Haseeb Tariq
01-134172-019
Muhammad Hassan
01-1341-086

COVID-19 Diagnosis Based on CT- Scan of Lungs

Bachelor of Science in Computer Science

Supervisor: Ma'am Maryam Bibi

Department of Computer Science
Bahria University, Karachi

May 2021

Abstract

Novel Virus COVID-19 is a new kind of virus from coronavirus family which causes sickness from common cold to Middle East Respiratory Syndrome and Severe Acute Respiratory Syndrome. Ever since the Novel Virus COVID-19 began back in December 2019 it has cause massive destruction across world. As of May 1, 2021, it has claimed 3.17 million lives. One of the major strategies against this virus is to conduct tests at larger scale. Testing not only helps in treatment of infected patients but also helps states to formulate policies to prevent its spread. CT-scan base diagnosis is more preferred than PCR testing as it has higher sensitivity as compared to PCR testing. Furthermore, it also helps to measure Morbidity of disease. Only problem with CT-scan base testing is that it requires medical professional to generate report which could take hours. The objective of this project is to build web application which will computerize the process for CT-scan based diagnosis of Novel Virus COVID-19 . A Convolutional neural network was trained and deployed at server side which takes CT-Scan image from client side and classify them into Control (negative), Regular and Severe class.

During model Training phase in addition to original dataset we also composed a patient wise dataset and train well know architectures such AlexNet, VGG16 and ResNet50 on this dataset. We further modified and tune these architectures to extract best out of them and to improve their accuracies. Out of all architectures we tried modified form of Resnet50 with increased dropout rate and batch normalization in dense layer gave us the highest accuracy. This model showed accuracy of 99%,80% and 90% on training, validation, and testing sets, respectively.

Acknowledgments

In the Name of Allah, the Most Beneficent, the Most Merciful. All praise belongs to Allah. It is due to his blessing that we were able to complete this project successfully. We would like to express our deepest thanks and gratitude towards Ma'am Maryam Bibi who gave us chance to work under her supervision. She helped us and guided us throughout the project. She provided us many helping materials and helped us in trouble shooting problem that occurred during this project. No doubt without her guidance we won't be able to complete this project successfully. We would also like to thank Mr. Yu Xue who helped us compose patient base dataset at end we would also like to thank our families for supporting us.

HASEEB TARIQ, MUHAMMAD HASSAN
Bahria University Islamabad, Pakistan

May 2021

Contents

Abstract **i**

Acknowledgments **ii**

Acronyms and Abbreviations **vi**

1 Introduction **1**
1.1 Project Background/Overview 1
1.2 Problem Description 2
1.3 Project Objectives 2
1.4 Project Scope 3

2 Literature Review **4**
2.1 Hybrid CNN Based Architectures 4
2.2 ResNet50 Based Architectures 5
2.3 CNN+LSTM Based Architectures 6
2.4 Simple Convolution Based Architectures 8

3 Requirement Specifications **11**
3.1 Existing System 11
3.2 Proposed System 11
3.3 Requirement Specifications 12
3.3.1 Functional Requirements 12
3.3.2 Non-Functional Requirements 13
3.4 Use Cases 13
3.4.1 USE CASE 1: Enter Credentials 14
3.4.2 USE CASE 2: Upload Image 15
3.4.3 USE CASE 3: Start Test 15
3.4.4 USE CASE 4: Receive report/ download report 16
3.4.5 USE CASE 5: View Stats 17

4 Design **18**
4.1 System Architecture 18
4.2 Design Constraints 19
4.3 Design Methodology 19
4.4 High Level Design 20
4.4.1 Component Diagram 20
4.4.2 System Interaction Diagram 21
4.4.3 Class Diagram 23

4.5 GUI Design . . . 24

5 System Implementation **26**
5.1 System Architecture . . . 26
5.1.1 Algorithmic Workflow . . . 26
5.2 System Components . . . 27
5.2.1 Implement client server architecture . . . 27
5.2.2 Dataset . . . 28
5.2.3 Image pre-processing . . . 31
5.2.4 Image classification and Training Deep learning model . . . 32
5.2.5 Database . . . 34
5.3 Tools and Technologies . . . 34

6 System Testing and Evaluation **35**
6.1 Graphical User Interface Testing . . . 35
6.2 Usability Testing . . . 35
6.3 Software Performance Testing . . . 35
6.4 Exception Handling . . . 36
6.5 Load Testing . . . 36
6.6 Test Cases . . . 36
6.6.1 Launching website test case . . . 36
6.6.2 Sending data to server side . . . 37
6.6.3 Image Classification . . . 37
6.6.4 Displaying Stats Graph . . . 37
6.7 Analysis and discussion of results . . . 38
6.7.1 Results on original dataset . . . 38
6.7.2 Results on patient-base dataset . . . 39
6.7.3 Comparison . . . 45

7 Conclusions **46**
7.1 Future Work . . . 46

A User Manual **48**
A.1 Start Web Application . . . 48
A.2 Enter Details . . . 49
A.3 upload CT-scan image . . . 50
A.4 Error on fields . . . 51
A.5 Processing State . . . 52
A.6 Receiving Report . . . 53
A.7 Download Report . . . 54
A.8 Opening Stats Page . . . 54
A.9 Database Operation Error . . . 55

References **58**

List of Figures

2.1 The hybrid-learning architecture of HUST-19 [1]. 5
2.2 Framework of DRE-Net [2]. 6
2.3 framework of hybrid network [3]. 7
2.4 CTnet-10 architecture [4]. 8
2.5 CTnet-10 architecture [5]. 9

3.1 Use Case Diagram 14
3.2 Use case 1: Enter Credentials 14
3.3 Use Case 2: Upload Image 15
3.4 Use Case 3: Start Test 16
3.5 Use Case 4: Receive report/ download report 16
3.6 Use Case 5: View stats 17

4.1 System Architecture 19
4.2 Methodology 20
4.3 Component Diagram 21
4.4 System Interaction Diagram 1 22
4.5 System Interaction Diagram 2 23
4.6 Class Diagram 23
4.7 Welcome/Splash Screen 24
4.8 Home Screen 25

5.1 Algorithm Workflow 27
5.2 This figure shows sample images from Control class. Here lungs are at their healthy state and there at no sign of ground class opacities or paving marks. 29
5.3 This figure shows sample images from Regular class. These images show lungs of patient suffering from mild COVID-19 virus and there are some sign of ground class opacities or paving marks. 30
5.4 This figure shows sample images from Severe class. These images show lungs of patient suffering from Severe COVID-19 virus and there are intense sign of ground class opacities or paving marks. 31

6.1 Simple AlexNet 40
6.2 AlexNet with increased dropout rate 40
6.3 AlexNet with increased dropout rate and l2 regularization 41
6.4 Simple VGG 42
6.5 VGG with increased dropout rate 42
6.6 VGG with increased dropout rate and l2 regularization 43
6.7 simplest ResNet50 44

6.8 Resnet with increased dropout rate 44
6.9 Resent with increased dropout rate and batch normalization 45

A.1 Start Application. 49
A.2 Entering patient details. 50
A.3 uploading CT-scan image. 51
A.4 Validation error on fields. 52
A.5 Processing animation. 53
A.6 Diagnosis report page. 53
A.7 Downloaded report. 54
A.8 Stats page. 55
A.9 Database operation failure error. 56

List of Tables

2.1 Summary of literature 10

3.1 Enter Credentials 15
3.2 Select video from gallery 15
3.3 Start Test 16
3.4 Receive report/ download report 17
3.5 View Stats 17

6.1 Table for test case 1 36
6.2 Table for test case 2 37
6.3 Table for test case 3 37
6.4 Table for test case 4 38
6.5 Table of accuracies for seen faces of MIRACL-VC1 39
6.6 AlexNet Results 39
6.7 AlexNet Results 41
6.8 AlexNet Results 43

Acronyms and Abbreviations

CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Project Background/Overview

Coronaviruses is a class of viruses which causes sickness ranging from common cold to Middle East Respiratory Syndrome and Severe Acute Respiratory Syndrome. According to Study conducted by Ozsahin, Ilker, et al [6], In 2019 a corona virus was identified in Wuhan, China it was named as "Novel Virus COVID-19" by World health organization. within few months this virus spread across the world and WHO declared it as pandemic. As of 1st May 2021, 3.17 million casualties have been reported. This virus enters through nose or mouth. From there it moves into the air sacs inside lungs knows as alveoli. After it reaches the alveoli, the virus uses it unique spike structure proteins to takeover other healthy cells. Like all other viruses Novel Virus COVID-19 makes multiple copies of itself. Once the RNA of Novel Virus COVID-19 enters into the structure of healthy cell it kills original cell by making multiple copies, which release new viruses that in turn effects healthy neighboring cells in the alveolus.

There are two main common approaches for testing Novel Virus COVID-19. Most commonly used one is a laboratory-based approach, in which a sample is obtained through a nasopharyngeal swap by inserting a 6-inch-long swab into the cavity between the nose and mouth (nasopharyngeal swab) for 15 seconds after that this swap is exposed to a paper containing artificial antibodies which provide a visual result. This test provide sensitivity of 86% and specificity of 96%. Second approach is using medical imaging diagnostic tools such as X-ray and computed tomography. CT scan provides a very detailed images of lungs and CT has a high sensitivity for diagnosis of Covid-19 furthermore CT scan test is better in terms of availability and people are also more comfortable to CT scan test as compare to PCR test.

According to study carry out by Carotti, Marina, et al [7], CT scans of Novel Virus COVID-19 patients in early stages show bilateral process with ground glass opacities. These ground glass opacities are noticed to be rounded in shape.Ground glass opacity means an opacity that does not obscure underlying vessels i.e., we can see vessels through opacity. In later ages CT-scan of

Novel Virus COVID-19 patient shows paving marks. These features can be used to identified Novel Virus COVID-19 patients, but this diagnosis requires expert radiologist. Fortunately, modern days machine algorithms have become very accurate in image classification. we are intended to develop a convolution neural network model for classification of CT-scan image in Control (negative), Regular and Severe class.

In current era machine learning is involved in almost every filed of life form business to medical science but the real breakthrough really happed in past couple of decades with an increase in data and computational power many deep learning algorithms came into existence. These algorithms are very effective and accurate in pattern recognition and feature extraction. Deep learning is being widely used in field of medical science specially for medical imaging diagnosis. Deep learning algorithms are proved to be very effective in diagnosis of different kind of tumors and cancer. We are intended to use convolutional neural network which is not only computational efficient than traditional deep learning algorithm but also best suited for recognition of patterns like ground glass opacities and paving marks in lungs for diagnosis of Novel Virus COVID-19.

1.2 Problem Description

Although CT-scan base diagnosis of Novel Virus COVID-19 is very effective, it provides additional information regarding morbidity of disease, but it has a major downside. This technique requires a medical radiologist to compose report based on CT-scan imaging provided by machine. Radiologist carefully analyses Ct-scan image and tries to look for features like ground glass opacities and paving marks. Based on these finding he compose a report stating whether patient has Novel virus Covid-19 or not. This process could take hours. During this time patient remains in state of uncertainty and could even infect other people furthermore during this pandemic era doctors and paramedical staff are already over burden hence availability of radiologist is also an issue. Hence An automatic method is required which reduce this delay and automatically generate report. A web application with CNN classifier could be built for this purpose which would classify give image according to its features.

1.3 Project Objectives

Objective of this project is to develop a web application for the diagnosis of Novel Virus COVID-19. This application will overcome the delay that a patient could face in process of getting diagnosis report. CT-scan image received from client side will be classified into control, Regular, or severe class. End Result will be presented to user on web portal and will be stored into database. Information stored into database will be used to show user different stats.

1.4 Project Scope

This project aims to deliver a web application which will perform diagnosis of Novel Virus COVID-19 based on imaging of CT scan machine. This web application will not preform CT-scan imaging on its own and hence requires CT-scan image as input. Image received from client Are classified into Control(negative), Regular and Severe class and report is showed on web portal. In addition to this, application will also show stats based on information collected and stored in database previously. This application will only work on specific format of CT scan imaging of lungs, and it will not work on other type of medical imaging like x-rays etc. CT scan image should be clear and distortion free.

Chapter 2

Literature Review

Medical imaging has been using for decades in diagnosis of different tumors and cancers. Novel Virus OCIVD-19 on other hand is relatively a new problem as it was discovered by scientist at the end of 2019. Hence mostly work done on diagnosis of Novel Virus COVID-19 is in recent years of 2020 and 2021. Features like ground glass opacities and paving marks can be used for diagnosis of COVID-19 virus. Generally, both machine learning and non-machine learning techniques are used when it comes to medical image diagnosis. But in Novel Virus COVID-19 Mostly existent work is machine learning based. Deep learning has become very prominent in field of medical images analysis in past couple of years. Convolutional neural network approach is ideal in problems where we must recognize patterns such as in current case where we want to classify images in COVID-19 positive and COVID-19 negative class based on patterns in lungs CT scan. Some of the most More recent work done in pattern recognition of lungs CT scan are discussed below.

2.1 Hybrid CNN Based Architectures

Hybrid CNN takes heterogenous data as inputs and combine their results to achieve overall high accuracy. Following paper used a hybrid CNN model for diagnosis of Novel Virus COVID-19. Ning, Wanshan, et al [1] proposed a hybrid model for classification of lung's CT scan images. This hybrid network integrates two heterogenous dataset first dataset is composed of chest computed tomography (CT) imaging whereas second dataset is based on clinical features (CFs) of patients. data is collected from 1170 anonymous patient including 649 laboratory-confirmed, 222 Novel Virus COVID-19 negative/control and 299 suspected patients. After slicing image dataset had three classes NiCT, pCT and nCT with 5705, 4001 and 9979 images respectively. Overall model architecture is shown in figure 2.1.

In preprocessing stage lung parenchyma portion is extracted from CT scan image author also resized image to 200×200 pixels. To avoid distortion author used bilinear interpolation and scikit-image. In HUST-19 At very first stage network classify CT scan images into NiCT , pCT and

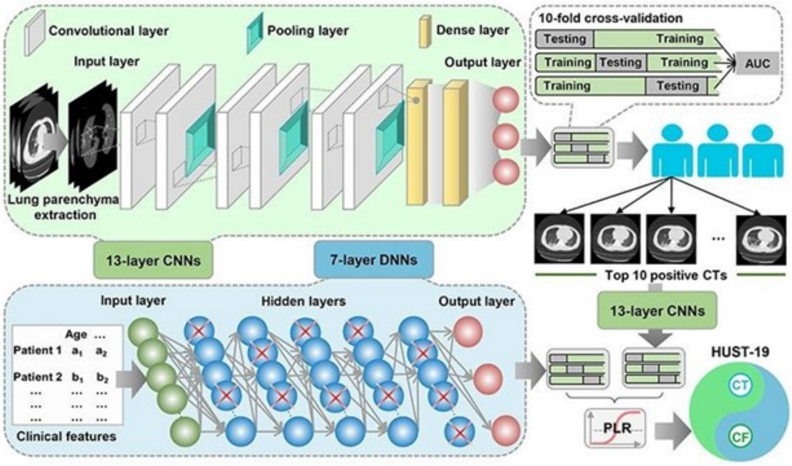

Figure 2.1: The hybrid-learning architecture of HUST-19 [1].

nCT for this purpose a deep learning framework of 13 -layer convolutional neural network was use that contain three sets of convolution layer and (3x3) pooling layer followed by two fully connected layers and at last an output layer. In Second stage another 13-layer convolutional network is used to transform CT scan image base prediction into patient base prediction. This CNN classify patients into control type 1 or type 2. In third stage CF based classification is performed by seven fully connected layers to classify patients into three types. This seven-layer DNN contain one input one output and 5 hidden layers. At end penalized logistic regression algorithm is used to integrate predictions of made on CT and CF data. Combination of CT and CF prediction allowed author to achieve and over all AUC value of 0.978, 0.921 and 0.931 in predicting control, Type 1, and Type 2 patients respectively.

2.2 ResNet50 Based Architectures

Adding more layer cause problem of vanishing gradient in CNN architecture. Resnet prepose solution to this problem and introduced concept of Residual Block in CNN. Basically, it skips layers to pass information to deeper layers. In network with residual block every layer feed information to layer next to it and to layers which are 2-3 hops away. This technique allows Resnet50 to have higher number of layers. Overall, 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. Following are papers which are based on ResNet50.

