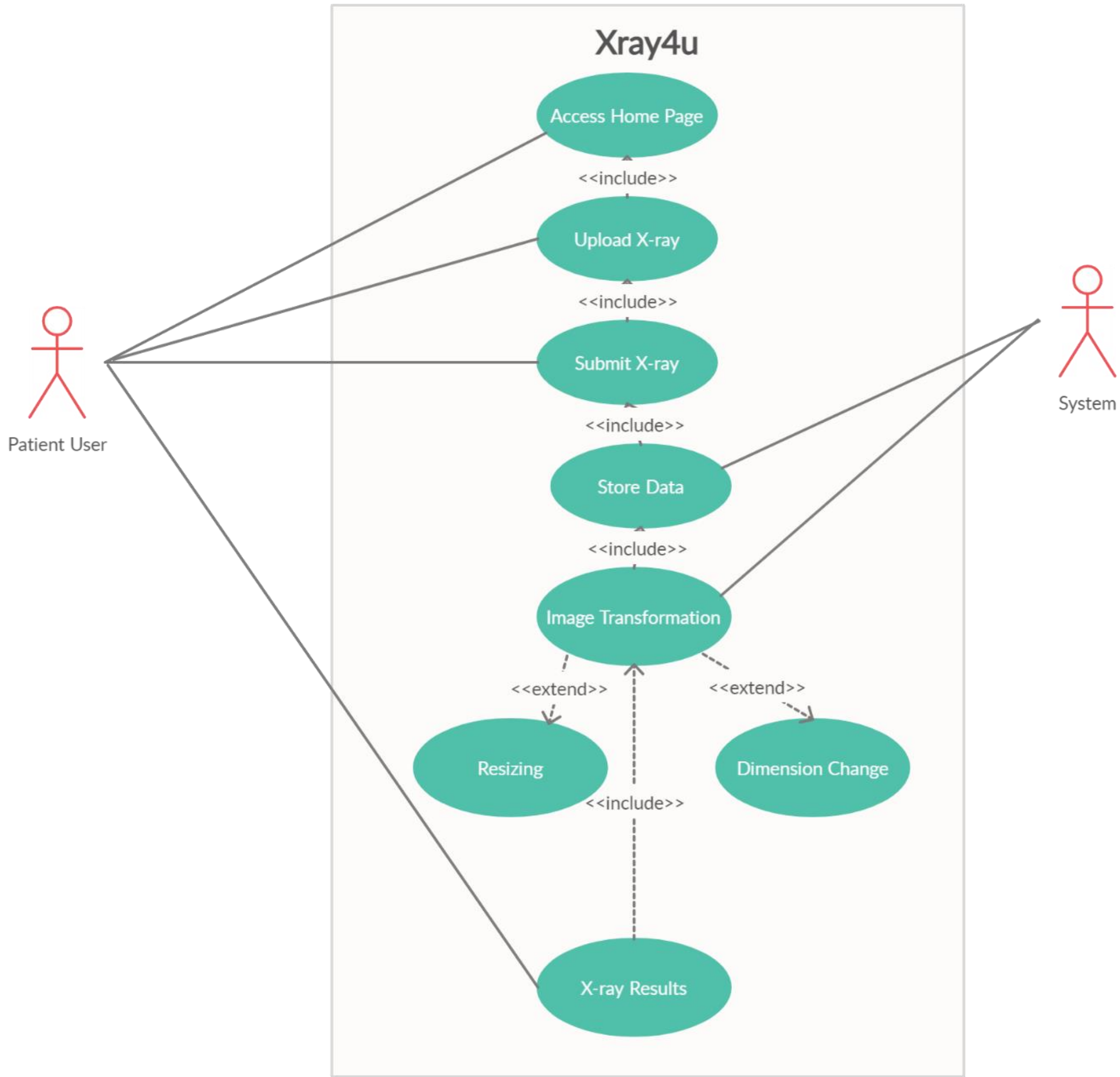

Figure 6.1 Use-Case diagram of system

6.1.3 Activity diagram

The activity diagram is another important diagram in the UML diagram to describe different aspects of the system. The activity diagram is essentially an advanced version of the flow chart that modeling the flow from one activity to another activity of the system. In Figure 6.2 below we have the activity diagram of our system which explains the working of every activity of our system.

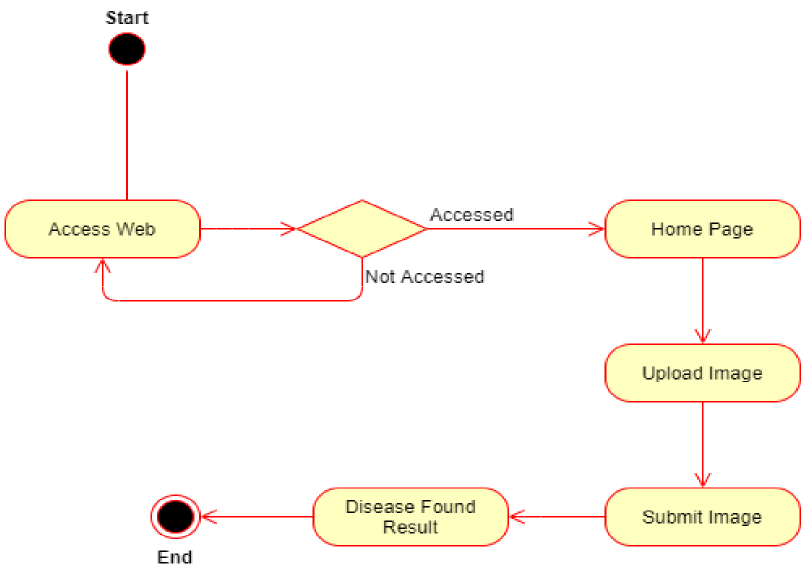

Figure 6.2 Activity diagram of system

6.1.4 Sequence diagram

Sequence diagram show the sequence of activity that how a function or activity will be performed. Sequence diagram is another important UML diagram which models high level interactions between different activities of system. In Figure 6.3 we shown the sequence diagram of our system focusing on the functional requirements of system. As you can see the Figure 6.3 it shows the user is accessing the website and is redirected to the home page of system. User can further upload the CXR Image and then submit it to system then system perform further functions at the backend and then generate the report for the user.

