University of Mumbai

MedX - Covid-19 & Lung Disease Detection Portal

Submitted at the end of semester VIII in partial fulfilment of requirements for the degree of

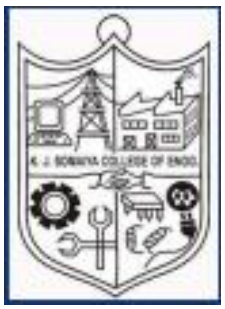
Bachelors in Technology

by

Mayank Ahuja
Roll No: 1714002
Deep Chothani
Roll No: 1714011
Udit Dayani
Roll No: 1714012

Guide

Prof. Era Johri



Department of Information Technology
K. J. Somaiya College of Engineering, Mumbai-77

Batch 2017-21

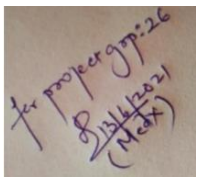
(Autonomous College Affiliated to University Of Mumbai)

K. J. Somaiya College of Engineering, Mumbai-77
Batch 2017-21

(Autonomous College Affiliated to University Of Mumbai)

Certificate

This is to certify that the dissertation report entitled MedX - Covid-19 & Lung Disease Detection Portal submitted by Mayank Ahuja under the guidance of Prof Era Johri at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfilment of requirements for the degree of Bachelors in Information Technology of University of Mumbai



Guide

Head of the Department

Principal

Date:

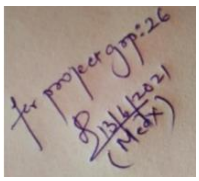
Place: Mumbai 77

K. J. Somaiya College of Engineering, Mumbai-77
Batch 2017-21

(Autonomous College Affiliated to University Of Mumbai)

Certificate

This is to certify that the dissertation report entitled MedX - Covid-19 & Lung Disease Detection Portal submitted by Deep Chothani under the guidance of Prof Era Johri at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfilment of requirements for the degree of Bachelors in Information Technology of University of Mumbai



Guide

Head of the Department

Principal

Date:

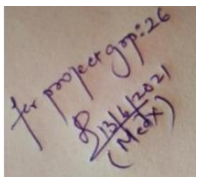
Place: Mumbai 77

K. J. Somaiya College of Engineering, Mumbai-77
Batch 2017-21

(Autonomous College Affiliated to University Of Mumbai)

Certificate

This is to certify that the dissertation report entitled MedX - Covid-19 & Lung Disease Detection Portal submitted by Udit Dayani under the guidance of Prof Era Johri at the end of semester VIII of LY B. Tech is a bona fide record for partial fulfilment of requirements for the degree of Bachelors in Information Technology of University of Mumbai



Guide

Head of the Department

Principal

Date:

Place: Mumbai 77

K.J. Somaiya College of Engineering,
Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **MedX – Covid-19 & Other Lung Disease Detection Portal** is bona fide record of project work done by **Mayank Ahuja**.

This project is approved for the award of Bachelors in Technology Degree in Information Technology of University of Mumbai.

Internal Examiner

External Examiner

Date: 27.05.2021

Place: Mumbai-77

K.J. Somaiya College of Engineering,
Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **MedX – Covid-19 & Other Lung Disease Detection Portal** is bona fide record of project work done by **Deep Chothani**.

This project is approved for the award of Bachelors in Technology Degree in Information Techology of University of Mumbai.

13/6/2021
MedX
BE Project
Exam Evaluation

Internal Examiner

External Examiner

Date:27.05.2021

Place: Mumbai-77

K.J. Somaiya College of Engineering,
Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **MedX – Covid-19 & Other Lung Disease Detection Portal** is bona fide record of project work done by **Udit Dayani**.

This project is approved for the award of Bachelors in Technology Degree in Information Technology of University of Mumbai.

Internal Examiner

External Examiner

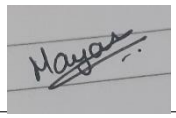
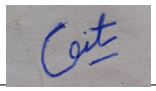
Date: 27.05.2021

Place: Mumbai-77

Declaration

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea / data / fact / source / original work / matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

 Signature of the student 1714002 Roll no.	 Signature of the student 1714011 Roll no.
 Signature of the student 1714012 Roll no.	