Rahimzadeh, Mohammad et al [8] proposed a methodology based on ResNet50V2 network with modified feature pyramid network alongside author's designed architecture for classifying the CT scan images. Author introduced a new data set called COVID-CT set which is gathered from Negin radiology located at Sari Iran. This dataset is composed of 15589 images that belong to 95 patients infected with Novel Virus COVID-19 and 48260 images of 282 healthy people. In preprocessing author preformed slicing of CT scan in which he extracted those images from consecutive CT scan images in which lungs portion of human body is properly shown for this author develop a new algorithm that select Those images which have fewer dark pixels.

Author used deep learning convolutional neural network to classify the selected images by author's designed algorithm. Author proposed model uses ResNet50V2 as the backbone in which FPN (Feature pyramid network) is like original FPN but unlike original FPN author used concatenation layers instead of adding layers inside the feature pyramid network. This FPN extract five final feature that each one presents the input feature on different scale. After this author used dropout layers followed by first classification layer. At end of this layer author used five classified layers each with two neuron and made ten neurons dense layer this layer was connected to final classification layer on which SoftMax is applied an overview of this architecture is as shown below in figure below. Overall, in single image evaluation phase the average result of five folds indicates author's proposed model has an accuracy of 98.449% and sensitivity of 94.96% for NOVEL VIRUS COVID-19 class.

Song, Ying, et al. [8] developed a deep learning-based diagnosis system called Deep Pneumonia. Dataset prepared by Author contains CT scan of 88 Novel Virus COVID-19 patients form two province of china, 101 CT scan of patient of other pneumonia and CT scan of 86 healthy peoples. In preprocessing stage author sliced each 3D CT scan image into 15 slices after that lungs portion is extracted by using OpenCV functions. Overall, 777 ,505 ,708 sliced images were taken for positive class, other pneumonia class and healthy class, respectively [8].

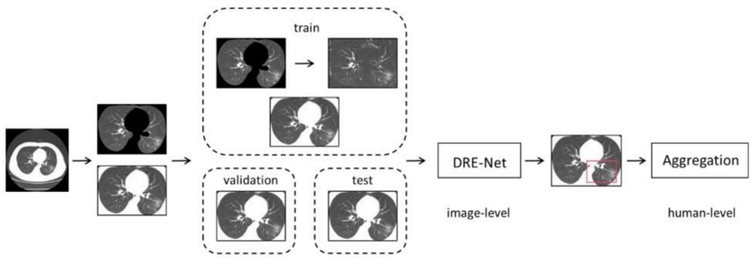

Figure 2.2: Framework of DRE-Net [2].

Author proposed model DRE-Net is built on pre trained ResNet50. Architecture is shown in figure **??** A feature pyramid network was added in order to extract to relevant features from images. In this way model was able to detect important regions of image and predict overall output. Images results were aggregated for individual patient for which mean pooling was used. Proposed model on testing data achieved AUC of 0.92 in the image level, and AUC of 0.95 and recall of 0.96 in the human level [8].

2.3 CNN+LSTM Based Architectures

In CNN LSTM architecture CNN is used for feature extraction of input data and LSTM is used for interpreting the features. In this network CNN model is placed at top followed by LSTM model and at end Dense layers are placed. Following paper is based on CNN+LSTM architecture.

Islam, Md Zabirul et al [3] proposed deep learning technique based on the combination of convolutional neural network and short-term memory to diagnosis NOVEL VIRUS COVID-19 from x ray images. In this system CNN will be used for feature extraction whereas LSTM will used for detection using extracted features. Dataset used by author contains 4575 images with three classes which are (1) NOVEL VIRUS COVID-19 with 1525 images (2) Normal with 1525 images and (3) Pneumonia with 1525 images. Preprocessing was applied on raw images of collected dataset which include resizing of data to 224x224, data shuffling and data normalization. After the preprocessing stage data was split into training and testing set[5].

Author's proposed model is basically a combination of convolutional neural network and long short-term memory (LSTM) network. CNN is used to extract complex features from images whereas LSTM is used as classifier. the network contains 20 layers overall in which there are 12 convolutional layers , fiver pooling layers one FC layer , one LSTM layer and finally one output layer in last with softmax activation function. Each convolution black have 2-3 CNNs and a pooling layer after that dropout layer is used. filter of 3x3 were used for convolutional layers and kernels of 2x2 were used for pooling layers. In last part LSTM layers is implemented to extract time information. At end a fully connected layer is used to make final classification. In terms of accuracy author's proposed model showed 98.3% and 97.0%, training and validation accuracy respectively, at epoch 125. Which is better than traditional CNN which showed accuracy of 96.7% and 94.4%, training and validation accuracy respectively. Architecture is shown in figure 2.3

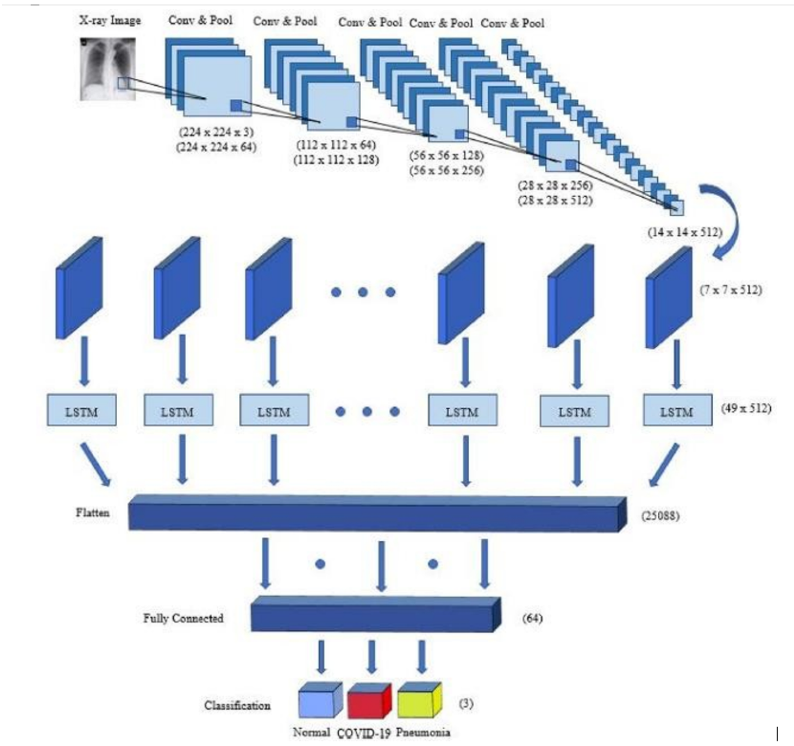

Figure 2.3: framework of hybrid network [3].

2.4 Simple Convolution Based Architectures

Simple convolutional layered models are also very effective when it comes to image classification based on extracted features. These types of architectures uses convolutional layers at front that extract features from input images. These types of architecture learn by through training which features need to be extract. Followed by CNN there are fully connected layer which take extracted features as input and classify given image accordingly. Following are some papers which are based on simple convolutional neural networks.

Author He, Xuehai, et al. [9] proposed self-trans approach for Novel Virus COVID-19 diagnosis which is basically self-supervised learning combined with transfer learning .Author developed his own dataset named as COVID19-CT by collecting data from different papers on medRxiv and bioRxiv this dataset contains 349 scan of NOVEL VIRUS COVID-19 positive patients and 397 scan for NOVEL VIRUS COVID-19 negative people. Author also performed resizing of dataset and splitting in preprocessing stage.

In author's designed self-Trans model self-supervised learning is used alongside transfer learning in order to tune weights (which are pretrained) in such a way bias influenced by original source data is reduced. author created some auxiliary task on dataset images where supervised labels are not human generated. these auxiliary tasks are used to tune network weights. In this way bias influenced by source image were reduced. Author's proposed model achieved an accuracy of 0.86 on COVID19- CT dataset and AUC of 0.49.

Author Shah, Vruddhi, et al [4] developed his own network named as CTnet-10 and compare it with several different networks such as ResNet-50, InceptionV3, VGG- 16, VGG-19 and DenseNet-169. Author used COVIFCT-19 dataset with 738 images in total. This dataset has 349 CT scan images of 216 NOVEL VIRUS COVID-19 patients[7].

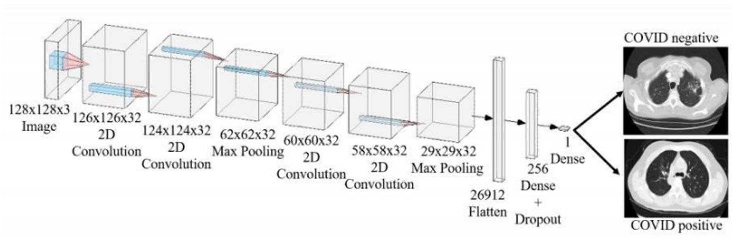

Figure 2.4: CTnet-10 architecture [4].

Author's proposed model receives input of size 128*128*3. This network has two set of two convolutional layer followed by one pooling layer. Output of these two sets is changed into flatten format and sent to dense layer of 256 neurons. At end output layer has single neuron with dropout value of 0.3, this last neuron provides result of classification of given image. first set of convolutional consist of 126x126x32 and 124x124x32 dimensional convolutional layers followed by pooling layer of 62x62x32 dimension. Second set is consisting of 60x60x32, 58x58x32 dimensional convolutional layer followed by pooling layer of 29x29x32 dimension. Beside CTnet-10 author also implement other networks overall author's proposed model showed accuracy of 82.1% where as VGG-19 showed better accuracy of 94.52%.

Zheng, Chuansheng, et al [5] proposed a deep convolutional network named DeCovNet. Dataset used by author is based on 313 patients of Novel Virus COVID-19 and 229 patients without Novel Virus COVID-19. In preprocessing UNet was used to generate lungs masks which was concatenated with original CT-volume to create CT-Mask volume. Author also preformed data augmentation to overcome problem of limited CT scan volume and to avoid overfitting [9].

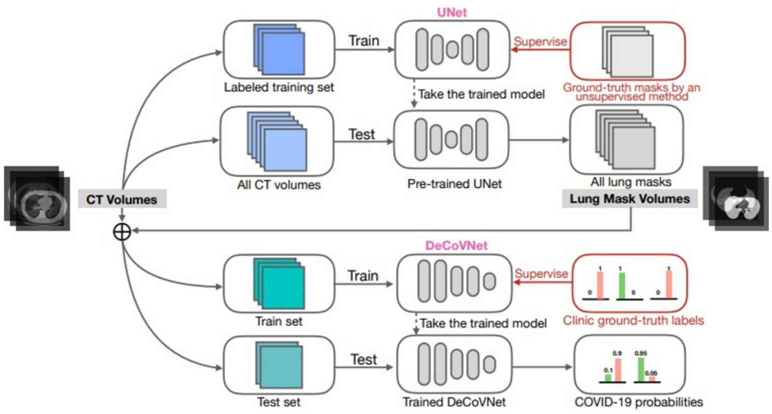

Figure 2.5: CTnet-10 architecture [5].

Author's proposed model DeCoVNet is basically a Deep convolutional neural network which takes CT volume and lungs mask from UNet as input. ThisDeCoVNet network has three main stages. In first stage there are one convolutional layer with kernel size of 5x7x7 dimension, one batchnorm layer and a pooling layer. Second stage is consisting of two residual block each of this block feature map is passed through 3D convolutional layer with batchnorm layer and a connecting convolution. third and final layer is consisting of three convolutional layers followed by dense layer with softmax which provide end classification result. DeCoVNet showed ROC AUC of 0.959 and 0.9776 PR AUC.

S.No	Title, Author, Year	Algorithm / Techniques	Remarks
1	iCTCF: an integrative resource of chest computed.tomography images and clinical features of patients with NOVEL VIRUS COVID-19 pneumonia. Ning, Wanshan, et al April,2020.	Hybrid convolutional neural network and neural network based on CT scan data and clinical features data.At end penalized logistic regression algorithm is used to integrate predictions of made on CT and CF data	Proposed model showed AUC value of 0.978, 0.921 and 0.931 in predicting control, Type 1, and Type 2 patient class respectively.
2	A Fully Automated Deep Learning-based Network for Detecting NOVEL VIRUS COVID-19 from a New and Large Lung CT Scan Dataset. Rahimzadeh, Mohammad et al. September,2020.	Convolutional neural network based on ResNet50V2 in which Feature pyramid network uses concatenation layers.	In single image evaluation phase, the average result of five folds indicates proposed model has an accuracy of 98.449% and sensitivity of 94.96% for NOVEL VIRUS COVID-19 class.
3	A combined deep CNN-LSTM network for the detection of novel coronavirus (NOVEL VIRUS COVID-19) using X-ray images. Islam, Md Zabirul et al. June 2020.	Proposed model is basically a combination of convolutional neural network and long short- term memory (LSTM) network.	Proposed model showed an accuracy of 98.3% and 97.0%, training and validation accuracy respectively
4	Sample-Efficient Deep Learning for NOVEL VIRUS COVID-19 Diagnosis Based on CT Scans. He, Xuehai, et al. April ,2020.	Proposed model uses self-supervised learning with transfer learning.	Proposed model achieved an accuracy of 0.86 on COVID19-CT dataset and AUC of 0.49
5	Diagnosis of NOVEL VIRUS COVID-19 using CT scan images and deep learning techniques. Shah, Vruddhi, et al. July 2020.	Proposed CTnet-10 is based on convolutional neural network.	Proposed model showed accuracy of 82.1%
6	Deep learning Enables Accurate Diagnosis of Novel Coronavirus (NOVEL VIRUS COVID-19) with CT images. Song, Ying, et al. February 2020.	Proposed model DRE-Net is built on pre trained ResNet50. For feature extraction feature pyramid network was added.	Proposed model on testing data achieved AUC of 0.92 in the image level, and AUC of 0.95 and recall of 0.96 in the human level
7	Deep Learning-based Detection for NOVEL VIRUS COVID-19 from Chest CT using Weak Label. Zheng, Chuansheng, et al. March 2020.	Proposed model DeCoVNet is basically a Deep convolutional neural network.	DeCoVNet showed ROC AUC of 0.959 and 0.9776 PR AUC.

Table 2.1: Summary of literature

Chapter 3

Requirement Specifications

3.1 Existing System

In this section, we will look at the work that has already been done on this field. Mostly work done for classification of CT scan images involves deep learning approach. For instance Rahimzadeh, Mohammad et al [8] proposed method based on ResNet50V2 network and a modified feature pyramid network alongside his designed architecture for classifying the selected CT images. Author achieved accuracy of 98.49 percent [1]. Ning, Wanshan et al [1] is another author who developed a novel framework of Hybrid-learning that integrates CF and CT data. Author used penalized logistic regression algorithm at end to combine prediction based on CF and CT data. Overall author achieved striking accuracy with an AUC value of 0.978, 0.921 and 0.931 in predicting Control, Type 1, and Type 2 patient class respectively . Shah, Vruddhi et al [4] is another author who compared different architectures such DenseNet-169, VGG-16, ResNet-50, InceptionV3, and VGG-19 out of which he proved that VGG-19 is giving highest accuracy of 94.52%. Although there are many papers available on CT scan based Novel Virus COVID-19 classification but there is no such proper application or software for user to utilize these models. Furthermore, most of authors did binary classification for diagnosis of Novel Virus COVID-19.

3.2 Proposed System

Our system work on principle of deep learning methodology and use convolutional neural network (CNN) for classification of CT scan images based on features such as ground glass opacity etc. At first, we will apply relevant image preprocessing techniques to make image more suitable for classification and to make model training more efficient in term of computation. After that we will try different well known convolutional neural networks and tune them further , and we will choose model which will provide highest accuracy on testing data. This model will be deployed in client server architecture. Users will be provided with a user-friendly web portal through which they will be able to upload there CT-scan image to server side. After making classification, results will be displayed to user on client side / web portal.

3.3 Requirement Specifications

The functional and non-functional requirements of the application are specified below:

3.3.1 Functional Requirements

- **System must allow user to upload his/her information and CT-scan image to server side:**

  This is one of very crucial functionality. User must be provided with a web page that include form to enter personal information and a mechanism to upload CT-scan image. Once "start test" button is pressed all information must reach Server end.

- **System must apply relevant preprocessing:**

  Before user provided Ct-scan image is passed to trained model for classification. Relevant preprocessing must be applied. Preprocessing must be same as applied on images of training dataset.

- **System must correctly classify user provided CT-scan image:**

  This is the core functionality of system. Once preprocessing is applied on user provided image. Image is passed to trained model which must classify it into Control, Regular and Sever Correctly.

- **System must store diagnosis result and user information into database for future use.:**

  In order to generate stats in future based on data, system must store diagnosis result and user information into database. Once user provided CT-scan image is classified it's result and user information which include Name, CNIC, City etc. must be added to database.