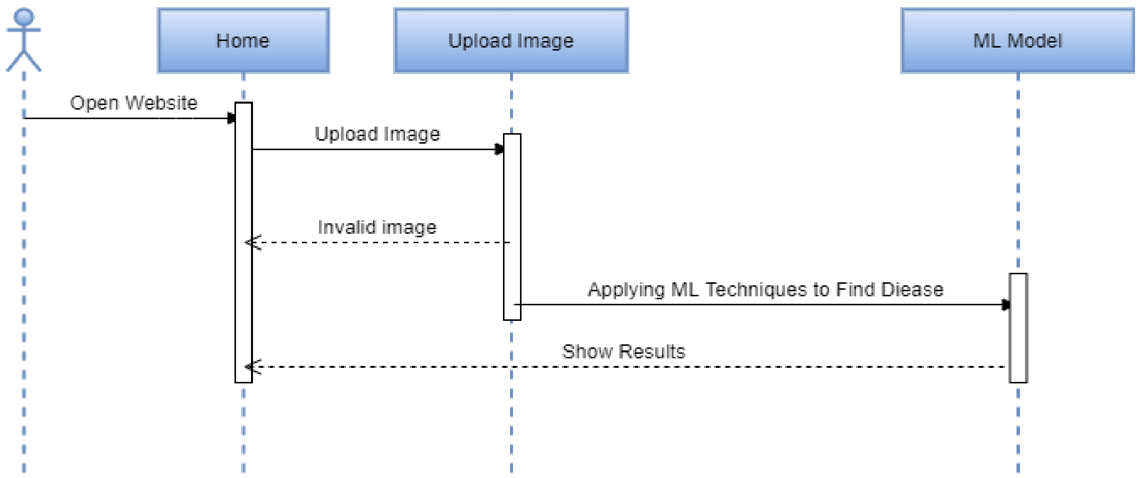

Figure 6.3 Sequence diagram of system

In Figure 6.4 we have the sequence diagram of our system's secondary functions. It is a precaution center. In precaution center we provide the precaution to prevent other and yourself from pneumonia and tuberculosis.

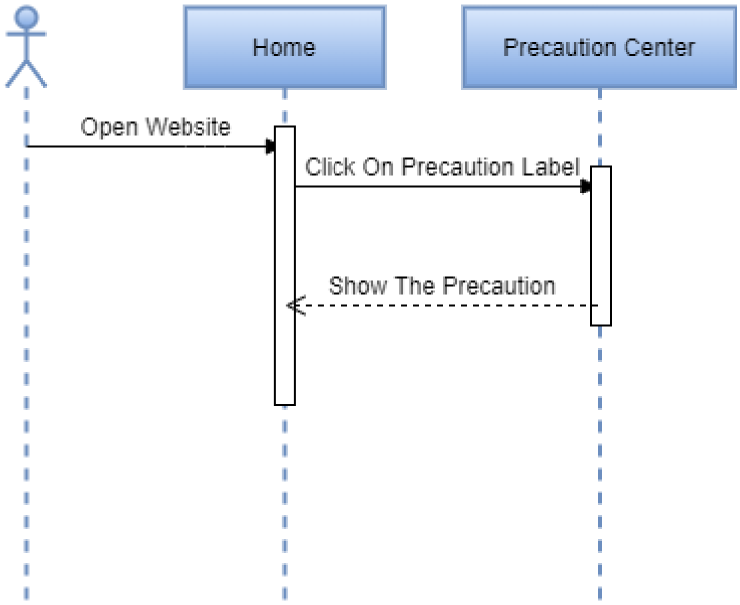

Figure 6.4 Sequence diagram of precaution center

In Figure 6.5 we have the sequence diagram of our system's secondary functions. It is a about modal. In about we discussed about the project its limitation and applicability.

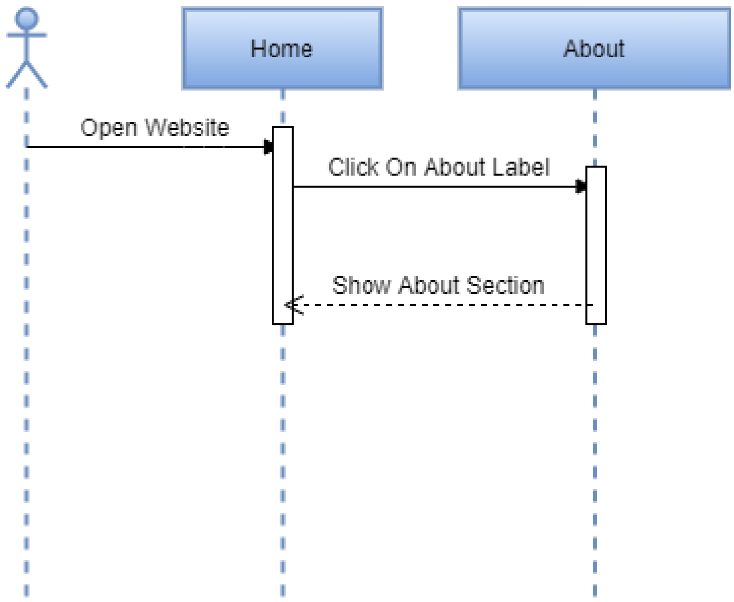

Figure 6.5 Sequence diagram of about Modal

In Figure 6.6 we have the sequence diagram of our system's prime function. It is a radiologist corner. In radiologist corner we provide the database of best radiologists providing the treatment for pneumonia and tuberculosis.

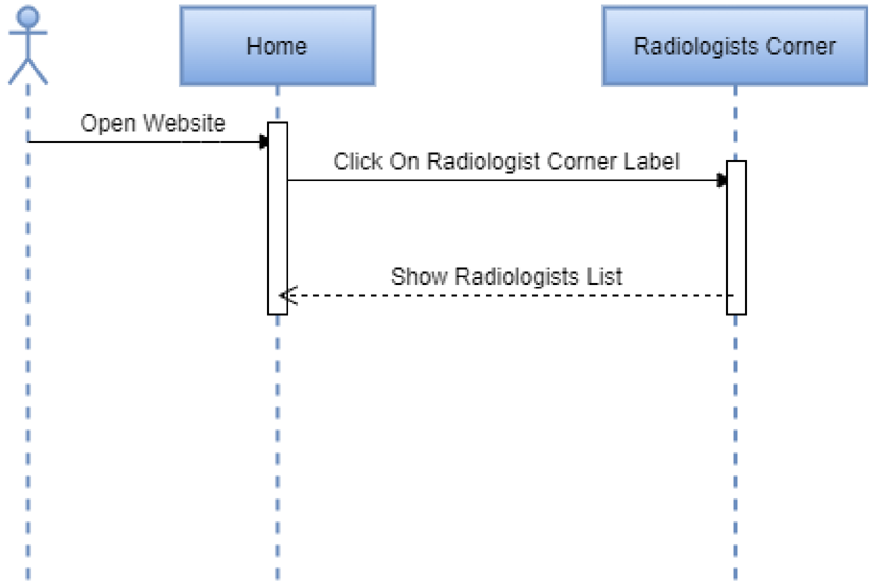

Figure 6.6 Sequence diagram of radiologist's corner

Chapter 7. Implementation

7.1 Discussion

The system implementation section defines the development, installation, testing and delivery of the proposed system. After thorough analysis and design of the system, the system implementation incorporates all other development phases to produce a functional system. Pneumonia and Tuberculosis Classification is the system developed with the latest and reliable framework and tools. The systems use all state-of-the-art libraries and updated IDE also.