Date: 27.05.2021

Place: Mumbai 77

Abstract

Spread of a deadly and highly contagious virus named COVID-19 has become conspicuous in the last couple of years. This disease has opened a new arena for scientists and researchers. A new challenge has entered the society to diagnose the disease at a very fast pace along with accuracy to help the medical practitioners. The superannuated approach taken in the medical history is the lung X- RAY. The amelioration in the image processing using convolution neural networks will help us to identify the damage caused by any such diseases which can be detected using chest X-rays. In this paper, we start developing convolutional neural networks of various architectures and compare their validation accuracy, validation loss, train accuracy and test loss over a determined number of epochs by using chest X-ray reports. The architecture with the best validation accuracy along with a good train accuracy can be used for the detection of COVID-19. Postulation like transfer learning will be implemented in our future models to increase the accuracy of the model and heat maps to find the affected region.

Therefore, our project, a web application that will detect whether a person is infected by Corona virus or by Lung Cancer or is suffering from Pneumonia. The web application would need the users to just upload a chest X-ray on the portal and then the system will carry out the test on the image using Convolution Neural Network (CNN) models, then display the in-depth report in a single click.

Contents

List of Figures

List of Tables

1 Introduction

This chapter contains a brief introduction for the project. It states the problem definition, scope, motivation, salient contribution and organization of black-book. It gives an overview about the reasons and motivations that were pivotal in selection of this project and the related background work for the same.

1.1 Problem Definition

To check whether the patient is infected from Covid-19 or not, by taking the X-ray reports and analyzing with previous data records. Our product will cater different types of lung disease viz. covid-19, pneumonia, lung cancer. To use Convolution Neural Networks to classify the X-ray image and identify whether the patient is suffering from any such disease or not.

1.2 Motivation of Thesis

Although there are different ways for conducting Covid-19 or Pneumonia or lung cancer tests, but there are certain difficulties and drawbacks associated with each one of them:

Detection of Covid19 through blood tests is costly.

Analysing Blood Tests take time as compared to the proposed system where the results are generated within seconds after uploading the report.

To detect the extent of Spread using this instantly.

Damage of lungs cannot be detected through blood tests.

The person has to get his X-ray done for damage to the lungs but the ratio of doctor to patient is not optimal therefore there may be certain cases in critical condition left.

1.3 Scope of the thesis

Functional Requirements:

i). A Graphic User Interface for uploading the X-ray Image and entering the patient details to enhance the user experience.

ii). A SignUp page for the user to create an account on the web application.

iii). A SignIn page for the user to sign into their created account.

iv). A home page which is the index page which leads to other pages like the upload page, SignUp page and SignIn page.

v.) Covid-19, Pneumonia, Lung Cancer web-pages containing upload form along with FAQs.

Non-Functional Requirements:

i.)Usability: We will be providing a Graphic User Interface for the user to upload their X-ray images. The first GUI will be made with tkinter, and as the database will grow eventually we'll move to Django or Flask.

ii.) Security: Security infrastructure of GCP which we will use for running the model will be used. In case if GCP is unable to handle the model then we will be having our own localhost security. The website will be 'https' website.

iii.) Performance: The result of whether the person has pneumonia or not or whether the person has covid 19 or not should be displayed within 2 seconds of clicking the button 'Check for disease".

iv.) Maintainability: We will be adopting an agile model. We will have a 24 hrs support team for easy maintenance.

v.) Error handling: The software will undergo a comprehensive testing procedure to make it error free. We will first perform Unit Test on each module, then followed by Integration Testing and then Big Bang testing. Then we will perform Usability Testing and Acceptance Testing on the UI. Different methods of white box testing will be conducted in the code structure.

vi.) Ease of use: The website will be easily navigable with proper text and annotations. We will be providing a Graphic User Interface for the user to upload his or her images.

1.4 Salient Contribution

This project is focused towards the development of a highly efficient model for detection of COVID-19 or other lung diseases which will be helpful for the users to get an instant report of diagnosis. The project end product is highly accurate and quick in generating reports from the uploaded X-ray image.