- **System must be able to display classification result to user:**

  After classification is carried out successfully. Server side must render a web page which will present report of diagnosis to user. This report must sate user information and the result of diagnosis i.e. (control, Sever or Regular class). In addition to this system must also provide functionality to user to download report in pdf format.

- **System must display stats graph:**

  Information gather over time could be used to display different stats graph. Once person click on stats menu, A web page must render which display table showing current cases of control, Regular and Sever class . Second graph should display positive cases of last seven days. And last graph is a bar graph showing active cases in different cities

3.3.2 Non-Functional Requirements

- **Availability:**

  It is very crucial for our system to remain available for end user because of nature of our application. One of the problem that may influence available of system is failure in database operation. Since database operation are involve in all other system functionalities hence if database operation fails other functionalities are also affected. To overcome this problem, we use exception handling so that even if database operation fails overall system remain operational.

- **Reliability:**

  It is also very important for our system to be reliable and makes accurate prediction. To ensure reliability of our system we choose model which gave highest accuracy of 90% on testing data. This model was deployed in client server architecture.

- **Usability:**

  Usability is also very important for our system. Since our application will be used in critical situation by user of different age groups. Hence user interface must be simple and user friendly. To ensure usability of system we applied Normans usability principles while designing interface. We make sure that all controls are visible and properly labeled. We also add different textual message and animation to give user proper feedback. Furthermore, all elements were properly mapped.

- **Re-usability:**

  During development we also assured that system support reusability. In order to achieve this different functionality were developed in different modules and similar functionalities we grouped in same file. Furthermore, comments were used to properly labeled functionality of these modules which also helps system to be reusable.

3.4 Use Cases

Our application is aimed at general public and people belonging to various other fields that can benefit from our application. Goal of the user would be to use the application to preform diagnosis of Novel Virus COVID-19 and its morbidity by giving image of his/her CT-Scan as an input to the application. The user required to provide his/her credentials so that a proper record would maintain at backend this data will be used in future in order to provide different kind of statistics and information. User required to provide input in form of CT-scan image and then that image would be processed in order to diagnose Covid through a classification that would display the result on screen after successful classification. General use case diagram is shown in Figure 3.1

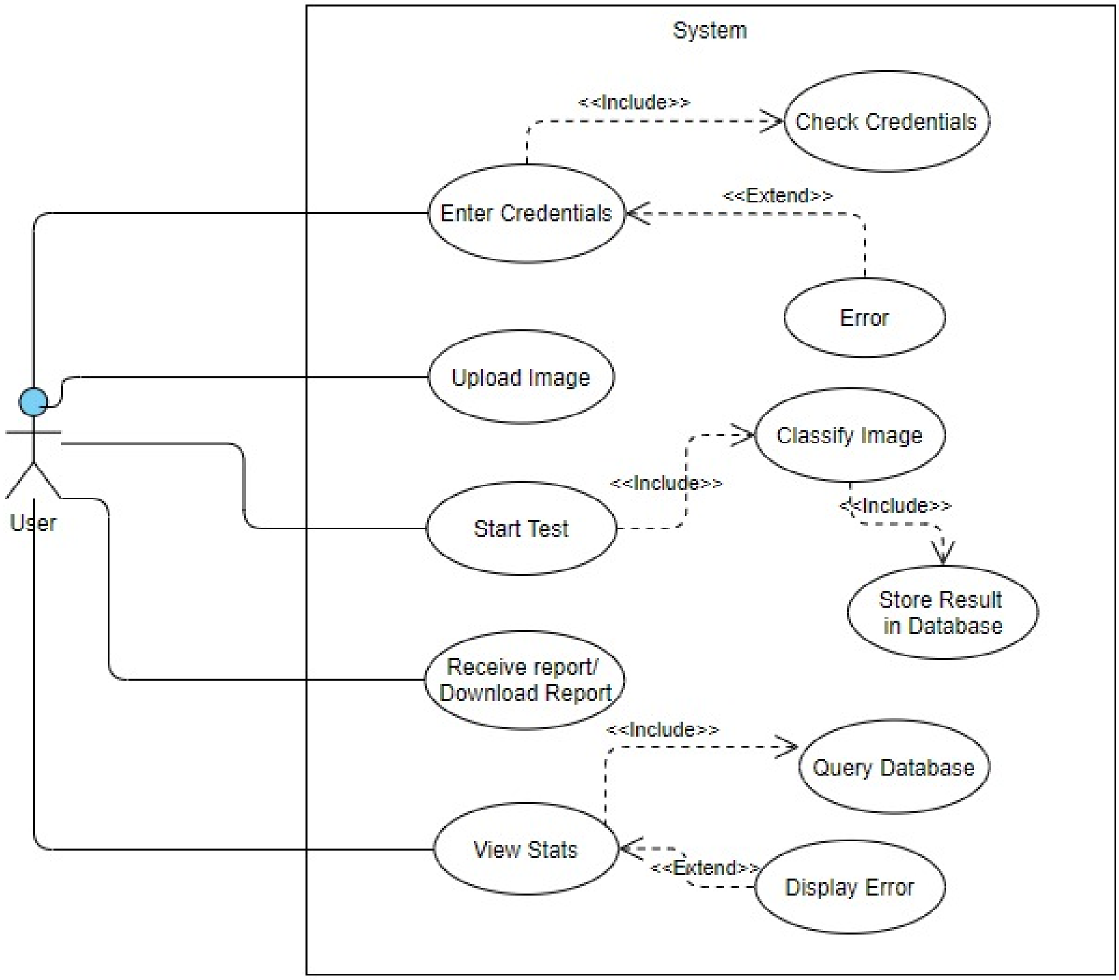

Figure 3.1: Use Case Diagram

3.4.1 USE CASE 1: Enter Credentials

The following Figure 3.2 and Table 3.1 shows how to enter user information into form.

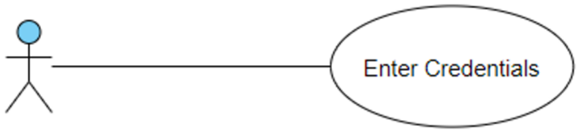

Figure 3.2: Use case 1: Enter Credentials

Title	Enter Credentials
Actor	User
Description	This use case specifies the details of the patient entered by the user
Main success factor	Application should should take user information through textfield and radio button.
Pre-condition	Web application should be running
Post-condition	The application form show display user entered information

Table 3.1: Enter Credentials

3.4.2 USE CASE 2: Upload Image

The following Figure 3.3 and Table 3.2 shows how to upload user Ct-scan image.

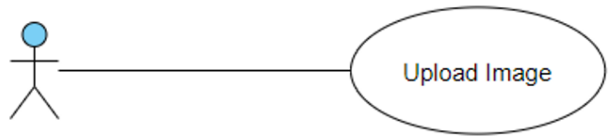

Figure 3.3: Use Case 2: Upload Image

Title	Upload Image
Actor	User
Description	This use case specifies the action of choosing an CT-scan image by the user
Main success factor	Application accepts the desired image chosen by the user and show preview
Pre-condition	Application should be running
Post-condition	The application should must show preview of the chosen image on the screen implying its acceptance

Table 3.2: Select video from gallery

3.4.3 USE CASE 3: Start Test

The following Figure 3.4 and Table 3.3 shows how to start test procedure and send data to server side.

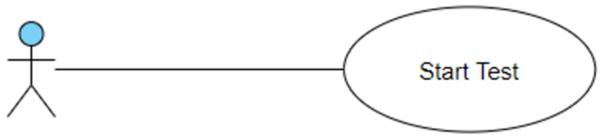

Figure 3.4: Use Case 3: Start Test

Title	Start Test
Actor	User
Description	This use case specifies how to start testing procedure
Main success factor	Application must Send data to server side Once "Start Test " Button is pressed and show processing animation followed by result report
Pre-condition	Application should be running
Post-condition	Application must display processing animation indicating that systme is processing followed be result report

Table 3.3: Start Test

3.4.4 USE CASE 4: Receive report/ download report

The following Figure 3.5 and Table 3.4 shows how to get report and save in device memory.

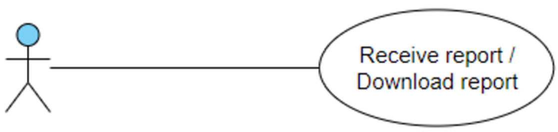

Figure 3.5: Use Case 4: Receive report/ download report

Title	Receive report/ download report
Actor	User
Description	This use case specifies the interaction of the user with application to download report
Main success factor	report is downloaded into user device
Pre-condition	image is classified by server and report generated
Post-condition	report is downloaded into user's desired location

Table 3.4: Receive report/ download report

3.4.5 USE CASE 5: View Stats

The following Figure 3.6 and Table 3.5 shows how we can view stats graph page.

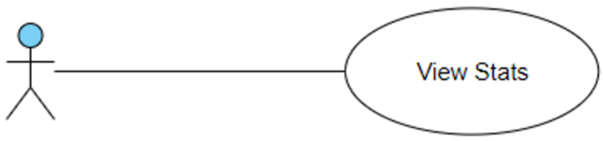

Figure 3.6: Use Case 5: View stats

Title	View Stats
Actor	User
Description	This use case specifies the action of viewing the Stats graph page
Main success factor	Application should display stats page with graphs and information extracted from database
Pre-condition	Network should be running and database should be connected
Post-condition	The application should display a stats Web page

Table 3.5: View Stats

Chapter 4

Design

System design defines the architecture, components, modules, interfaces, and data for system to satisfy specific requirements. Following represents the system design for developed application.

4.1 System Architecture

The developed system is a web application. The application has a simple architectural design comprising of following steps:

- Unlabbled image upload from client side.
- Trained Model who classifies the image.
- Output the result after successful classification to the user and save it in database.

The main physical components in the proposed system are input, processing and output. System architecture is illustrated in Figure 4.1.

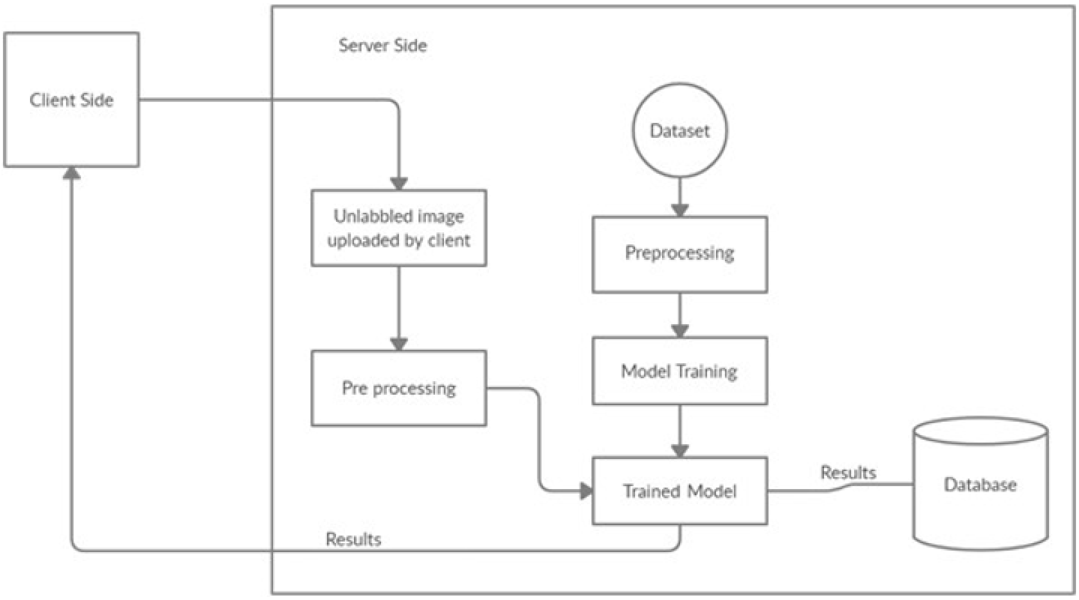

Figure 4.1: System Architecture

4.2 Design Constraints

Our model will only work on specific typed of CT scan images it will not work on other format and types like X-ray etc. Images which user enter for classification should be clear and should be distortion free. Those images should be pass to server for diagnosis in which lungs are at open state and clearly visible.

4.3 Design Methodology

Before training model some preprocessing will be applied on dataset. In this preprocessing stage at first, we will resize image to make training phase more computational efficient. Image will be resize in such a way that their features remain intact and visible. After resizing images, we will convert these images into matrix format such that one matrix contain all images while other contain their labels. After

matrix creating, we will normalize our dataset so that all pixel have value between 0 to 1. This makes convergence faster while training the network. After that we will shuffle dataset so that in batch training each batch contain mix instance form different classes. After shuffling dataset will be split into training, validation and testing portion. At last, a convolutional neural network will be used to train model. When user send an image for diagnosis same preprocessing techniques will be applied as done on training dataset and at end trained model will be used to classify user's unlabeled instance. A general approach is demonstrated through Figure 4.2.

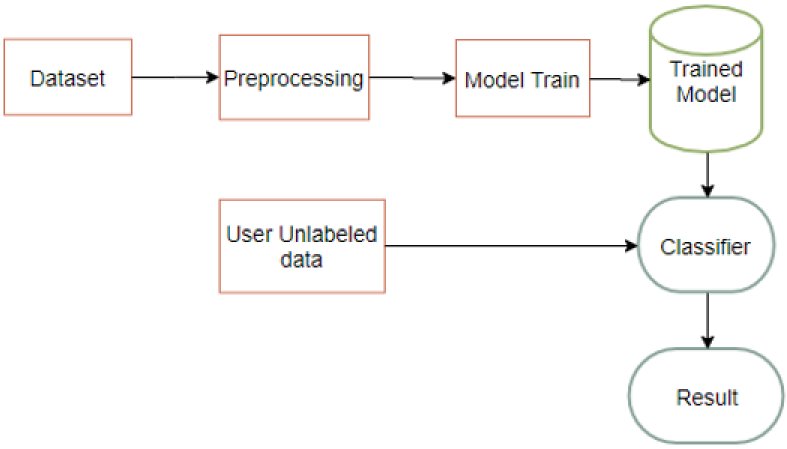

Figure 4.2: Methodology

4.4 High Level Design

High level designs have different viewpoints. Following are typical viewpoints:

4.4.1 Component Diagram

The component diagram shows the overall high level view of the application and explains the functionalities and responsibilities of the system, and how different tasks are divided and assigned to different components. Our system can be divided into three main modules as illustrated in Figure 4.3.

- Client Side
- Server Side
- Database
- Classifer
- Preprocessing

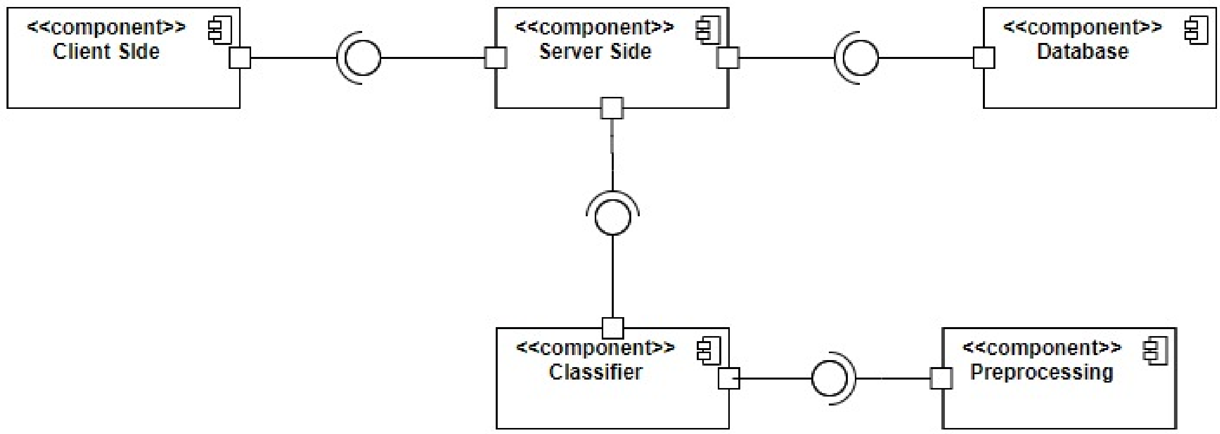

Figure 4.3: Component Diagram

4.4.2 System Interaction Diagram

System interaction diagram shows how the various processes within the system interact with one another. It also shows how the messages are exchanged between the objects to carry out the functionalities of the given scenario. Figure 4.4 shows the system interaction diagram.