In this implementation phase, we had proper meetings with our supervisor in which we discussed the work breakdown structure of the project modules and their implementation. It is divided into several steps which are as follows.

7.2 Implementation of system

7.2.1 Tools to implement classification model

Table 7.1 Tools for model

Tool	Type	Purpose
Python	Programming Language	The language which we use to train our algorithm
Colab	IDE	To implement and train the algorithm on dataset
Keras	Library	To implement CNN VGG-16 and Inception Net
Pickle	Library	To store the dataset in single file
Matplotlib	Library	To visualize the accuracies and dataset for model
SkLearn	Library	To load and splitting of dataset
OpenCV	Library	To Manipulate all the images of dataset
NumPy	Library	To store data in the form of NumPy arrays for algorithms

7.2.2 Tools to create system

Table 7.2 Tools for website

Tool	Type	Purpose
Flask	Web Framework	Integrating ML model in framework to use Model in production.
Python	Programming language	In python we are programming to deploy our model.
HTML	Markup language	To create the frontend.
CSS	Styling sheet	Used to style the web pages.
Bootstrap	Library	Used to create components of webpage.
JavaScript	Programming Language	Used to add animations in the webpage.
Keras	Library	To load our model in framework.
OpenCV	Library	Manipulate the image by changing its dimension according to the model dimensions.
JSON	Format	In this format our model is used in our framework.

7.3 System

In this section we will show the flow of system. The flow means flow chart of the system in these diagrams we show the both flows of our machine learning classifier and our system model.

7.3.1 Flow of model

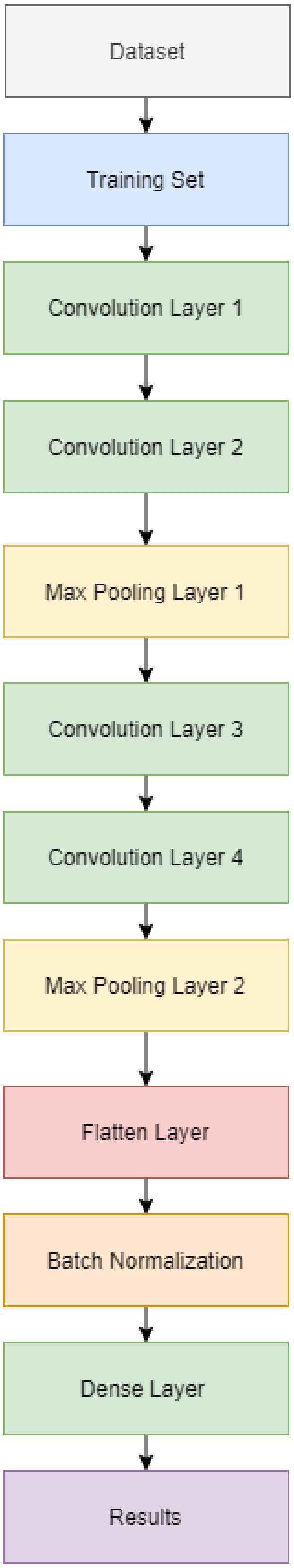

Figure 7.1 Flow of CNN

7.3.2 Flow of system

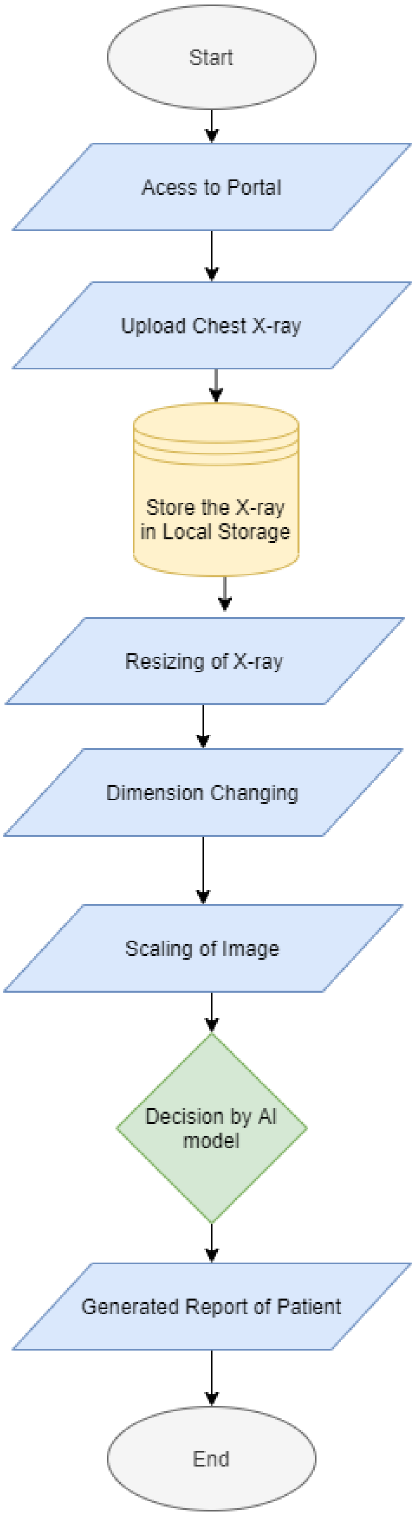

Figure 7.2 Flow of system

7.3.3 Physical system

The home page of our system is simple easy to use for users. On our home page we are providing the prime functionalities of our system. When user access the website the main page which is our home page shows the portal to choose the X-ray from your system and then upload it and get results. In Figure 7.3 we have home page when user access the website, he is directed to this page. It is the main page because we focused to develop single page app which provide all the functionalities of system.