1.5 Organization of the thesis

1.Introduction: This chapter contains a brief introduction for the project. It states the problem definition, scope, motivation, salient contribution and organization of black book. It gives an overview about the reasons and motivations that were pivotal in selection of this project and the related background work for the same.

2.Literature Survey: This chapter consists of all the relevant research papers that helped us define the base of this project. Detailed description of paper and methodology is explained in this chapter.

3.Software Project Management Plan: This is a detailed Project Plan of our topic. This chapter will provide a detailed explanation about the project goals and a plan to achieve the mentioned goals. The software project management plan comprises four sections namely the introduction, the project organization, roles and responsibilities, the management plan and concluded by the timetable.

4.Software Requirement Specification: This chapter is a Software Requirement Specification for our Application. The purpose of this document is to collect and analyse all assorted ideas that have come up to define the proposed system. This document provides a detailed overview of functional and non-functional requirements of the system. It defines how our team views the system
and its functionality.

5.Software Design Document: This chapter is a Design Document for our Application. The architecture of the system is defined and the hardware and software components are specified. Data flow through components are explained using data flow diagrams.

6. Software Test Document: This chapter provides Software Test Document for our Application. The document will explain the test approach, test plan and the test cases.

2 Literature Survey

This chapter presents the models and the methodology which have been built to detect the diseases using X-ray. Multiple models have been listed along with their accuracy and neural network structure, where every model has its own pros and cons. Various drawbacks have been identified which we are going to overcome in this project to be built.

2.1 Literature Review

Analysing X Ray reports using Convolution Neural Networks has been used for several decades to explore and visualize fractures or abnormalities in body organs and diagnose sensitive parts like bones, chest, teeth, skull etc. X ray images can be analysed by various methods like multilayer probabilistic method, learning vector quantization and generalized regression networks.

These are better in classification but less efficient in terms of accuracy and time of computation. Since both time and accuracy is important for detection of Corona, we use a deep learning model based on CNN architecture that has the power of extracting different level features from the images.

1. COVID-ResNet:

A Deep Learning Framework for Screening of COVID19 from Radiographs. Muhammad Farooq1 and Abdul Hafeez2 In this research paper they have used ResNet50 to detect whether the person has covid19 pneumonia or not. There is a balance between the performance and the speed of training it. Also it has attempted to increase the accuracy to about ninety percent. [1]

2. Covid-19:

Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. They in their research paper have used transfer learning techniques to train their neural network. Various neural network like VGG19 , Inception and Xception were used.[2]

3. Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning:

In this research paper they have used linear regression in addition to deep neural networks to predict the severity of the COVID 19 disease. They have used the DenseNet deep learning model to achieve this. They have used Geographic extent and Opacity score as the metrics for accuracy.[3]

4. COVID-Net:

A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images. In this research paper the researchers have calculated the efficiency and sensitivity of a novel neural network which was created by Darwin.ai. This neural net is named covid-net and the dataset which it works on is Covidx. Although in the research paper they said it is not yet ready for industrial use.[4]

2.2 Conclusion

Various CNN models and methods were examined which provides various accuracy and methodologies. This chapter represents information about diseases which can be detected using neural networks and different models with their accuracy. Each methodology has few drawbacks which need to be overcome. The next chapter gives us a brief description about the project management plan. It provides us with details about the goals to be achieved, how roles and responsibilities are being divided, what tools and techniques are going to be used for the project. It also provides information about the schedule of the project.

3 Software Project Management Plan

This chapter gives us a brief description about the project management plan. It provides us with details about the goals to be achieved, how roles and responsibilities are being divided, what tools and techniques are going to be used for the project. It also provides information about the schedule of the project.

3.1 Introduction

3.1.1 Project Overview

Project will check whether the patient is infected from covid-19 or not, by taking the x-ray reports, uploading it on the user interface into the trained model and getting the report on that UI. System will also check if the patient is free from covid, but still shares some similar report then it will check for pneumonia and lung cancer, and will give the outcome accurately.