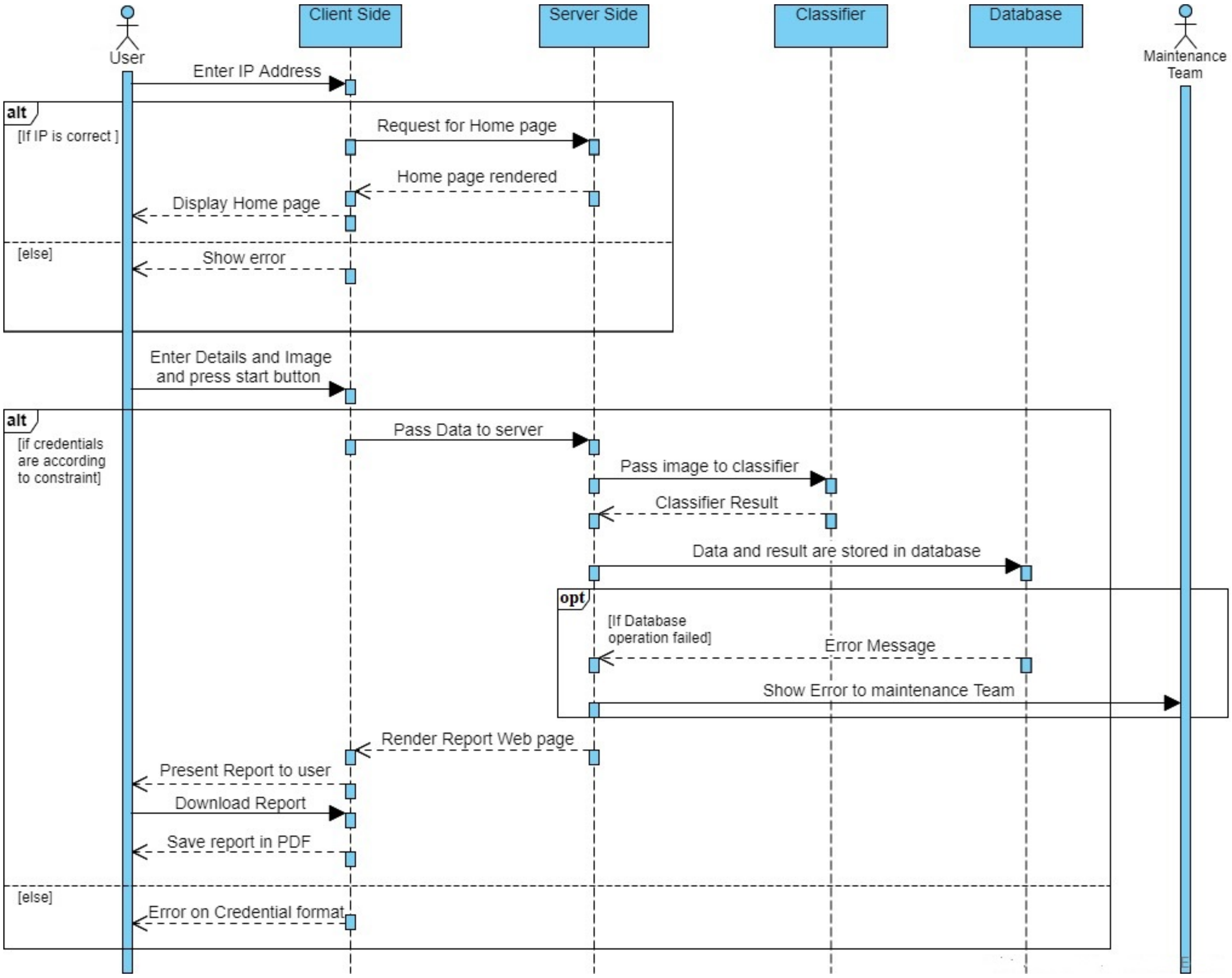

Figure 4.4: System Interaction Diagram 1

Sequence of interaction that take place to access Stats web page are shown in Figure 4.5 below.

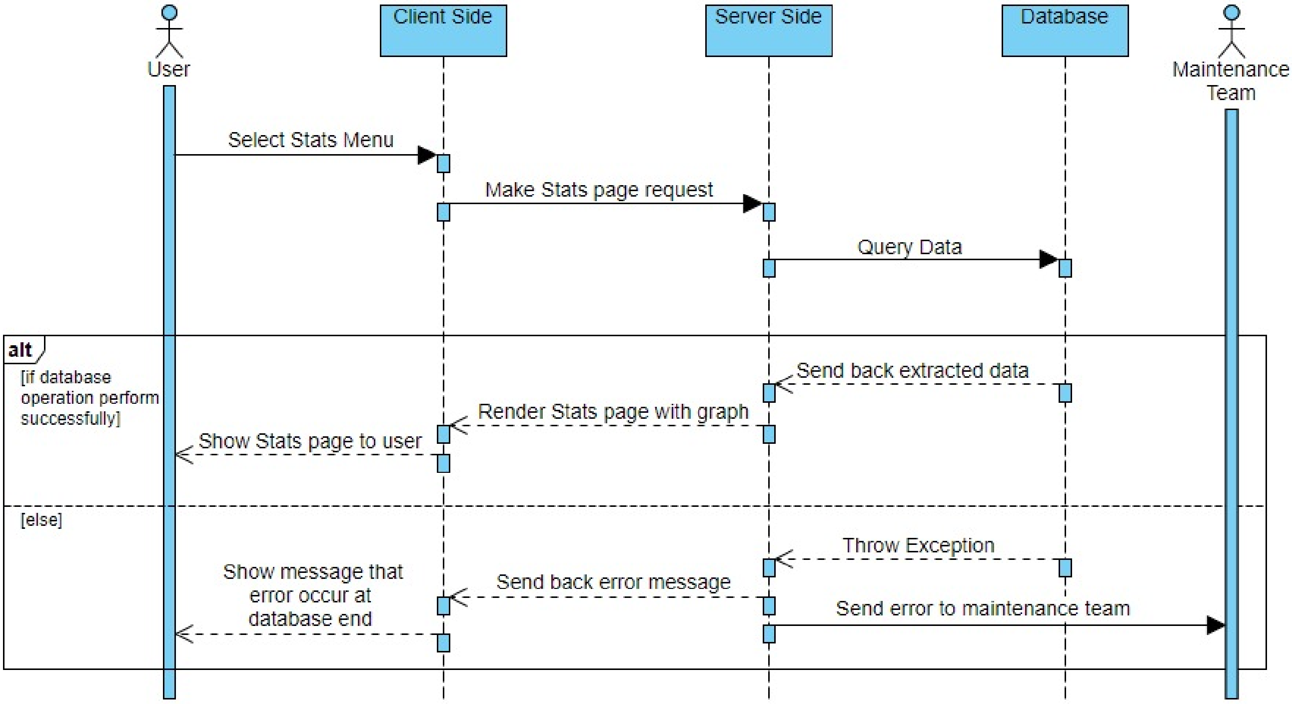

Figure 4.5: System Interaction Diagram 2

4.4.3 Class Diagram

Class diagram shows the main building block of object oriented modelling. Figure 4.6 shows the class diagram below.

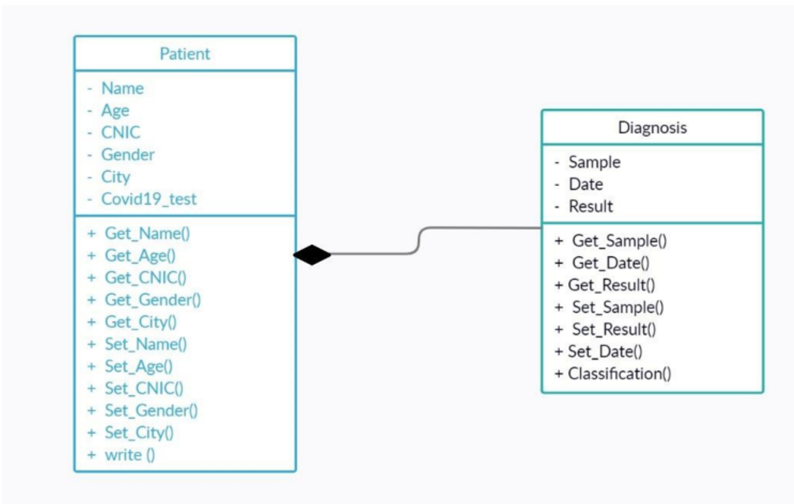

Figure 4.6: Class Diagram

4.5 GUI Design

The product offers easy to use and manage GUI for proper usage without having to understand any complicated technicalities. The application is a COVID-19 diagnoses system and GUI built around it works in manner, that makes it easy for users to upload their image and enters their credentials and then get the required output on the screen the result of the diagnoses. Figure 4.5 shows the GUI of the Home screen and figure 4.6 shows the test screen of Application.

- User will now access home screen where core functionality is place. user can enter his data and start diagnosis test from this web page as shown in Figure 4.7.

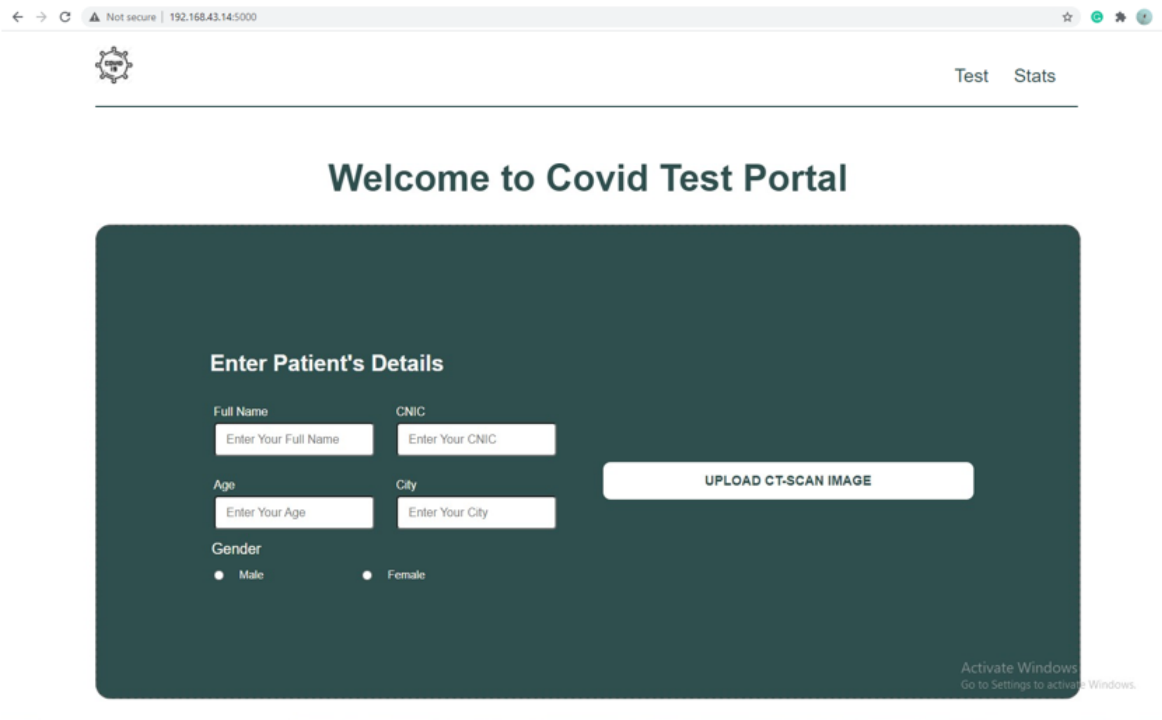

Figure 4.7: Welcome/Splash Screen

- this is the preview home page after all information is added. Figure 4.8.

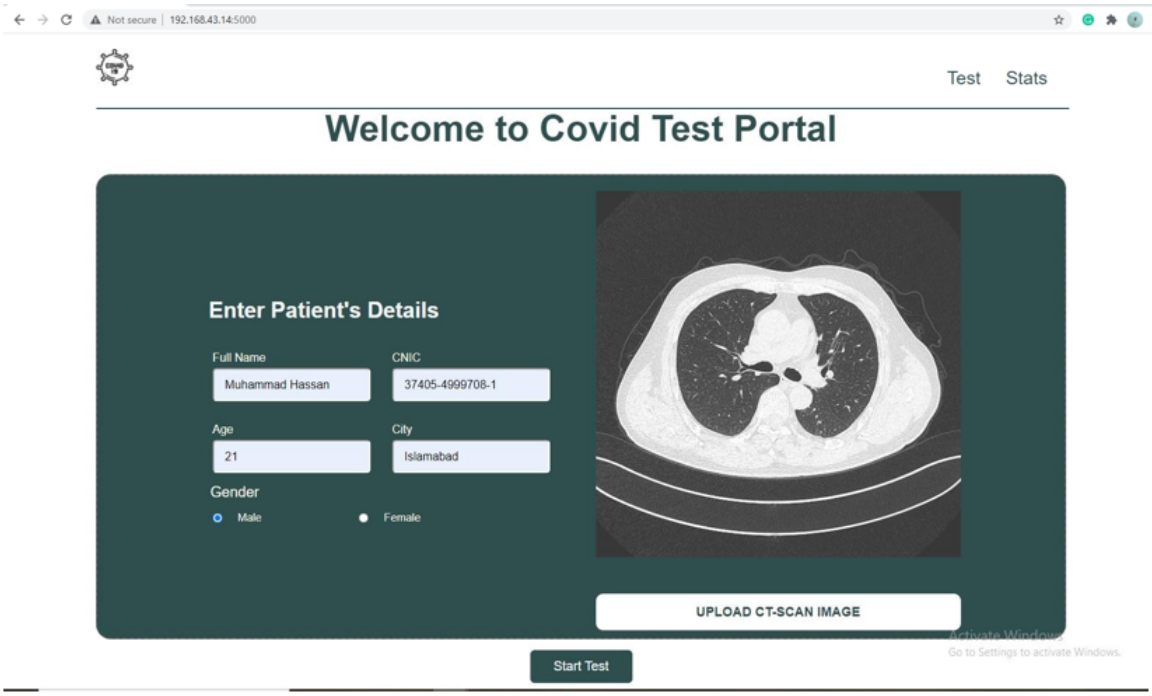

Figure 4.8: Home Screen

Chapter 5

System Implementation

5.1 System Architecture

Implementation of system was performed in two main phases. In first phase objective was to develop a convolutional neural network model which will take CT scan image of lungs as an input and classify into diagnosis classes. Once a model is trained which gave satisfactory results on testing data second phase of implementation was start. In this phase objective was to develop a client server architecture which will provide end user a web portal. Through this portal user can upload images of Lungs CT scan for diagnosis. These images once reached to server side are classified by trained model and results are delivered to user through web portal.

5.1.1 Algorithmic Workflow

The system activity diagram showing the overall work-flow of the system is presented in Figure 5.1

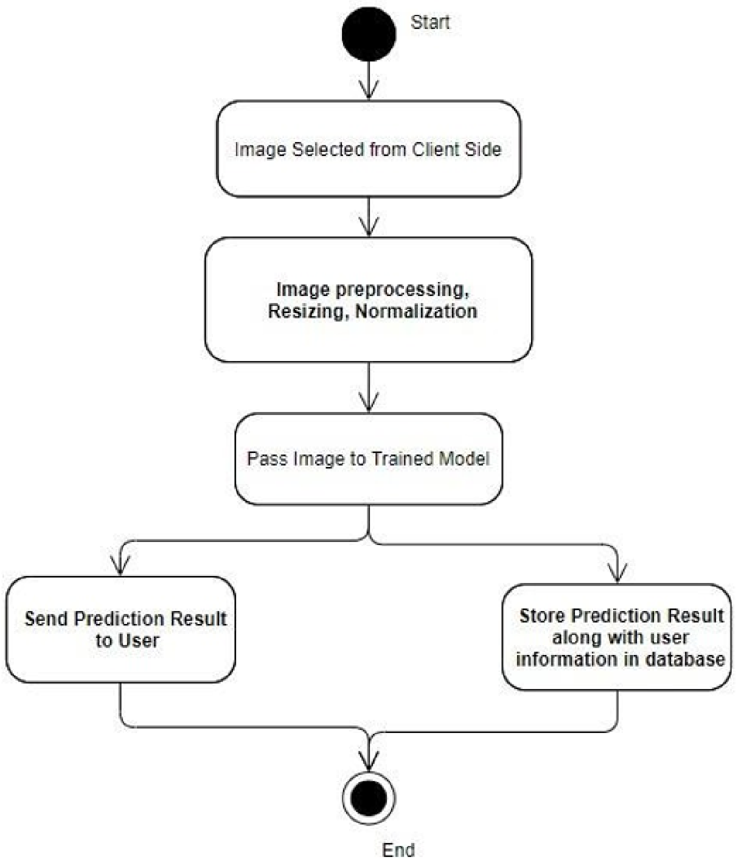

Figure 5.1: Algorithm Workflow

5.2 System Components

The following is a list of the components of the system based on algorithms deployed:

- Implement client server architecture
- Dataset
- Image pre-processing
- Model Training and Image classification
- Database

5.2.1 Implement client server architecture

Client server architecture is one the major component of system as it connects customer to server where all processing is being performed. Client server architecture allow user to send CT-scan image at server side where actual classification is preformed and results are generated, and these results are then sent to client side. In order to implement client server architecture, we used python flask library which allowed us to implement different network routes. We designed three routes in total. First route provides user home page which has a web form through which user can enter his information and upload CT-scan image. Upon pressing “start test” button this data is send to second route. Second route extract data from post request. Image from this data is passed to trained model. Result of classification and user data are store in database. At end a web page is rendered through which results are send to user. User can also download these results. Third and final route renders a web page. This web page contains several graphs these graphs take information from database and display stats to user.