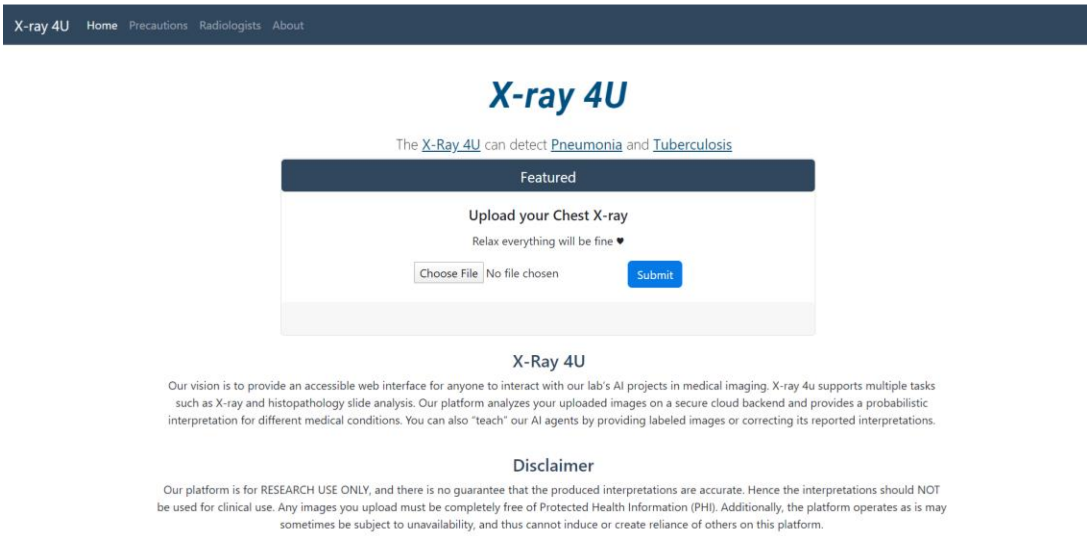

Figure 7.3 Home page screen

In Figure 7.4 we have precaution page in which the practices to prevent yourself from pneumonia and tuberculosis are mentioned.

#	Pneumonia	Tuberculosis
1	Avoid smoking and secondhand smoke, too.	good ventilation: as TB can remain suspended in the air for several hours with no ventilation
2	Get regular cardiovascular exercise to boost lung fitness and overall health.	natural light: UV light kills off TB bacteria
3	Practice deep breathing, or diaphragmatic breathing, exercises	good hygiene: covering the mouth and nose when coughing or sneezing reduces the spread of TB bacteria.
4	Get your annual flu shot and encourage others in your household to do the same.	Early diagnosis and treatment is the most effective way to prevent the spread of tuberculosis.
5	Get regular check-ups from your primary care physician and discuss any concerns about your lung health	Having a healthy immune system is the best form of defence against TB: 60% of adults with a healthy immune system can completely kill TB bacteria.

Figure 7.4 Precaution modal screen

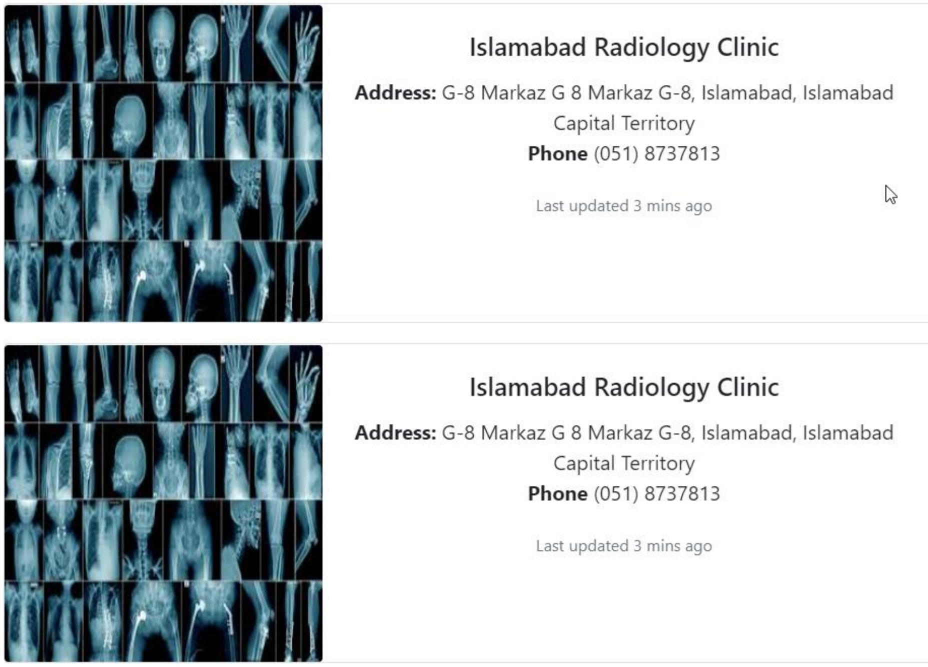

Figure 7.5 Radiologists corner screen

In the Figure 7.5 we have radiologist's corner we are showing all the well-known radiologists of our city. User can use this feature if he is not satisfied with the results of our classification system.

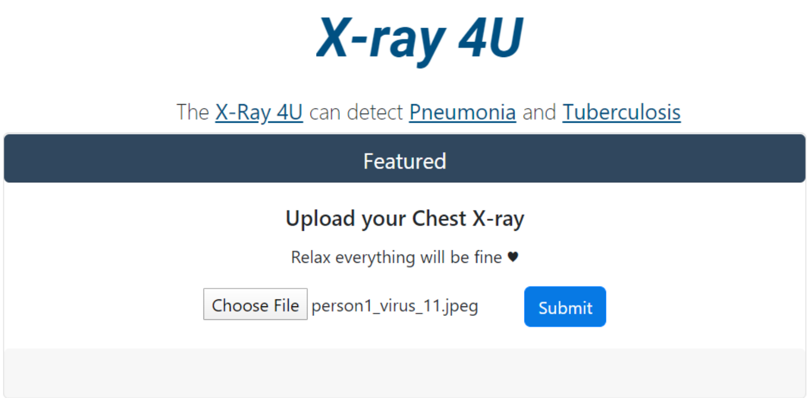

Figure 7.6 X-ray upload form

In Figure 7.6 we have the upload form after clicking the chose file button it redirects you to windows file explorer. Where you can select the X-ray after selecting the X-ray it will show the name. It means our system taken the image and show the name of image to verify.

In Figure 7.7 the report generated by our system is shown. In this report the table is given with all the possible conditions on which our model is trained. In the following report our X-ray is identified as normal.