3.1.2 Project Deliverables

i.) UI code done with the help of Reactjs, HTML, CSS, Chart.js, MaterialUI

ii.) .py file for the trained model of pneumonia dataset

iii.) .py file for the trained model of covid dataset

iv.) .py file for the trained model of lung cancer dataset

v.) Dataset of .jpg files for training and testing of pneumonia

vi.) Dataset of .jpg files for training and testing of covid-19

vii.) Dataset of .jpg files for training and testing of lung cancer.

3.2 Project Organization

3.2.1 Software Process Model

Deep learning projects are highly iterative; as you progress through the DL lifecycle, we will find ourselves iterating on a section until reaching a satisfactory level of performance, then proceeding forward to the next task (which may be circling back to an even earlier step). Moreover, a project isn't complete after we ship the first version; you get feedback from real-world interactions and redefine the goals for the next iteration of deployment.

1. **Planning and project setup**
    - Define the task and scope out requirements
    - Determine project feasibility
    - Discuss general model tradeoffs (accuracy vs speed)
    - Set up project codebase

2. **Data collection and labelling**
    - Define ground truth (create labelling documentation)
    - Build data ingestion pipeline
    - Validate quality of data
    - Revisit Step 1 and ensure data is sufficient for the task

3. **Model exploration**
    - Establish baselines for model performance
    - Start with a simple model using initial data pipeline
    - Overfit simple model to training data
    - Stay nimble and try many parallel (isolated) ideas during early stages
    - Find SoTA model for your problem domain (if available) and reproduce results, then apply to your dataset as a second baseline
    - Revisit Step 1 and ensure feasibility
    - Revisit Step 2 and ensure data quality is sufficient

4. **Model refinement**
    - Perform model-specific optimizations (ie. hyperparameter tuning)
    - Iteratively debug model as complexity is added
    - Perform error analysis to uncover common failure modes
    - Revisit Step 2 for targeted data collection of observed failures

5. **Testing and evaluation**
    - Evaluate model on test distribution; understand differences between train and test set distributions (how is “data in the wild” different than what you trained on)
    - Revisit model evaluation metric; ensure that this metric drives desirable downstream user behaviour
    - Write tests for:
        - Input data pipeline
        - Model inference functionality
        - Model inference performance on validation data
        - Explicit scenarios expected in production (model is evaluated on a curated set of observations)

6. **Model deployment**
    - Expose model via a REST API
    - Deploy new model to small subset of users to ensure everything goes smoothly, then roll out to all users
    - Maintain the ability to roll back model to previous versions
    - Monitor live data and model prediction distributions

7. **Ongoing model maintenance**
    - Understand that changes can affect the system in unexpected ways
    - Periodically retrain model to prevent model staleness
    - If there is a transfer in model ownership, educate the new team

3.2.2 Roles and Responsibilities

- **Data Engineer** (builds the data ingestion pipelines) Deep Chothani, Mayank Ahuja
- **Machine Learning Engineer** (train and iterate models to perform the task) Mayank Ahuja, Udit Dayani.
- **Software Engineer** (aids with integrating machine learning model with the rest of the product) Deep Chotani
- **Project Manager** (main point of contact with the client) Mayank Ahuja

3.2.3 Tools and Techniques

- ❏ CNN which will be used for Covid19: AlexNet, ResNet, Google Net, VGG16 Network etc.
- ❏ Libraries which we will be using: ScikitLearn, TensorFlow, Keras.
- ❏ Programming Language: Python
- ❏ Open Source: GitHub and Kaggle.

3.3 Project Management Plan

3.4.1 Tasks

3.4.1.1 Task-1 : Requirement gathering and analysis

3.4.1.1.1 Description
In this task we will be gathering and analysing the requirements by the literature review of the pre-existing papers and having a conversation with the supervisor.

3.4.1.1.2 Deliverables and Milestones
Milestone 1: Delivery of the presentation in the .pptx format.

3.4.1.1.3 Resources Needed

[1] COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs Muhammad Farooq1 and Abdul Hafeez2

[2] Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Ioannis D. Apostolopoulos Tzani A. Mpesiana

[3] Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning Joseph Paul Cohen Lan Dao , Paul Morrison , Karsten Roth , Yoshua Bengio , Beiyi Shen, MD Almas Abbasi, MD Mahsa Hoshmand-Kochi, MD Marzyeh Ghassemi , Haifang Li , Tim Q Duong .