5.2.2 Dataset

5.2.2.1 Original Dataset

In beginning we started working on original dataset use by Ning, Wanshan et al [4] in his propose system for classification of Novel Virus COVID-19 patient. This dataset was collected from randomly 61 and 93 individuals with and without Novel Virus COVID-19 infection by Liyuan and Union Hospital of china. This dataset was composed of 19,685 CT scan images out of which 5705 images belonged to non-informative class ,4001 images were belonged to positive class and 9975 images were from negative class. All images have dimension of 512x512 with three channels. These images are basically slices of CT scan which shows human lungs in axial view. Since the initial objective of our project was diagnosis of Covid positive and Covid negative patient we took these two classes from this dataset.

5.2.2.2 Custom Patient-based data

Later, in implementation phase we found that there is need to make patient base dataset in which all images are also labeled patient wise so that all images from one patient lays in either training, testing or validation set. El. And Huazhong University of Science and Technology provide patient wise data, but this data was not in compiled form, so we manually downloaded 240 patient’s data. Originally there were six Morbidity classes including Critically ill, Severe, Regular, mild, suspected and control. Among these classes suspected class is the one which can’t be detected through CT scan as these are the CT of patients who had positive PCR testing yet according to CT scan, they are healthy, so we exclude this class. Furthermore, critically ill class had only 35 patient data, so we combined severe and critically class together as they both had similar nature. In this way we formulated a dataset with three classes Sever (patient highly effected by Novel Virus COVID-19) sample image is show in figure 6.1, Regular (patient less effected by Novel Virus COVID-19) and Control class (individuals without Novel Virus COVID-19). Each class had 80 patient’s data. Original patient’s data had multiple images some had lungs in expiration form, in other words images in which lungs are not shown so these images were excluded manually for all 240 patients. On average each patient was left with 50 images.

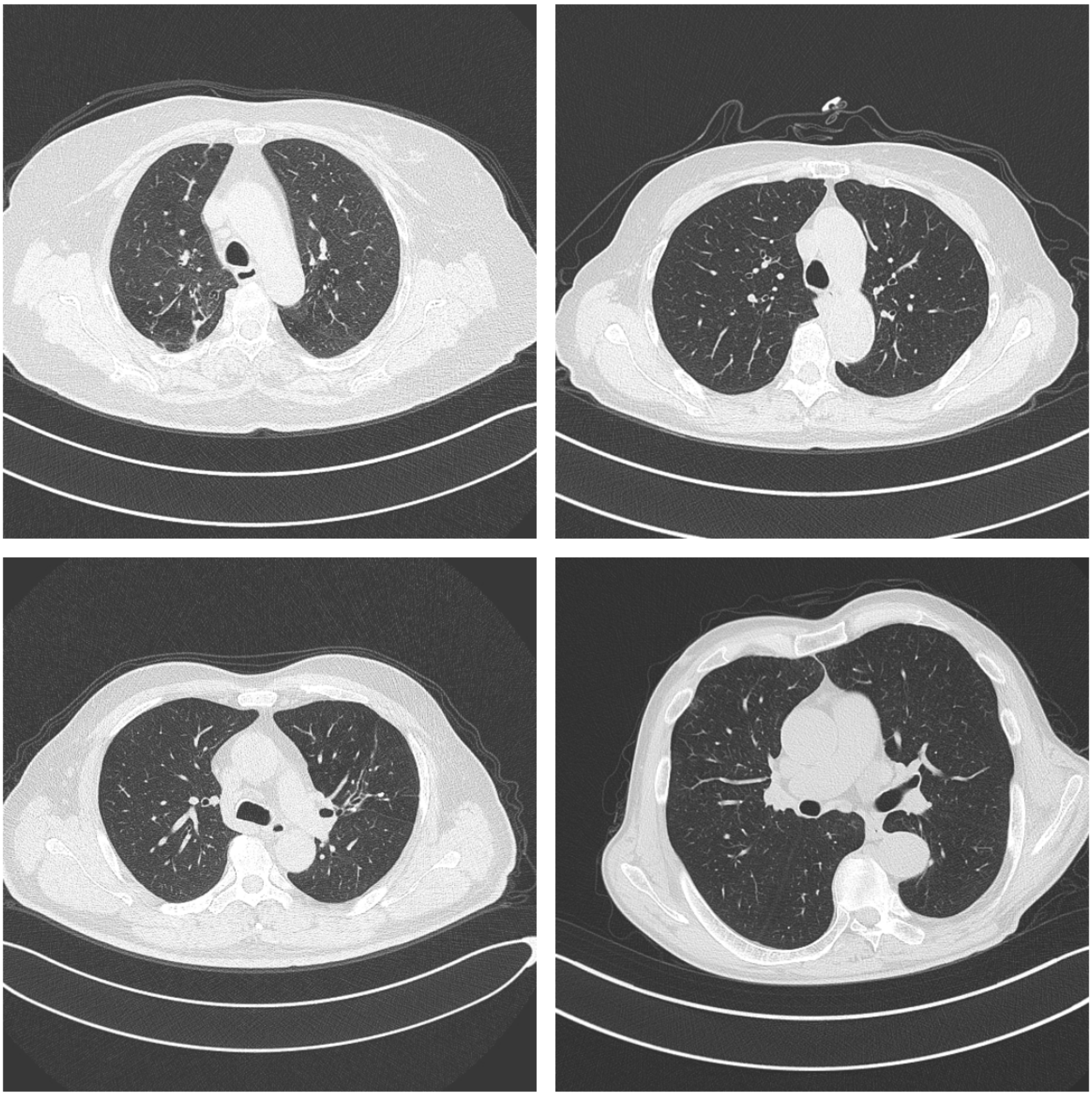

Figure 5.2: This figure shows sample images from Control class. Here lungs are at their healthy state and there at no sign of ground class opacities or paving marks.

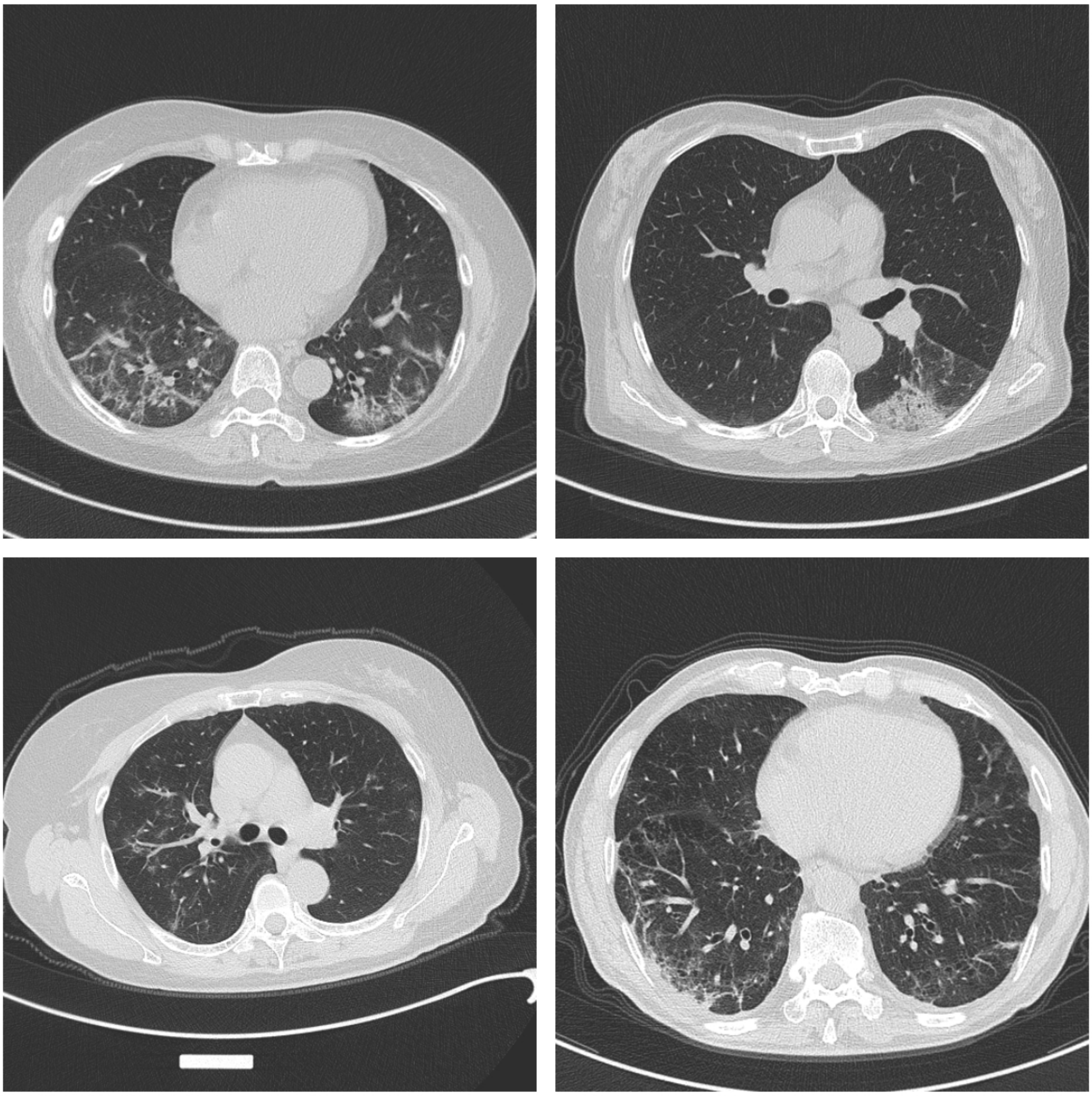

Figure 5.3: This figure shows sample images from Regular class. These images show lungs of patient suffering from mild COVID-19 virus and there are some sign of ground class opacities or paving marks.

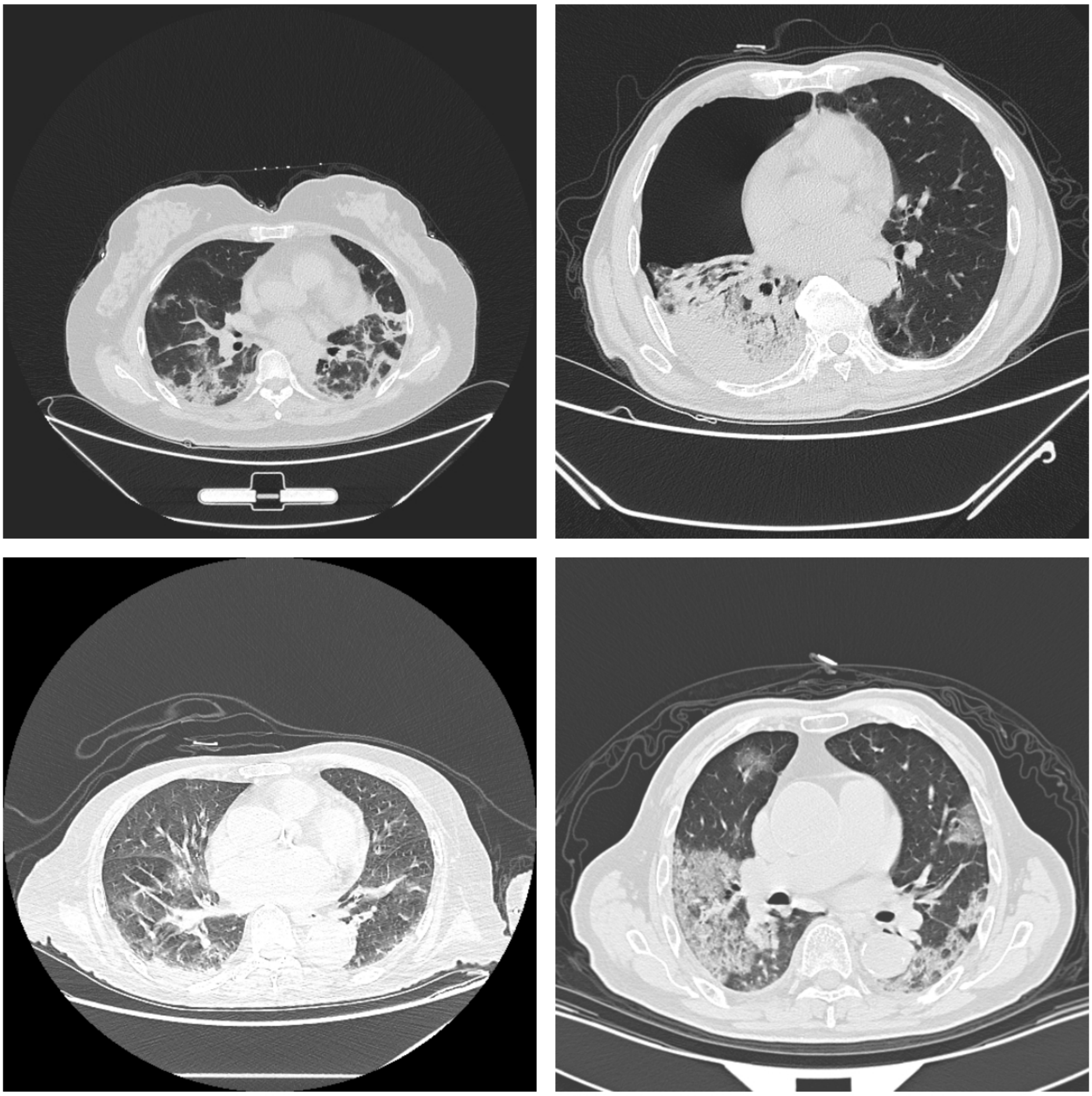

Figure 5.4: This figure shows sample images from Severe class. These images show lungs of patient suffering from Severe COVID-19 virus and there are intense sign of ground class opacities or paving marks.

5.2.3 Image pre-processing

CNN are very robust in feature extraction and making classification on basis of these features and one of the major advantages of CNN is that they work pretty well for end-to-end approach hence preprocessing was used at very small extent. Image preprocessing procedure involved in project are as following.

- **Resizing of images:**

Original size of images was 512x512. Training on such dataset would have very expensive in term of memory. Hence, we down scale image to a point where features of image remain intact, and their size also reduce. We down scale images to 224x224.

- **Splitting dataset into training, validation, and testing:**

  Dataset was split patient wise. With 60% patient in training set, 20% patient in validation set and 20% patient in testing set. Overall training set has 7235 images, validation has 2199 images and testing has 2349 images.

- **Converting images into numpy matrix format:**

  Since we are creating model through keras which takes dataset in form of numpy array hence we made training, validation and testing numpy matrix containing images and their corresponding y-matrices containing labels.

- **Normalization of dateset:**

  it is very important to normalize dataset before training as it helps optimization algorithms to converge fast. Hence in order normalize image dataset we divide image containing matrices with 255 so that all pixel values range from 0 to 1.

- **Shuffling of dateset:**

  Since training is performed in batches hence it is very important that each batch gets shuffled data. we used random library to shuffle Training, Validation and Testing matrices.

5.2.4 Image classification and Training Deep learning model

After image preprocessing, image is passed to train model which classify image into Control, Regular or Severe class. This model which is deployed at server side showed highest accuracy on testing as compare to other models. We conducted test on two dataset first dataset is the one used by author Ning, Wanshan et al [4] and second dataset is self-constructed in which images were also labeled patient wise. we trained different well know architecture such as AlexNet, VGG16 and ResNet50 which are considered high preforming universal CNN model. we further modified these models to extract best out of them and to achieve highest possible accuracy through these models. Detail regarding these models and modification made in them are stated in section below.