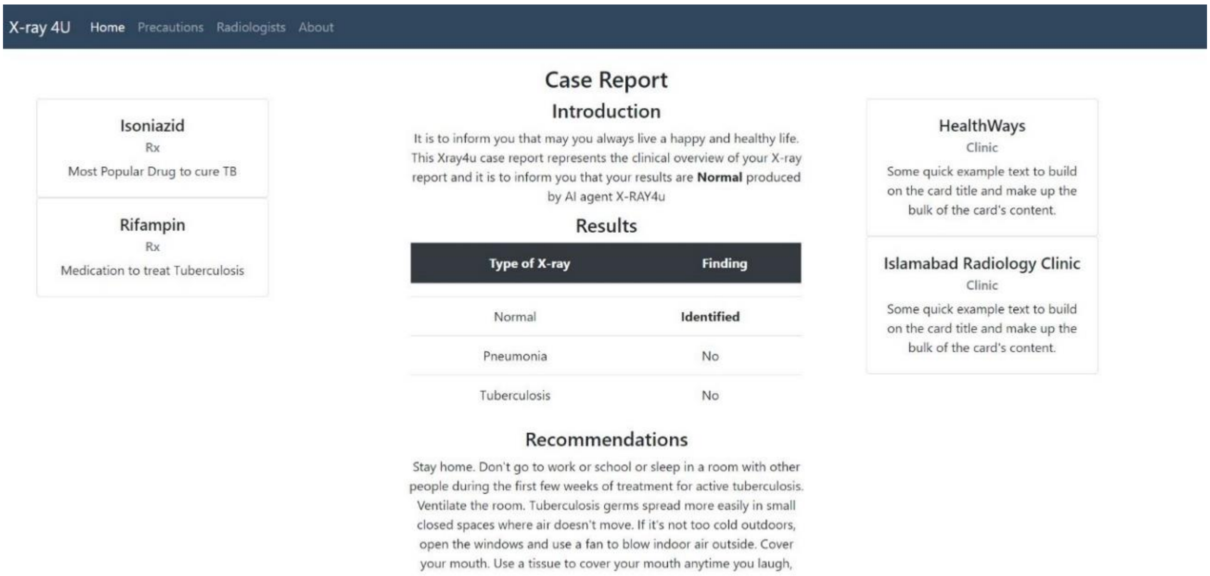

Figure 7.7 A sample generated report

In Figure 7.8 we have the about modal. In which we described the limitations of our system.

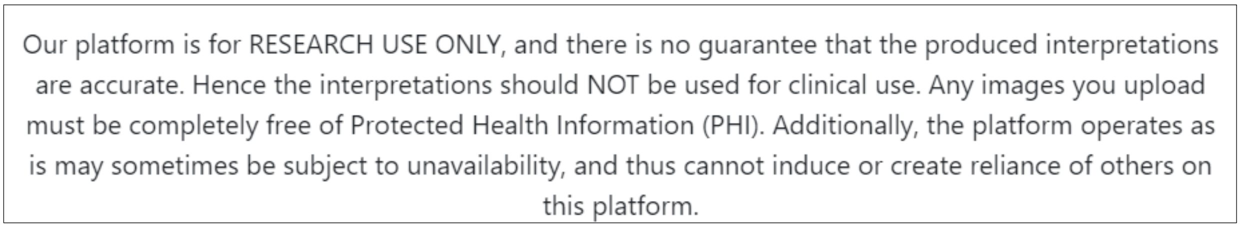
Our platform is for RESEARCH USE ONLY, and there is no guarantee that the produced interpretations are accurate. Hence the interpretations should NOT be used for clinical use. Any images you upload must be completely free of Protected Health Information (PHI). Additionally, the platform operates as is may sometimes be subject to unavailability, and thus cannot induce or create reliance of others on this platform.

Figure 7.8 About modal screen

Chapter 8. Testing

8.1 Testing techniques

For testing our system, some of the best practices of software quality assurance and testing have been adopted briefly described, few methods are listed in subsequent sections.

8.1.1 Verification

The phase of verification was done after the completion of each part of the system i.e., after login, specify an input, and specify constraints and crossover and mutation function. All the features were found to completely follow the requirements that were constructed during the project approval.

8.1.2 Validation

After the development of each module process of validation was done. To achieve the desired functionality, the modules were merged with less effort, and was integrated entirely to form the desired outcome.

8.1.3 Usability testing

After completing the business logic of the system, the usability testing was done as in the design document, and the interface was coming in shape. The results turned out to be successful implementation of the user interface of the web. The users were able to understand and recall the process efficiently and use the system with ease. At each step, it was checked and satisfied that every field and buttons are working correctly and performing complete functions.

8.1.4 Unit testing

All units (functions) were tested individually for any malicious errors and memory leaks. All the errors that were found were removed successfully, and the product's unit came into operation, the units were tested thoroughly, and the system was kept operational for hours to see if any memory leaks were identified that could crash the system.

8.1.5 Integration testing

The top-down approach was used for integration testing. Stubs were created for components. The system's outputs were verified, and they gave positive results. Each form was first tested separately then was tested after merging for proper functioning.

8.1.6 System testing

The system was integrated and was tested as a whole after the unfinished integration was tested successfully. All the errors were identified and removed successfully.

8.1.7 Acceptance testing

Acceptance testing was done with the supervisor. The test was generated successfully when the outputs were matched, and expected results were achieved from the system.

8.2 Test cases

The test cases were developed due to the end-user's interaction is most with the GUI and that must have to interact with the database.

8.2.1 Model testing

- Check if the dataset images are resized properly.
- Check if the dataset images dimensions are all equal.
- The Check dataset is generated successfully.
- Check the model after training if it is performing well.
- Check if the model is overfitting or underfitting
- Check if the model is performing successfully.

8.2.2 System testing

- Check the model is integrated into framework successfully.
- Check if the user can upload X-ray successfully.
- Check if the reports are generated successfully or not.

Table 8.1 Test Case 01 – Image resizing

Test Case # 01		**Test Case Name**: Image resizing		
System: Xray4u		**Sub-System**: Not a subsystem		
Designed by: Hassan		**Design Date**: 15-05-2020		
Executed by: Abubakar		**Execution Date**: 18-05-2020		
Short Description: Test to check images for dataset are resized on same size or not.				
Pre-conditions (if any): No pre-conditions.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Collect images from the folders from python code	Get all the images of the dataset and displayed them on command line	Pass	Self-explanatory
2	Collect all images from folder with their folder name	Get the folder name by using Indexing function	Pass	Self-explanatory
3	Resize the image	Using the OpenCV we resize the image and store it into the List data structure the dimensions are 90 X 90 for all images	Pass	Self-explanatory
4	Confirming the size	Converted the List to Numpy array and then checked its size it is according to our requirement	Pass	Self-explanatory

Table 8.2 Test Case 02 – Dimension reduction

Test Case # 02		**Test Case Name:** Dimension reduction		
System: Xray4u		**Sub-System:** Not a subsystem		
Designed by: Hassan		**Design Date:** 15-05-2020		
Executed by: Abubakar		**Execution Date:** 18-05-2020		
Short Description: Dataset dimension checker.				
Pre-conditions (if any): Computer/Laptop should have a working internet connection.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Get all the images from folders	Get all the images from folder using OS library	Pass	Self-explanatory
2	Collect all images from folder with their folder name	Get the folder name by using indexing function	Pass	Self-explanatory
3	Change dimensions	Using OpenCV we change the dimension of our dataset from 3D to 2D	Pass	Self-explanatory

Table 8.3 Test Case 03 – Generation of dataset

Test Case # 03		**Test Case Name**: Generation of dataset		
System: Xray4u		**Sub-System:** Not a subsystem		
Designed by: Hassan		**Design Date:** 15-05-2020		
Executed by: Abubakar		**Execution Date:** 18-05-2020		
Short Description: Dataset dimension checker.				
Pre-conditions (if any): Computer/Laptop should have a working internet connection.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Get the created NumPy of images and labels	Their dimensions are according to our requirements	Pass	Self-explanatory
2	Converting dataset pickle file	Importing pickle library and convert them into two separate pickle files one for labels and other for images	Pass	Self-explanatory
3	Check the dataset	Imported pickle file and checked its contents it is okay	Pass	Self-explanatory