[4] COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images Linda Wang , Zhong Qiu Lin , and Alexander Wong .

3.4.1.1.4 Dependencies and Constraints

Dependencies:
i).Have access to MS PowerPoint.
ii).Have access to pdf viewer to view the research paper.
iii).MS Word should be installed in the computer.

Constraints:
i).Have an internet browser
ii) Have access to the World Wide Web.

3.4.1.1.5 Risks and Contingencies

Risk: Absence of MS PowerPoint
Contingency: Use Google pptx

3.4.1.2 Task-2 : Data Collection

3.4.1.2.1 Description
In this task we will be collecting dataset for testing and training of our deep learning model

3.4.1.2.2 Deliverables and Milestones
Milestone having a well defined dataset of pneumonia, Covid-19 and normal X-rays.

3.4.1.2.3 Resources Needed

i). Covid positive samples(pprox. 142 samples)
ieee8023/covid-chestxray-dataset: We are building an open database of COVID-19 cases with chest X-ray or CT images. These contain the samples of covid-19 positive x ray images

ii). Normal Chest X rays(pprox. 142 samples)
https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.These contain the x ray images of normal cold(without covid-19) Identify the resources needed to execute the task (equipment, access to papers/books, etc.)

3.4.1.2.4 Dependencies and Constraints

Dependencies
i).Have an account on Kaggle
ii).Have an account on GitHub

Constraints
i).Git or Git Bash to be used.
ii).The dataset must be in jpeg format.
3.4.1.2.5 Risks and Contingencies
Risk: Data not in jpeg format.
Contingency: Use an image converter.

3.4.1.3 Task-3 : Project Initiation with Module 1

3.4.1.3.1 Description
In this task we will be initiating our final year B.Tech project by working on the
SRS, SDD, SPMP and STD and the first module of our project.

3.4.1.3.2 Deliverables and Milestones

Milestone 1: Delivery of the final revisions of the SRS, SDD, and SPMP;

milestone 2: Delivery of the STD;

milestone 3: Might complete version 1 (development & test) of detecting pneumonia in general.

3.4.1.3.3 Resources Needed

i). Covid positive samples(pprox. 142 samples)
ieee8023/covid-chestxray-dataset: We are building an open database of COVID-19 cases with chest X-ray or CT images. These contain the samples of covid-19 positive x ray images

ii). Normal Chest X rays(pprox. 142 samples)
https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia These contain the x ray images of normal cold(without covid-19)

iii). Research papers:

[1] COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs Muhammad Farooq1 and Abdul Hafeez2

[2] Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Ioannis D. Apostolopoulos Tzani A. Mpesiana

[3] Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning Joseph Paul Cohen Lan Dao , Paul Morrison , Karsten Roth , Yoshua Bengio , Beiyi Shen, MD Almas Abbasi, MD Mahsa Hoshmand-Kochi, MD Marzyeh Ghassemi , Haifang Li , Tim Q Duong .

[4] COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images Linda Wang , Zhong Qiu Lin , and Alexander Wong .

iv). Jupyter Notebook on a cloud platform.

3.4.1.3.4 Dependencies and Constraints

Dependencies
i).Have access to Jupyter notebook.
ii).Have access to online (cloud) computing platform.
iii).MS Word should be installed in the computer.

Constraints
i).Python programming language must be used
ii).The dataset must be in jpeg format.

3.4.1.3.5 Risks and Contingencies

Risk: Online (Cloud) platform doesn’t work
Contingency: Use the already installed Jupyter notebook on the computer.