5.2.4.1 Training on original dataset

We started with simplest 2 convolutional layer neural network and approach was to gradually increase complexity. At first images of dataset were downscaled to 250x250 to save computational resources and then images were normalized after that a model was trained which had first convolutional layer of 16 filters (with 3x3 kernel size) followed by max pooling layer and second convolution layer with 32 filters followed another max pooling layer at end 2 dense layers were applied first one with 64 units and second one with 1 unit. Upon training, this model showed very unusual behavior from the very first epoch both training and validation loss become decimal figure and training accuracy became 0.97 and validation accuracy became 1. Achieving 100% percent accuracy in very first epoch was very unrealistic. We further trained model with 3 and 4 convolutional layers in both cases validation accuracy became 99% and training accuracy became 95% from the very first epoch.

5.2.4.2 Feature Extraction and Classification on Patient-based data

Newly composed patient-based dataset was splinted into 60 percent training 20 percent testing and 20 percent validation set. Upon training a very simple two convolutional layer and two dense layer model gave 99 percent accuracy on training set and 40 percent on validation set which verified our assumption that previous dataset contained multiple images from single patient which were more or less same and hence validation and testing set was not completely separate from training set. With this new dataset now, objective was to increase validation accuracy but for that we required some starting point. Architectures like Alexnet etc. are best suited groundwork for classification problems, further detail is given below.

Convolutional neural network is very effective when it comes to detect features in images and classify image on basis of these features. Architectures like AlexNet, Resnet are very robust for the extraction of features as shown by their exceptionally good performance on larger dataset such as ImageNet. Following are the Architectures which were used to extract features alongside with modified fully connected network for classification.

- **AlexNet and Modifications:**

  AlexNet has the highest accuracy on ImageNet. Basic architecture of Alexnet is consist of five sets of convolutional and maxpooling layers which are used as feature extractor followed by 3 dense layer which acts as classifier. Although AlexNet gave around seventy percent accuracy after training model for 7 epochs but gape between training loss and validation loss indicated overfitting hence following modification were preformed

  In order to overcome the over fitting, dropout rate of second last and last layer was increased to 0.9 and 0.8 respectively and after training model for 18 epochs, gap between training and testing accuracy was decreased. Furthermore l2 regularization was also added in Convolutional layers and after model was trained for 34 epochs which further aid in overcoming overfitting and increased validation accuracy.

- **VGG16 and Modifications:**

  VGG16 is also one of very high preforming architecture on ImageNet. Basic structure of VGG16 contains 13 convolutional layers in between these convolutional layers there are 5 maxpoling layers. At end there are 3 dense layers. at first original version of VGG16 was trained for 9 epochs which gave less accuracy as compare to AlexNet.

  Training VGG16 architecture on our dataset showed sign of overfitting to overcome this we increased dropout rate of 3rd last and 2nd last dense layer to 0.7 and 0.9 and trained model for 6 epochs which reduced overfitting. Furthermore we also added l2 regularization in convolutional layers and trained model for 6 epochs which helped in preventing overfitting further.

- **Resnet50 and Modifications:**

  Resnet proves to be very effective on our dataset as it showed highest accuracy among all models it first original ResNet50 based model was trained for 6 epochs but it also shows overfitting signs to overcome this following modification were made.

  In order to overcome overfitting at first dropout rate in dense layer of ResNet50 was tuned to 0.9 and model was trained for 7 epochs. In second attempt Batch normalization was also applied in last dense layer and model was trained for 6 epochs which gave highest accuracy on testing set.

5.2.5 Database

Database hold user information which includes user's name, age, city, CNIC , date, result of his diagnosis and image serial number. Database store information regarding testing results of patient. This information is use in later stage to show different stats. To Implement database for this project we use mangodB which is a NoSQL database well known for its ease of use and reliability. Furthermore, it easily integrates with python through pymongo library

5.3 Tools and Technologies

Database hold user information which includes user's name, age, city, CNIC , date, result of his diagnosis and image serial number. Database store information regarding testing results of patient. This information is use in later stage to show different stats. To Implement database for this project we use mangodB which is a NoSQL database well known for its ease of use and reliability. Furthermore, it easily integrates with python through pymongo library.

Chapter 6

System Testing and Evaluation

6.1 Graphical User Interface Testing

GUI is the key element of any system as through GUI user can intact with system and underlying technology. While developing user interface Human computer interaction principles were followed to make application functional and to ensure positive user experience. GUI was designed in a way to make it simple and essay to use so that user could accomplished his task with less effort. Overall GUI which is basically a website has two main pages first page provide core functionality and allow user to upload image along with some user information. On clicking "start test "button user is provided with test report. Second page provide user statistics of previously conducted tests. Website is tested from the perspective of GUI and different test cases. Website proves to be effective in term of it use.

6.2 Usability Testing

Usability testing was made to make sure that the website is effective and usable in its functionalities. Fundamental need for this testing was to ensure that during real life senior when a patient interact with system, he does not face any difficulty in using it. Website is designed in a very simplified form all functionalities and buttons are very clear. Core functionality which is to upload image is strategically place on main page. User can effectively achieve desire result with minimum number of interactions. During usability testing website 5 individuals used this website and find it essay and effective in its use.

6.3 Software Performance Testing

Software performance testing was conducted to check responsiveness and stability of system. As this system is implemented in client server architecture hence all intense work required in classification of image is performed on server side on server resources this save not only resource of clients but also reduces execution time due to use of GPU at server side however execution time and system responsiveness may get effected according to network quality of user. Overall system is very efficient and responsive according to individuals who tested it. Beside classification phase in which system takes few seconds, system preform all task in reasonable amount of time.

6.4 Exception Handling

The main exception that can occur during execution of client server architecture is that system may fail to connect to database in that case system will be unable to save user data into database nor it will show any stats. In such case both user and authorities at server side will be notified.

6.5 Load Testing

During load testing phase system was given multiple testing set images to check how system will perform in such condition. Since we are not using any heavily resource dependent preprocessing and images are passed to trained model almost as in their original condition therefore system takes short amount of time in classification of single image hence overall system took reasonable amount of time in classification of multiple images.

6.6 Test Cases

The following test cases were implemented.

6.6.1 Launching website test case

The following Table 6.1 shows test case that that website launch correctly and work across all types of devices.

Test Case ID	01	
Description	Launching website	
Applicable for	All type of device which have web browser	
Requirements	REQ01	
Initial conditions	None	
Step	Task and expected result	
1	Open application	Pass
2	Verify all website element looks properly aligned across all devices y	Pass

Table 6.1: Table for test case 1

6.6.2 Sending data to server side

The following Table 6.2 shows test case that ensures that data is properly passed to server from client side.

Test Case ID	02	
Description	Sending data to server side	
Applicable for	All type of device which have web browser	
Requirements	REQ02	
Initial conditions	Website lunch properly	
Step	Task and expected result	
1	Enter patient information in text box	
2	Click on upload image button and enter image	Pass
3	Make sure text data and image is receive at server end	Partial

Table 6.2: Table for test case 2

6.6.3 Image Classification

The following Table 6.3 shows test case that ensures that image classification is preformed properly.

Test Case ID	03	
Description	Image classification	
Applicable for	All type of device which have web browser	
Requirements	REQ03	
Initial conditions	Data is properly received at server end	
Step	Task and expected result	
1	ake sure all preprocessing is preformed properly	
2	During classification time user is notified by processing animation	Pass
2	After classification is preformed result are sent to user on client side.	Pass
2	Allow user to download diagnosis report	Pass
3	Storing results and patient information in database	Partial

Table 6.3: Table for test case 3

6.6.4 Displaying Stats Graph

The following Table 6.3 shows test case that ensures that stats graph are shown to user properly.

Test Case ID	03	
Description	Displaying stats graph	
Applicable for	All type of device which have web browser	
Requirements	REQ03	
Initial conditions	Database is connected	
Step	Task and expected result	
1	On clicking stats button in menu stats web page should open	Pass
2	Relevant data should be extracted from database and displayed in form of graph	Partial

Table 6.4: Table for test case 4

6.7 Analysis and discussion of results

In this project we trained multiple models based on different well know architecture accuracies of these trained model are stated in evaluation matrix. Among all these models the one with highest accuracy was chosen to deploy at server side for classification of CT-scan images. As we have stated earlier two dataset were used in this project at first we use dataset used by Ning, Wanshan, et al [4] but after realizing that there were multiple images from single patient which causes validation set and testing not completely separate from training set we decided to compose patient base dataset so that all images from same patient gets in either training or validation or testing set. As we are using end to end approach preprocessing only involved rescaling images according to architecture and normalization of images. After preprocessing well know architecture such as AlexNet, VGG16 and ResNet50 were trained along with some modification. Results from these architectures are discussed in section below.

6.7.1 Results on original dataset

For binary classification original dataset contain 13980 images out of which 4001 belong to positive class and 9979 belong to negative class. Upon training 2 convolutional layered, 3 convolutional layered and 4 convolutional layered neural network following results were obtained after 3 epochs.

Data split	No of Conv layer	Train Accuracy	Validation Accuracy	Test Accuracy
60% training 20% validation,20% testing	2	0.9999	1	1
	3	0.9975	0.9982	0.9989
	4	0.9971	0.9993	0.9992
70% training 15% validation,15% testing	2	0.9991	0.9976	0.9980
	3	0.9991	0.999	1
	4	0.9991	0.9962	0.9952
70% training 15% validation,15% testing	2	0.9998	1	0.9993
	3	1	0.9993	0.9993
	4	0.9969	1	0.9986

Table 6.5: Table of accuracies for seen faces of MIRACL-VC1

Reason of having very high training validation and testing accuracy after 3 epochs is that images of validation and testing set are similar images to those in training set as they come from same patient.

6.7.2 Results on patient-base dataset

This dataset had 3 classes Control (negative), Regular and Severe each class each class had 80 patient and each patient had 50 images on average. dataset was split 60% training, 20% validation and 20% testing. dataset was split on basis of patient so that patient of training testing and validation are different following are the results of training different architecture along with their modified form.

6.7.2.1 AlexNet Results

At first original version of AlexNet was trained on dataset although 0.71 percent accuracy was achieved on validation set but there was huge gap between training and validation loss indicating overfitting. To overcome that modified version of AlexNet was trained with increased dropout rate and l2 regularization. Following are the results obtained through Alex Net

The table given below is for unseen male faces:

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Simple AlexNet	12.23	10.00	10.00
Alexnet with Increased dropout rate 0.9 and 0.8 (second last and last layer)	27.77	5.00	16.00
With dropout and l2 regularization	11.54	10.00	10.00

Table 6.6: AlexNet Results

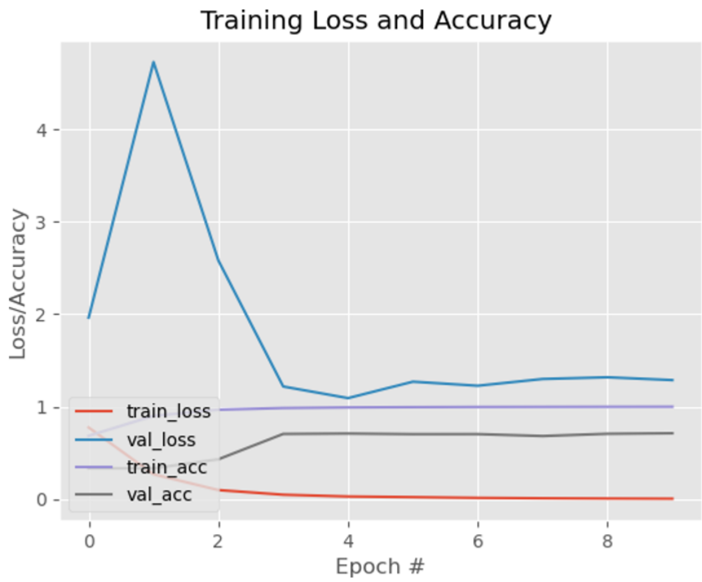

Figure 6.1: Simple AlexNet

Figure 6.2: AlexNet with increased dropout rate

Figure 6.3: AlexNet with increased dropout rate and l2 regularization

6.7.2.2 VGG16 Results

VGG16 was also trained on dataset at its original from in start but to reduce over fitting we tune dropout rate of second last and third last layer to 0.9 and 0.7 respectively to further decrease overfitting we also added L2 regularization following are the results obtained through VGG16.

The table given below is for unseen male faces:

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Simple VGG	0.98	0.38	0.57
VGG with Increased dropout rate 0.9 and 0.7 (second last and third last layer)	0.95	0.51	0.86
With dropout and l2 regularizationn	0.96	0.448	0.80

Table 6.7: AlexNet Results

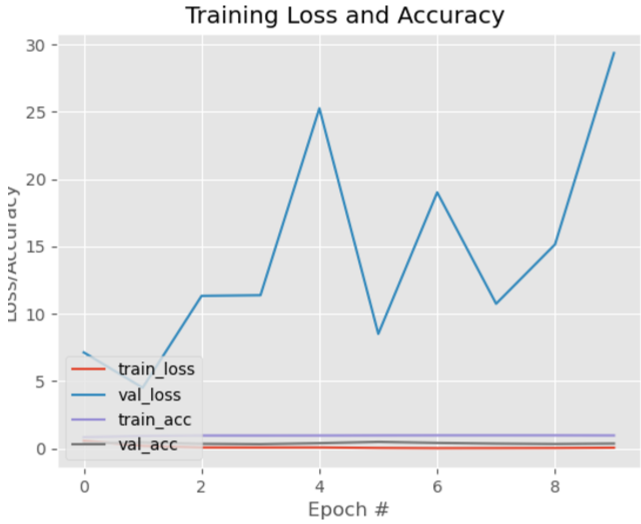

Figure 6.4: Simple VGG

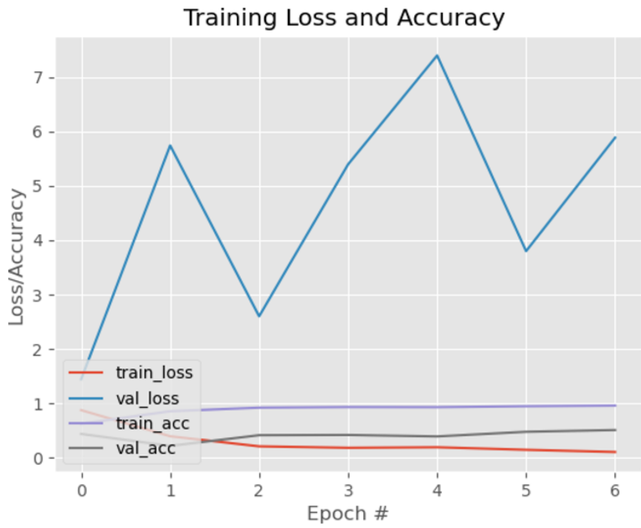

Figure 6.5: VGG with increased dropout rate

Figure 6.6: VGG with increased dropout rate and l2 regularization

6.7.2.3 Resnet50 Results

At first simplified version of Resnet50 was trained on dataset which gave and accuracy of 0.79 on validation set then in order to reduce gap between training loss and validation loss dropout rate was tuned to 0.9 and batch normalization was added.

The table given below is for unseen male faces:

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Simple Resnet	0.99	0.79	0.86
ResNet with Increased dropout rate 0.9 in second last dense layer	0.99	0.84	0.89
With batch normalization	0.99	0.80	0.90

Table 6.8: AlexNet Results

Figure 6.7: simplest ResNet50

Figure 6.8: Resnet with increased dropout rate

Figure 6.9: Resent with increased dropout rate and batch normalization

6.7.3 Comparison

In this section comparison will be made between our selected model and Ning, Wanshan et al [4] since both are trained on dataset which are from same distribution. Author achieved and AUC value of 0.978, 0.921 and 0.931 for detecting control, Type 1 and Type 2 patients respectively whereas our system achieved an overall accuracy of 90 percent on testing data. Although authors accuracy is higher than ours, but author is using a hybrid system which is not only dependent on CT-Scan image data but also relying on clinical feature data such as blood test report whereas our system make prediction solely based on CT-scan image data. Hence in terms of simplicity our system is superior as compared to authors proposed system.