Table 8.4 Test Case 04 – Model training

Test Case # 04		**Test Case Name**: Model training		
System: Xray4u		**Sub-System**: Learning algorithm		
Designed by: Hassan		**Design Date**: 15-05-2020		
Executed by: Abubakar		**Execution Date**: 18-05-2020		
Short Description: Test to check the Model is training or not.				
Pre-conditions (if any): Computer/Laptop should have a working internet connection. User must have a registered account User must have created the event.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Inserting raw image	Failed to train give error	Pass	Self-explanatory
2	Inserting generated dataset images in the list	Failed to load need NumPy array	Pass	Self-explanatory
3	Inserting generated dataset NumPy array	Successful and started training	Pass	Self-explanatory

Table 8.5 Test Case 05 – Model overfitting

<table>
<tr><td colspan="2">Test Case # 05</td><td colspan="3">Test Case Name: Model overfitting</td></tr>
<tr><td colspan="2">System: Xray4u</td><td colspan="3">Sub-System: Learning algorithm</td></tr>
<tr><td colspan="2">Designed by: Hassan</td><td colspan="3">Design Date: 15-05-2020</td></tr>
<tr><td colspan="2">Executed by: Abubakar</td><td colspan="3">Execution Date: 18-05-2020</td></tr>
<tr><td colspan="5">Short Description: Test to check the over fitting of model.</td></tr>
<tr><td colspan="5">Pre-conditions (if any): Computer/Laptop should have a working internet connection. User must have a registered account.</td></tr>
<tr><th>Step</th><th>Action</th><th>Expected System Response</th><th>Pass/Fail</th><th>Comments</th></tr>
<tr><td>1</td><td>Giving model only 30 images</td><td>Model overfits</td><td>Fail</td><td>Self-explanatory</td></tr>
<tr><td>2</td><td>Given dataset of 5996 images</td><td>Validation accuracy > training accuracy. In some cases, like 2 to 3 epochs only which is tolerable</td><td>Pass</td><td>Self-explanatory</td></tr>
<tr><td>3</td><td>Using batch normalization</td><td>Over fitting ended good results achieved</td><td>Pass</td><td>Self-explanatory</td></tr>
</table>

Table 8.6 Test Case 06 – Model execution

Test Case # 06		**Test Case Name:** Model execution		
System: Xray4u		**Sub-System:** Learning algorithm		
Designed by: Hassan		**Design Date:** 15-05-2020		
Executed by: Abubakar		**Execution Date:** 18-05-2020		
Short Description: Test to check the trained model is working or not.				
Pre-conditions (if any): Computer/Laptop should have a working internet connection.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Save model and load It on new cell	Loaded on new cell of Colab notebook	Pass	Self-explanatory
2	Load a dummy image	Loaded	Pass	Self-explanatory
3	Resize and change the dimensions of dummy image	System changed its dimensions and resize the image	Pass	Self-explanatory
4	Pass the image to model	Detected the disease	Pass	Self-explanatory
5	Pass the image without resize and dimensions reduction	Cannot read image	Failed	Self-explanatory

Table 8.7 Test Case 07 – Model integration

Test Case # 07		**Test Case Name:** Model integration		
System: Xray4u		**Sub-System:** Learning algorithm		
Designed by: Hassan		**Design Date:** 15-05-2020		
Executed by: Abubakar		**Execution Date:** 18-05-2020		
Short Description: Test to check the model is integrated successfully				
Pre-conditions (if any): Computer/Laptop should have a working internet connection.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Save the model using Keras save function	Stored to the desired Directory	Pass	Self-explanatory
2	Open the Workingdir.py file of flask and import Keras	Successfully imported	Pass	Self-explanatory
3	Give the path of model and load it in the flask	Successfully imported and working	Pass	Self-explanatory

Table 8.8 Test Case 08 – System execution

Test Case # 08		**Test Case Name:** System execution		
System: Xray4u		**Sub-System:** Not a Subsystem		
Designed by: Hassan		**Design Date:** 15-05-2020		
Executed by: Abubakar		**Execution Date:** 18-05-2020		
Short Description: Test to check the website is running successfully.				
Pre-conditions (if any): Computer/Laptop should have a working internet connection.				
Step	**Action**	**Expected System Response**	**Pass/Fail**	**Comments**
1	Run command Python Workingdir.py	Localhost IP is provided	Pass	Self-explanatory
2	Click on the upload image button	File explorer is opened and selection of desired image	Pass	Self-explanatory
3	Click on the submit button	Give results on uploaded X-ray	Pass	Self-explanatory

Chapter 9. Analysis and Results

9.1 Algorithms

The algorithms we use to train our system are given below.

- Convolution neural network
- Inception network
- VGG-16 architecture neural network