Chapter 7

Conclusions

We developed a system which preform diagnosis of Novel Virus COVID-19 by classifying give image into Control (negative), Regular or Severe class. We developed a web application which allow user to send CT-scan image of patient to server where after applying preprocessing a trained CNN model classify given image into appropriate class and result were provided to user on his device. Since we are digitalizing process of CT-scan base diagnosis we also store user data into database. This information was used later to generate useful stats which indicated usefulness of digitalization of old process. During the course of this project, we found that dataset on which we were originally working on had a serious issue. It had multiple images from same patient and after shuffling and splitting similar images lays in training testing and validation set so we composed our own dataset patient wise to solve this issue. We trained well know architecture such as AlexNet, VGG16 and Resent50 and modified and tune them to extract best out of them and to achieve high accuracy. Out of all trained model Resnet50 with increased dropout and batch normalization gave highest accuracy.

7.1 Future Work

We carried out this project during the pandemic of Novel Virus COVID-19. We were limited in terms of computational resource. All of the training preformed during this project was done on mobile version of RTX 2060 GPU with 6gb of ram this limit us to pick 80 patient per class at once. Adding more data could have aid in overcoming problem of overfitting and thus improve system accuracy on testing data but we could not add more data due to limited resources. Secondly a lot of our valuable time was wasted on original dataset and to trouble shoot problem with it hence due to limitation of time we only tried few architectures and tune them to get higher accuracy. More effort and time could be spent to improve tuning of hypermeters and to improve architecture of convolutional neural network to get higher accuracy. We used store data from database to generate some stats that could also be improve too. Principle of data science could be applied on this data to generate more useful information and to make future predictions.

Appendix A

User Manual

Introduction

The user manual is a guide document to help a new user get used to the application for the first use. This document will be a help and guide to user to use the app appropriately.

A.1 Start Web Application

This is the home page of web application. Core functionality which is to conduct COVID-19 diagnosis is placed in middle. User is supposed to enter his data at this page. Menu of web application is placed at top right corner. Home page is shown in Figure **??**.

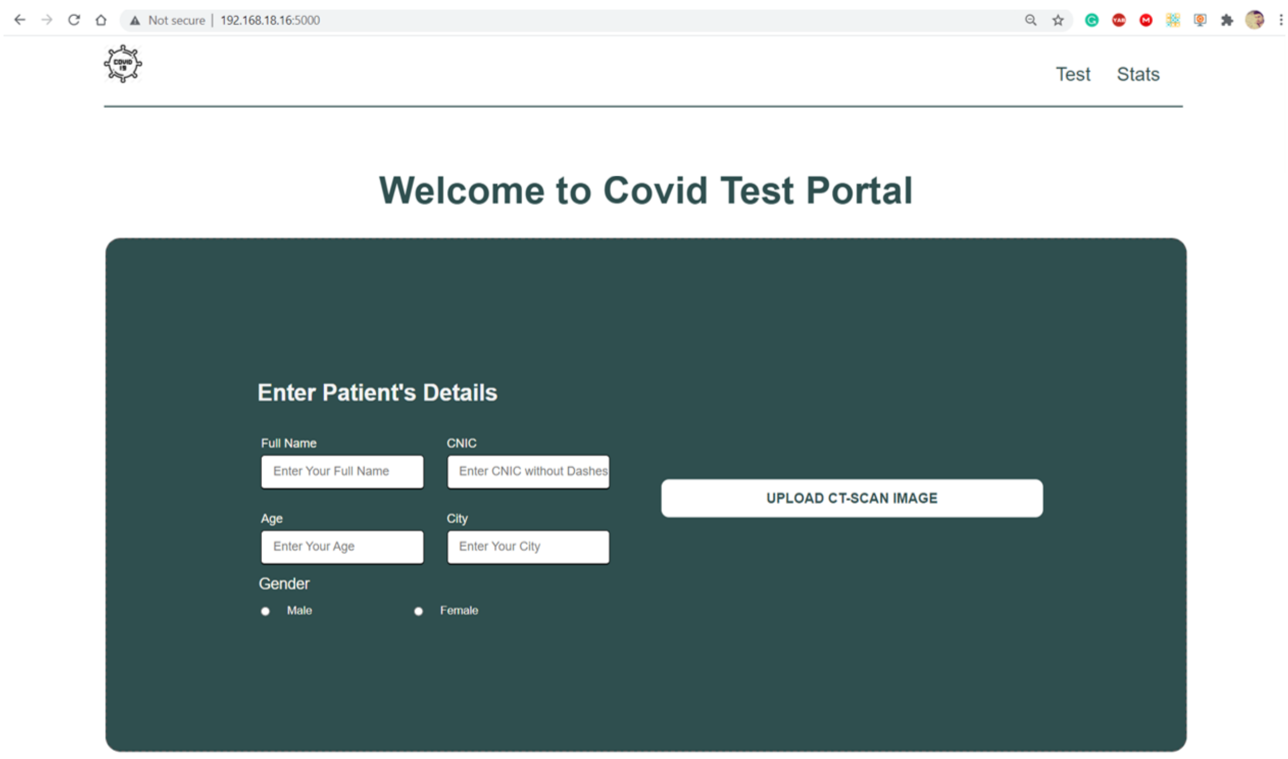

Figure A.1: Start Application.

A.2 Enter Details

At first patient will enter his/her details which include name, CNIC, Age, City and gender as shown in Figure A.2

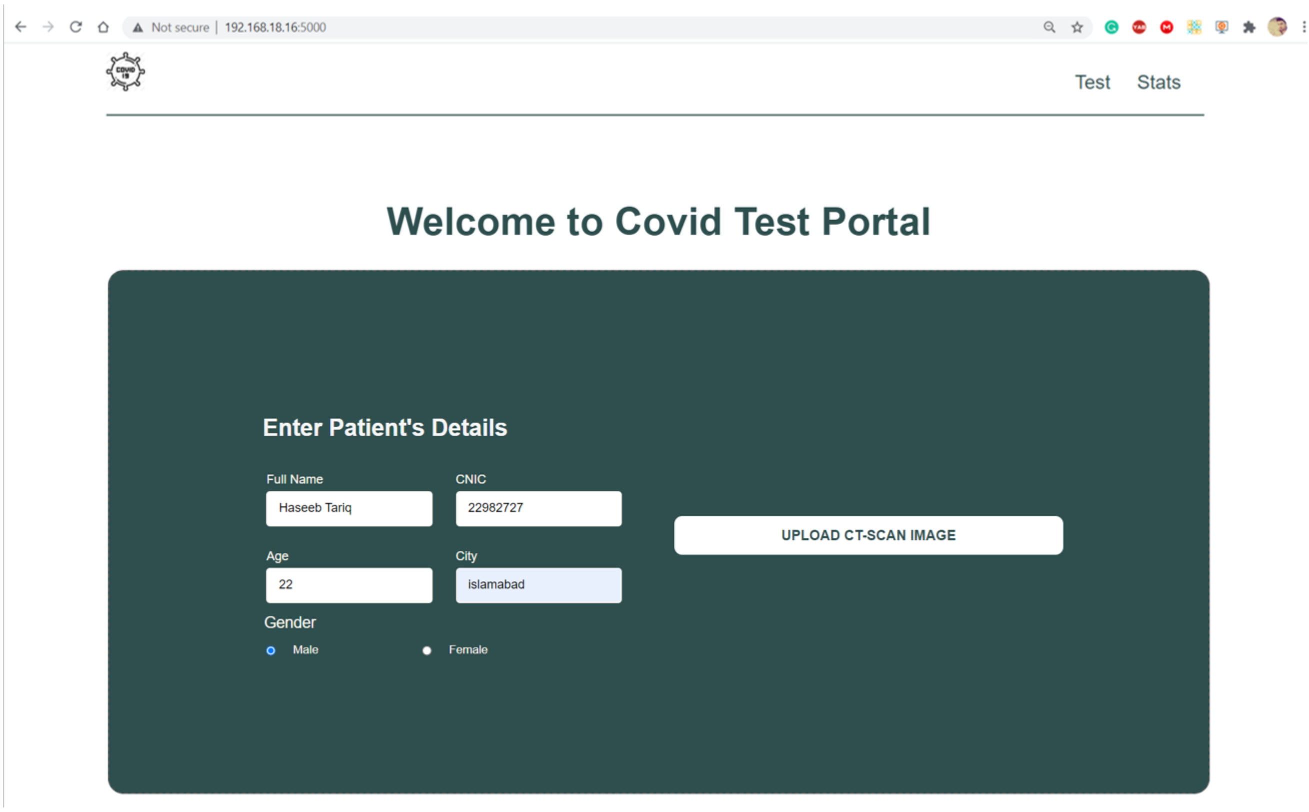

Figure A.2: Entering patient details.

A.3 upload CT-scan image

in Second step patient will upload CT-scan image from device files. once image is upload a preview will be displayed to user. And "Start Test" button will appear on screen as shown in Figure A.3

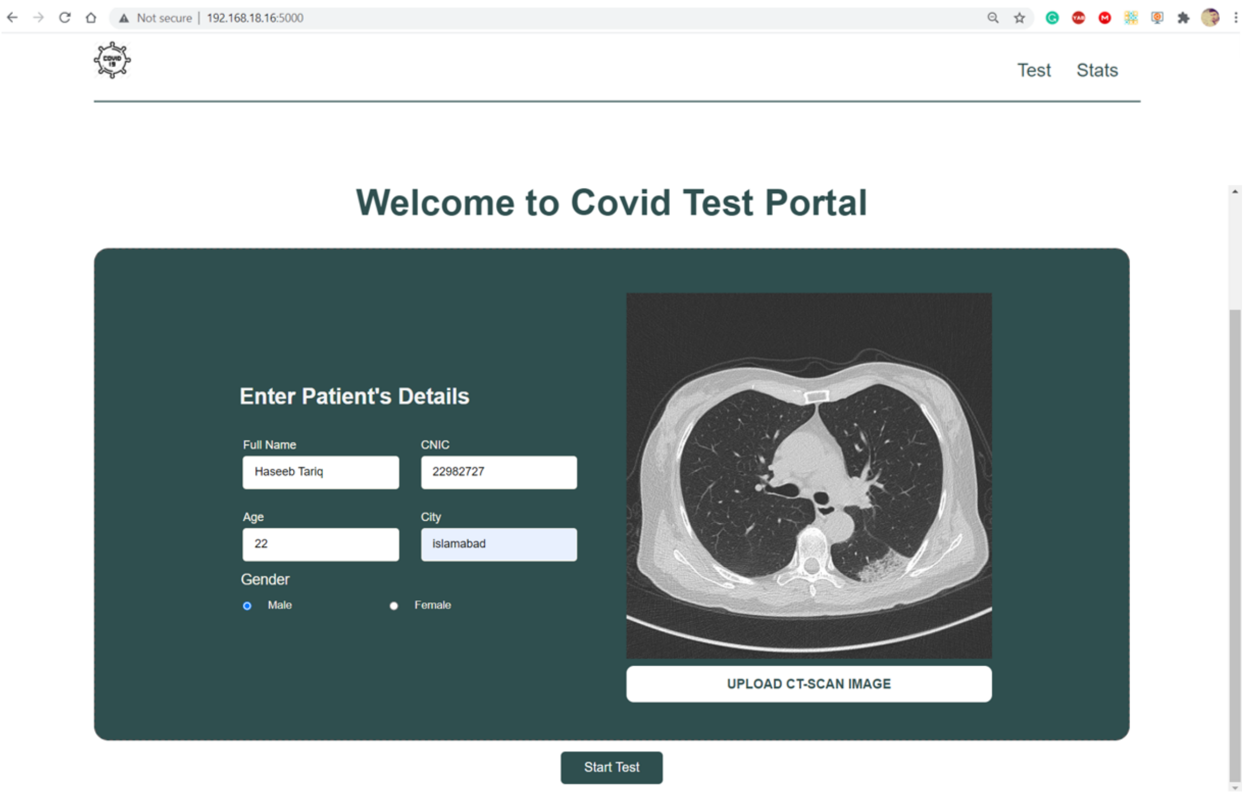

Figure A.3: uploading CT-scan image.

A.4 Error on fields

if user enters data which is not according to constrains a validation error will come up to notify user to correct input as shown in Figure A.4.

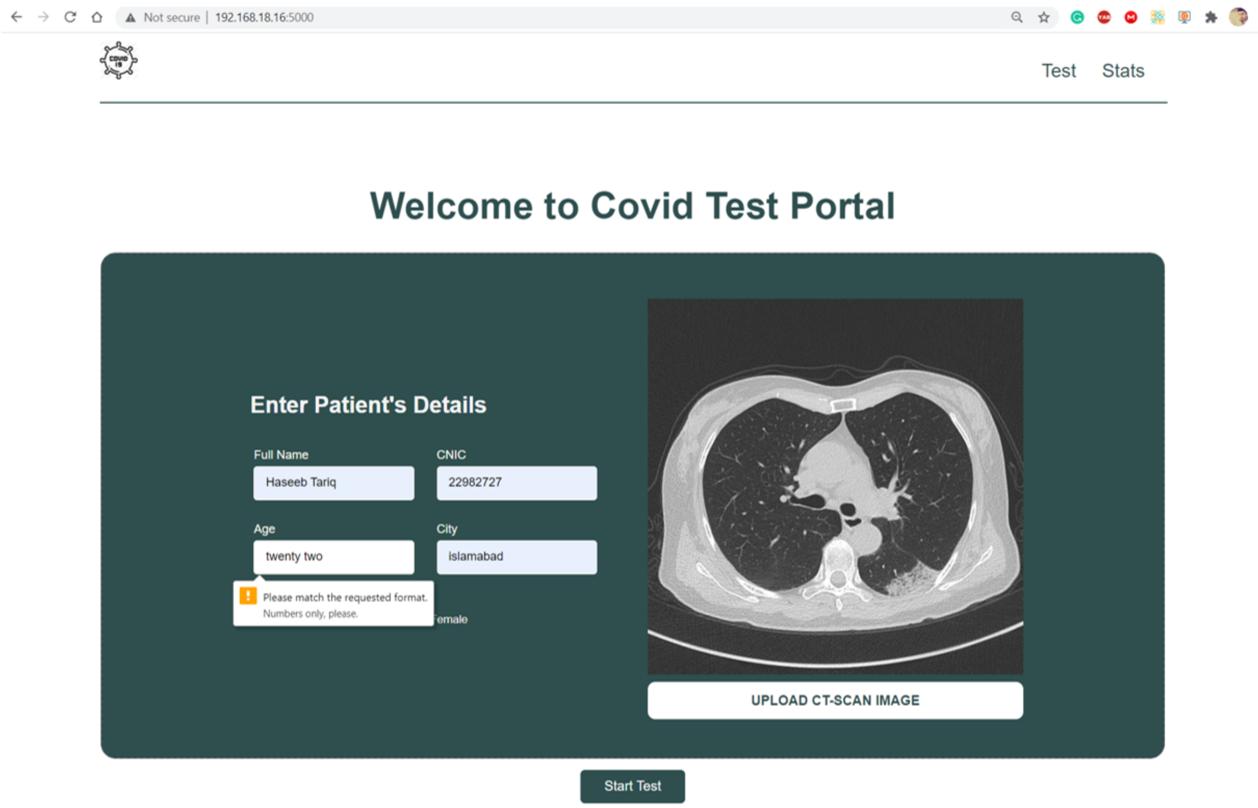

Figure A.4: Validation error on fields.

A.5 Processing State

Once "start Test" button is pressed data will be send to server side and processing will start. During this time proper feedback will be displayed to user through processing animation as shown in Figure A.5.

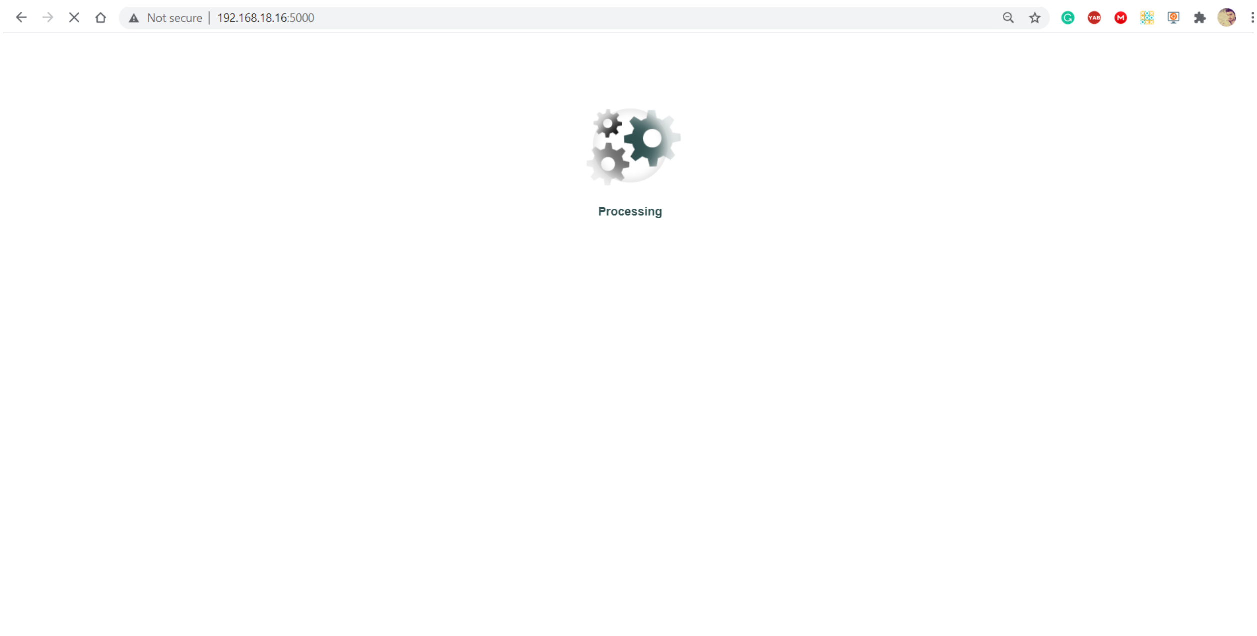

Figure A.5: Processing animation.

A.6 Receiving Report

Once processing is complete user is presented with diagnosis report as shown in Figure A.6

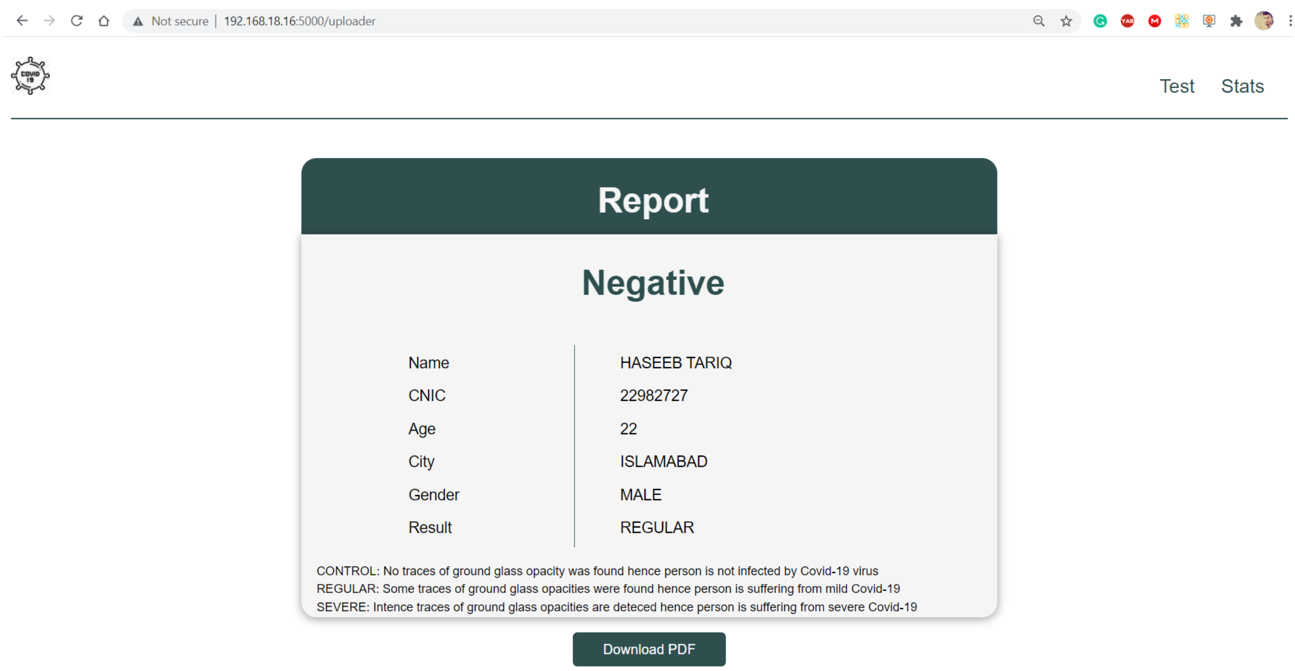

Figure A.6: Diagnosis report page.

A.7 Download Report

User can save report by clicking "Download PDF" button and selecting location. Downloaded report is shown in Figure A.7

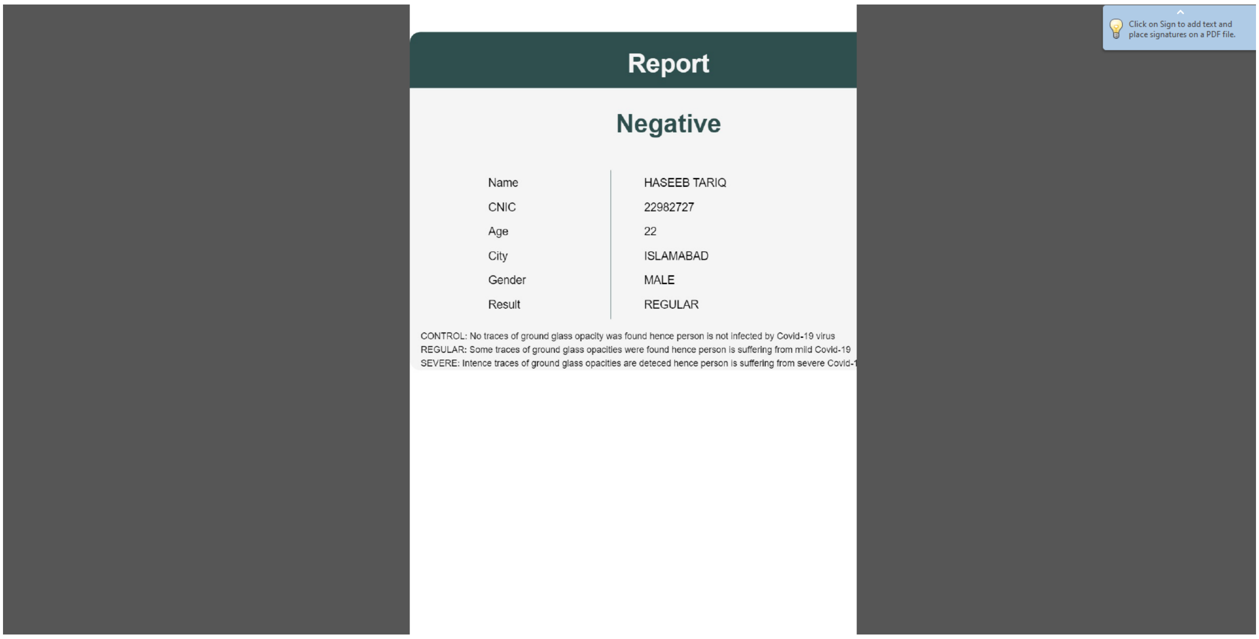

Figure A.7: Downloaded report.

A.8 Opening Stats Page

User can see stats based on data collect overtime by clicking on "Stats" menu. Stats page is shown in Figure A.8.

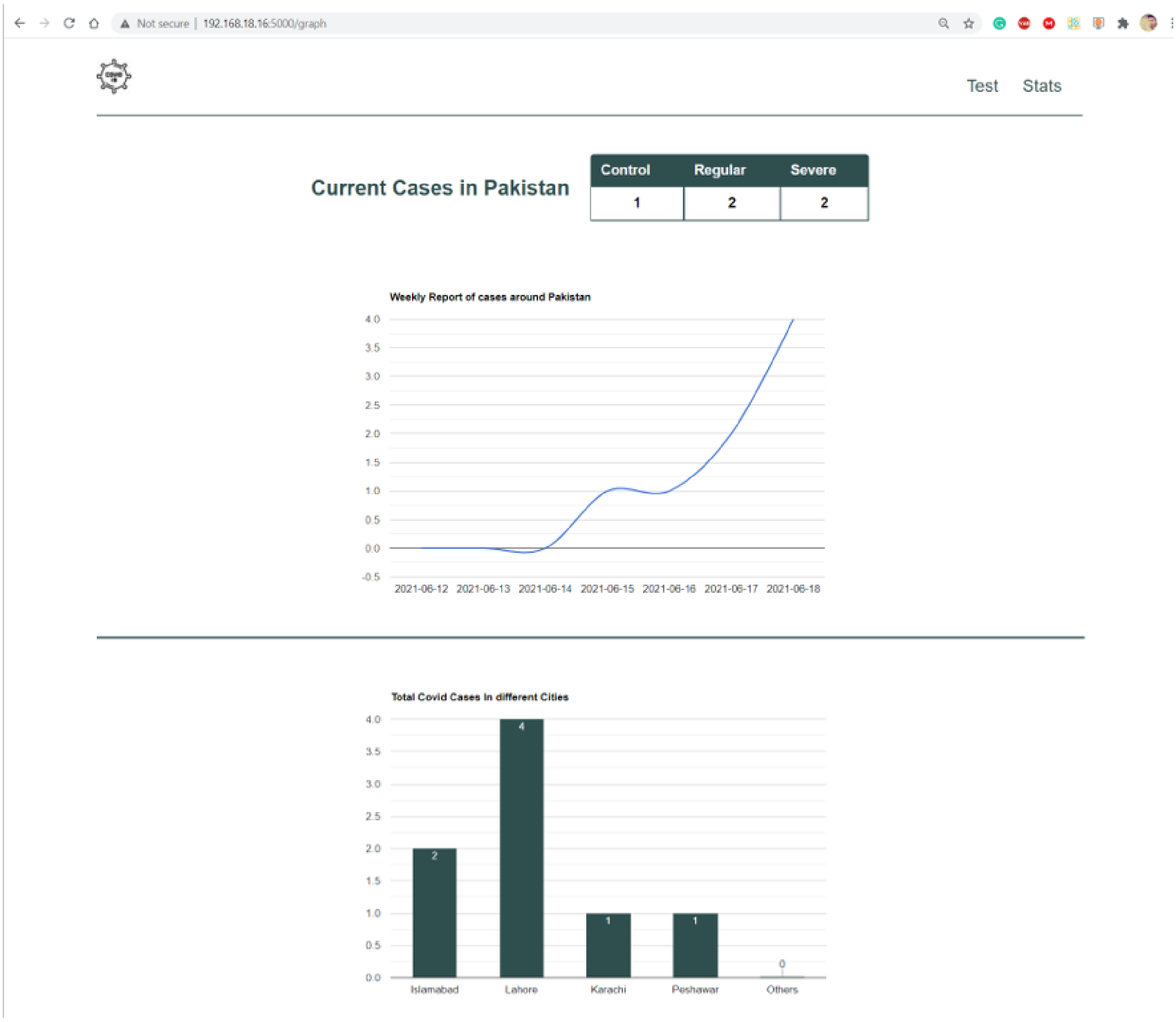

Figure A.8: Stats page.

A.9 Database Operation Error

In case Server is unable to extract data from database to generate stats. An Error message will be displayed to user as shown in Figure A.9.

Figure A.9: Database operation failure error.

Bibliography

[1] Wanshan Ning, Shijun Lei, Jingjing Yang, Yukun Cao, Peiran Jiang, Qianqian Yang, Jiao Zhang, Xiaobei Wang, Fenghua Chen, Zhi Geng, et al. ictcf: an integrative resource of chest computed tomography images and clinical features of patients with covid-19 pneumonia. 2020. Cited on pp. 4, 5, and 11.

[2] Y Song, S Zheng, L Li, X Zhang, X Zhang, Z Huang, J Chen, H Zhao, Y Jie, R Wang, et al. Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. medrxiv. *Preprint posted online February*, 25, 2020. Cited on p. 6.

[3] Md Zabirul Islam, Md Milon Islam, and Amanullah Asraf. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Informatics in medicine unlocked*, 20:100412, 2020. Cited on pp. 6 and 7.

[4] Vruddhi Shah, Rinkal Keniya, Akanksha Shridharani, Manav Punjabi, Jainam Shah, and Ninad Mehendale. Diagnosis of covid-19 using ct scan images and deep learning techniques. *Emergency radiology*, pages 1–9, 2021. Cited on pp. 8 and 11.

[5] Chuansheng Zheng, Xianbo Deng, Qing Fu, Qiang Zhou, Jiapei Feng, Hui Ma, Wenyu Liu, and Xinggang Wang. Deep learning-based detection for covid-19 from chest ct using weak label. *MedRxiv*, 2020. Cited on p. 9.

[6] Ilker Ozsahin, Boran Sekeroglu, Musa Sani Musa, Mubarak Taiwo Mustapha, and Dilber Uzun Ozsahin. Review on diagnosis of covid-19 from chest ct images using artificial intelligence. *Computational and Mathematical Methods in Medicine*, 2020, 2020. Cited on p. 1.

[7] Marina Carotti, Fausto Salaffi, Piercarlo Sarzi-Puttini, Andrea Agostini, Alessandra Borgheresi, Davide Minorati, Massimo Galli, Daniela Marotto, and Andrea Giovagnoni. Chest ct features of coronavirus disease 2019 (covid-19) pneumonia: key points for radiologists. *La radiologia medica*, 125:636–646, 2020. Cited on p. 1.

[8] Mohammad Rahimzadeh, Abolfazl Attar, and Seyed Mohammad Sakhaei. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. *Biomedical Signal Processing and Control*, page 102588, 2021. Cited on pp. 5 and 11.

[9] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *MedRxiv*, 2020. Cited on p. 8.

References

[1] Wanshan Ning, Shijun Lei, Jingjing Yang, Yukun Cao, Peiran Jiang, Qianqian Yang, Jiao Zhang, Xiaobei Wang, Fenghua Chen, Zhi Geng, et al. ictcf: an integrative resource of chest computed tomography images and clinical features of patients with covid-19 pneumonia. 2020. Cited on pp. 4, 5, and 11.

[2] Y Song, S Zheng, L Li, X Zhang, X Zhang, Z Huang, J Chen, H Zhao, Y Jie, R Wang, et al. Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images. medrxiv. *Preprint posted online February*, 25, 2020. Cited on p. 6.

[3] Md Zabirul Islam, Md Milon Islam, and Amanullah Asraf. A combined deep cnn-lstm network for the detection of novel coronavirus (covid-19) using x-ray images. *Informatics in medicine unlocked*, 20:100412, 2020. Cited on pp. 6 and 7.

[4] Vruddhi Shah, Rinkal Keniya, Akanksha Shridharani, Manav Punjabi, Jainam Shah, and Ninad Mehendale. Diagnosis of covid-19 using ct scan images and deep learning techniques. *Emergency radiology*, pages 1–9, 2021. Cited on pp. 8 and 11.

[5] Chuansheng Zheng, Xianbo Deng, Qing Fu, Qiang Zhou, Jiapei Feng, Hui Ma, Wenyu Liu, and Xinggang Wang. Deep learning-based detection for covid-19 from chest ct using weak label. *MedRxiv*, 2020. Cited on p. 9.

[6] Ilker Ozsahin, Boran Sekeroglu, Musa Sani Musa, Mubarak Taiwo Mustapha, and Dilber Uzun Ozsahin. Review on diagnosis of covid-19 from chest ct images using artificial intelligence. *Computational and Mathematical Methods in Medicine*, 2020, 2020. Cited on p. 1.

[7] Marina Carotti, Fausto Salaffi, Piercarlo Sarzi-Puttini, Andrea Agostini, Alessandra Borgheresi, Davide Minorati, Massimo Galli, Daniela Marotto, and Andrea Giovagnoni. Chest ct features of coronavirus disease 2019 (covid-19) pneumonia: key points for radiologists. *La radiologia medica*, 125:636–646, 2020. Cited on p. 1.

[8] Mohammad Rahimzadeh, Abolfazl Attar, and Seyed Mohammad Sakhaei. A fully automated deep learning-based network for detecting covid-19 from a new and large lung ct scan dataset. *Biomedical Signal Processing and Control*, page 102588, 2021. Cited on pp. 5 and 11.

[9] Xuehai He, Xingyi Yang, Shanghang Zhang, Jinyu Zhao, Yichen Zhang, Eric Xing, and Pengtao Xie. Sample-efficient deep learning for covid-19 diagnosis based on ct scans. *MedRxiv*, 2020. Cited on p. 8.