9.2 Analysis and results

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 86, 86, 128)       9728
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 83, 83, 128)       262272
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 20, 20, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 17, 17, 128)       262272
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 14, 14, 128)       262272
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 3, 3, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
batch_normalization_1 (Batch (None, 1152)              4608
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 3459
=================================================================
```

Figure 9.1 CNN Architecture

In Figure 9.1 we have presented the architecture of our primary algorithm with the best results achieved so far. This is the architecture of neural network we used to achieve the best results on our data. Architecture is simple it consists of two convlolutional layers with max-pooling layers and again the convolution layer with max-pooling layers then we have a flatten layer to merge the data into one layer then we use batch normalization layer and at the end, we have a dense layer with softmax function.

In Figure 9.2 we have the training and validation graph of our neural network. This graph is showing us how the algorithm is fitting on the data accurately. The training accuracy is rising in every epoch but validation accuracy results are not good as compare to the training ones this shows data is not overfitting because when training accuracy is lesser than validation it means our model is overfitting but in some epochs it is surpassing the training accuracy it doesn't mean it is overfitting it means the data in that epoch which our model is validated have easy examples in it that is why it is performing well on them.

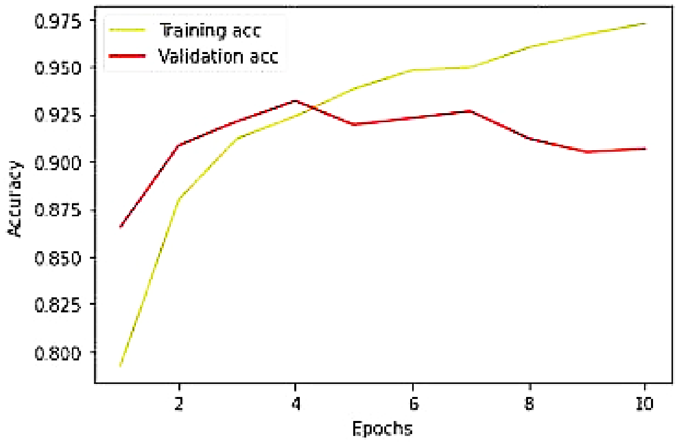

Figure 9.2 Training and validation graph accuracy

In Figure 9.3 we have training and validation loss graph of our neural network. This graph is showing us the loss in each epoch our model is having, and it shows our model is having loss decreased in every epoch and results are getting better and better.

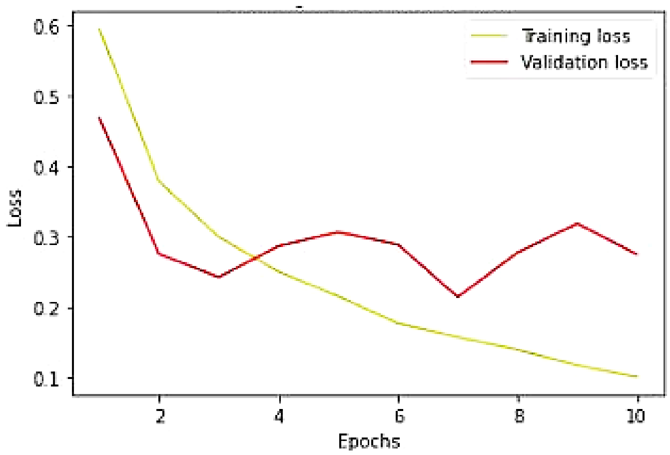

Figure 9.3 Training and validation loss graph

In Figure 9.4 we are using keras evaluator function which is used to evaluate our model. By using this method, we will know the performance of our model it evaluates the model on our testing set and then gives the accuracy of the model based on how many examples it classifies correctly so we have **92.97%** accuracy by keras evaluator.

```
1394/1394 [==============================] - 1s 392us/step
Evaluation 1
accuracy: 92.97%
```

Figure 9.4 Evaluation accuracy

```
Train on 5018 samples, validate on 558 samples
Epoch 1/10
5018/5018 [==============================] - 14s 3ms/step - loss: 0.5943 - accuracy: 0.7927 - val_loss: 0.4681 - val_accuracy: 0.8656
Epoch 2/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.3782 - accuracy: 0.8800 - val_loss: 0.2753 - val_accuracy: 0.9086
Epoch 3/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.2994 - accuracy: 0.9121 - val_loss: 0.2425 - val_accuracy: 0.9211
Epoch 4/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.2503 - accuracy: 0.9239 - val_loss: 0.2866 - val_accuracy: 0.9319
Epoch 5/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.2154 - accuracy: 0.9382 - val_loss: 0.3063 - val_accuracy: 0.9194
Epoch 6/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.1771 - accuracy: 0.9480 - val_loss: 0.2887 - val_accuracy: 0.9229
Epoch 7/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.1576 - accuracy: 0.9496 - val_loss: 0.2143 - val_accuracy: 0.9265
Epoch 8/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.1396 - accuracy: 0.9599 - val_loss: 0.2774 - val_accuracy: 0.9122
Epoch 9/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.1179 - accuracy: 0.9669 - val_loss: 0.3185 - val_accuracy: 0.9050
Epoch 10/10
5018/5018 [==============================] - 6s 1ms/step - loss: 0.1008 - accuracy: 0.9727 - val_loss: 0.2748 - val_accuracy: 0.9068
```

Figure 9.5 Training and validation accuracy of CNN

In Figure 9.5 we have training and validation results of our model also their losses are given which tells us how much better our training done. In Figure 9.5 we have terms used which are explained in Table 9.1

Table 9.1 Terms used in training of the neural network

Term	Purpose
Loss	It is used as an indicator to improve the performance of the model. If data is or not. If the loss is increasing it means the model is not learning weights should be changed and if it is decreasing epoch by epoch, then it is performing well.
Accuracy	It tells us how well our model is training on the data we feed to it. If the accuracy is not changing remains the same, it means our model is not learning. If accuracy is decreasing it means it is underfitting.
Val_loss	It stands for validation loss. It is the same as the training loss mentioned above. The validation loss is for validation set which is split from our training at the time of training.
Val_accuracy	It stands for Validation accuracy. Its purpose is the same as training accuracy, but it is the accuracy of our validation set. It is mainly used to indicate if our model is trained well then it should perform close enough to our training accuracy if it is not performing well or sometimes performing better than training then it means it may have overfitting.

```
Train on 5018 samples, validate on 558 samples
Epoch 1/50
5018/5018 [==============================] - 1508s 301ms/step - loss: 0.5583 - accuracy: 0.7237 - val_loss: 0.4992 - val_accuracy: 0.7515
Epoch 2/50
5018/5018 [==============================] - 1504s 300ms/step - loss: 0.4876 - accuracy: 0.7716 - val_loss: 0.4989 - val_accuracy: 0.7515
Epoch 3/50
5018/5018 [==============================] - 1501s 299ms/step - loss: 0.4874 - accuracy: 0.7716 - val_loss: 0.5055 - val_accuracy: 0.7515
Epoch 4/50
5018/5018 [==============================] - 1503s 300ms/step - loss: 0.4879 - accuracy: 0.7716 - val_loss: 0.5016 - val_accuracy: 0.7515
Epoch 5/50
5018/5018 [==============================] - 1512s 301ms/step - loss: 0.4868 - accuracy: 0.7716 - val_loss: 0.4994 - val_accuracy: 0.7515
Epoch 6/50
5018/5018 [==============================] - 1513s 302ms/step - loss: 0.4888 - accuracy: 0.7716 - val_loss: 0.4988 - val_accuracy: 0.7515
Epoch 7/50
5018/5018 [==============================] - 1512s 301ms/step - loss: 0.4888 - accuracy: 0.7716 - val_loss: 0.4996 - val_accuracy: 0.7515
Epoch 8/50
5018/5018 [==============================] - 1512s 301ms/step - loss: 0.4874 - accuracy: 0.7716 - val_loss: 0.5011 - val_accuracy: 0.7515
Epoch 9/50
5018/5018 [==============================] - 1510s 301ms/step - loss: 0.4873 - accuracy: 0.7716 - val_loss: 0.4997 - val_accuracy: 0.7515
```

Figure 9.6 Training and validation accuracy of VGG-16

In Figure 9.6 we have the results of our VGG-16 model it shows no learning all of the epochs have the same accuracy with the same loss, so it states our model is not learning at all. The terms in Figure 9.6 are described in Table 9.1.

```
Train on 4259 samples, validate on 1065 samples
Epoch 1/25
4259/4259 [==============================] - 29s 7ms/step - loss: 3.7346 - acc: 0.7637 - val_loss: 3.7830 - val_acc: 0.7640
Epoch 2/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.7187 - acc: 0.7677 - val_loss: 3.4520 - val_acc: 0.7847
Epoch 3/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.4611 - acc: 0.7835 - val_loss: 3.9451 - val_acc: 0.7534
Epoch 4/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.2772 - acc: 0.7953 - val_loss: 3.3476 - val_acc: 0.7909
Epoch 5/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.6935 - acc: 0.7694 - val_loss: 3.8834 - val_acc: 0.7577
Epoch 6/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.7864 - acc: 0.7638 - val_loss: 3.8734 - val_acc: 0.7584
Epoch 7/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.7864 - acc: 0.7638 - val_loss: 3.8733 - val_acc: 0.7584
Epoch 8/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.7864 - acc: 0.7638 - val_loss: 3.8734 - val_acc: 0.7584
Epoch 9/25
4259/4259 [==============================] - 27s 6ms/step - loss: 3.7864 - acc: 0.7638 - val_loss: 3.8733 - val_acc: 0.7584
Epoch 10/25
```

Figure 9.7 Training and validation accuracy of inception layer

In Figure 9.7 we have results of the inception layer which is not learning anything as you can see the training accuracy results of model it is the same not changed a bit. It also has high loss which is greater than the VGG-16. The terms used in Figure 9.7 are described in Table 9.1.

9.3 Accuracy comparison

Table 9.2 states the accuracy of all the algorithms obtained by keras evaluation method. We used convolution neural network, inception net, and vgg-16 without image net weights. The algorithms are trained on different image count as the count of X-ray images increases the accuracy of our model increases. The factor which played an important role to improve the accuracy is the quality of image before using the RGB (3d) image we trained our model on grayscale (2d) images the results are not good as compare to the RGB images

Table 9.2 Accuracy Comparison

Algorithms	Dataset count	Type of X-ray Images	Results
CNN	800	Grayscale	53%
Inception network	800	Grayscale	49%
CNN	6656	Grayscale	87%
CNN	6656	Colored	92.97%
VGG-16	6656	Colored	75%
Inception network	6656	Colored	78%

Chapter 10. Conclusions and Future Work

10.1 Conclusion and future work

Our project is giving 92.97% results. The system performed well on the detection of normal and pneumonia X-rays. System produced fair results on tuberculosis images due to a shortage of data of tuberculosis. So, system is correct to some extent people should use it to get overview of their chest diseases. In the future, our work is still dedicated to this system by maintaining the X-rays of patients and make it more accurate.

10.2 Detection of coronavirus from chest X-rays

Coronavirus is the severe stage of pneumonia and it can also be captured in chest X-rays. So, during the pandemic health professionals and researchers opensource the data of COVID positive patients which can be used to generate realistic and helpful insights. The data consists of chest X-ray and personal information of the patient. We used the data pre-processed and picked the posterior anterior X-rays which are in the low count and trained on our neural network but due to very low amount of data, we cannot train it perfectly also the facts and figures of patients varying from country to country.

10.3 Medical imaging class imbalance

In addition to this, we are also working on a medical imaging library for python. The main purpose of the library is to produce synthetic data for medical imaging imbalanced datasets. We are using different trained models to identify the medical image when it is passed to our library and then use GAN's to produce synthetic data to help researchers producing good results while training their machine learning models.

References

[1] WHO, 2020. Pneumonia. [online] WHO. [Accessed 28 July 2020].

[2] WHO EMRO, 2020. Tuberculosis. Stop Tuberculosis. [online] WHO, p.1 [Accessed 29 July 2020].

[3] S. Ali, "The poor state of Pakistan’s healthcare system", DAWN.COM, 2016. [Online]. [Accessed: 29- Jul- 2020].

[4] The Royal College of Radiologists, 2019. Clinical Radiology UK Workforce Census 2018 Report. [online] London: The Royal College of Radiologists, p.5. [Accessed 29 July 2020].

[5] "Home - Imagia", Imagia, 2020. [Online]. [Accessed: 28- Jul- 2020].

[6] "Home - Canon Medical Research USA, Inc. (CMRU)", Canon Medical Research USA, Inc., 2020. [Online]. [Accessed: 28- Jul- 2020].

[7] "Viz.ai, Inc.", Viz.ai, 2020. [Online]. [Accessed: 28- Jul- 2020].

[8] "Medical Imaging", Siemens-healthineers.com, 2020. [Online]. [Accessed: 28- Jul- 2020].

[9] C. Liu, M. Alcantara and B. Liu, "TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network", IEEE, 2017. [Accessed 31 July 2020].

[10] D. Jeoung, O. Stephen and M. Sain, "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare", Journal of Healthcare Engineering, vol. 2019, 2019. [Accessed 2 August 2020].

[11] W. Dai, H. Zhang and J. Doyle, "SCAN: Structure Correcting Adversarial Network for Organ Segmentation in Chest X-rays", Arxiv, 2017. [Accessed 2 August 2020].

[12] P. Rajpurkar, J. Irvin and K. Zhu, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning", Arxiv. [Accessed 31 July 2020].

[13] D. Dai, " https://towardsdatascience.com/can-machine-learning-read-chest-x-rays-like-radiologists-part-1-7182cf4b87ff", Can AI Read Chest X-rays like Radiologists? part 1, 2019.

[14] D. Dai, "https://towardsdatascience.com/can-machine-learning-read-chest-x-rays-like-radiologists-part-2-aa77dba219f0", Can AI Read Chest X-rays like Radiologists? Part 2, 2019.

[15] A. SK, "Tuberculosis Chest X-ray Image Data Sets | National Library of Medicine", Lhncbc.nlm.nih.gov, 2020. [Online]. [Accessed: 29- Jul- 2020].

[16] P. Mooney, "Chest X-Ray Images (Pneumonia)", Kaggle.com, 2018. [Online]. [Accessed: 29- Jul- 2020].

[17] M. Eric, "https://www.mihaileric.com/posts/support-vector-machines/", Basics of Support Vector Machines, 2020.

[18] A. Rosebrock, "https://www.pyimagesearch.com/2016/09/26/a-simple-neural-network-with-python-and-keras/", A simple Neural Network, 2016.

[19] T. Babs, "https://medium.com/coinmonks/the-mathematics-of-neural-network-60a112dd3e05", The Mathematics of Neural Networks, 2018.

[20] S. Saha, "https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53", A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, 2018.

[21] M. Ul hassan, "https://neurohive.io/en/popular-networks/vgg16/", VGG16 – Convolutional Network for Classification and Detection, 2020.

[22] Deep.ai "https://deepai.org/machine-learning-glossary-and-terms/inception-module", What is an Inception Module?, 2020.