3.4.2 Assignments

- **Data Engineer** (builds the data ingestion pipelines) Deep Chothani, Mayank Ahuja
- **Machine Learning Engineer** (train and iterate models to perform the task) Mayank Ahuja, Udit Dayani.
- **Software Engineer** (aids with integrating machine learning model with the rest of the product) Deep Chotani
- **Project Manager** (main point of contact with the client) Mayank Ahuja

3.4.3 Time Table

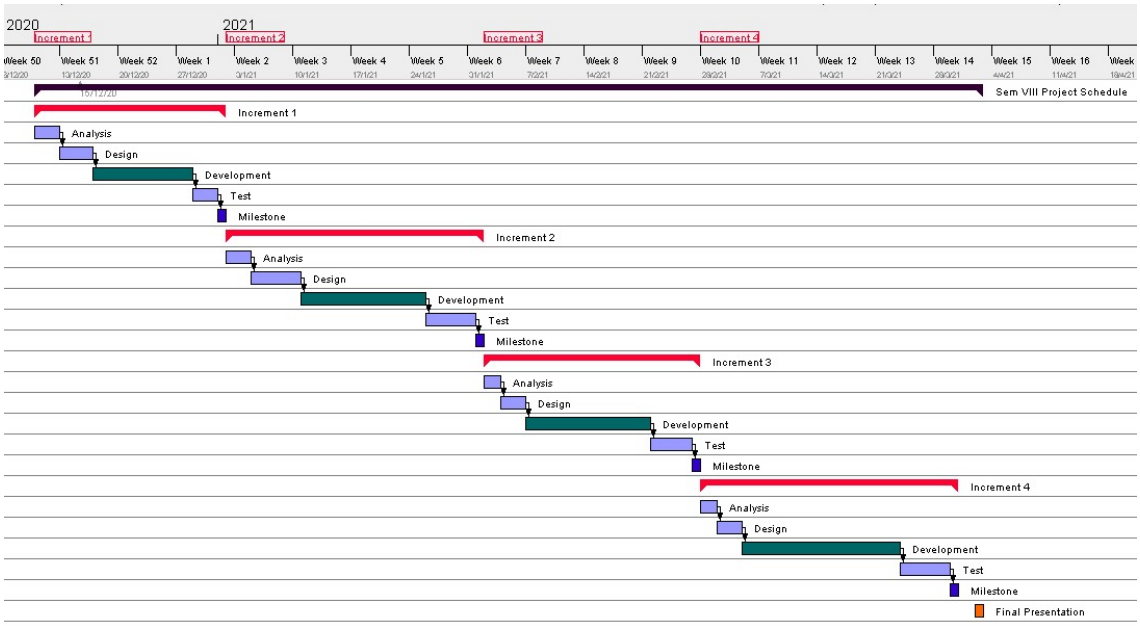


Figure 1: Sem VIII Gantt Chart

Name	Begin date	End date
Sem VIII Project Schedule	10/12/20	2/4/21
Increment 1	10/12/20	1/1/21
Analysis	10/12/20	12/12/20
Design	13/12/20	16/12/20
Development	17/12/20	28/12/20
Test	29/12/20	31/12/20
Milestone	1/1/21	1/1/21
Increment 2	2/1/21	1/2/21
Analysis	2/1/21	4/1/21
Design	5/1/21	10/1/21
Development	11/1/21	25/1/21
Test	26/1/21	31/1/21
Milestone	1/2/21	1/2/21
Increment 3	2/2/21	27/2/21
Analysis	2/2/21	3/2/21
Design	4/2/21	6/2/21
Development	7/2/21	21/2/21
Test	22/2/21	26/2/21
Milestone	27/2/21	27/2/21
Increment 4	28/2/21	30/3/21
Analysis	28/2/21	1/3/21
Design	2/3/21	4/3/21
Development	5/3/21	23/3/21
Test	24/3/21	29/3/21
Milestone	30/3/21	30/3/21
Final Presentation	2/4/21	2/4/21

Figure 2: Sem VIII Project Schedule

3.5 Conclusion

We will have a CNN that is a Convolutional Neural Network which will be one of the subsystems. The term CNN will be used for Convolutional Neural Network. Then we will have a UI which is a user interface. This UI stands for the web-based application where the user will upload the image.

4 Software Requirement Specification

The purpose of this chapter is to give a detailed description of the requirements for the “MedX” application. It will illustrate the purpose and complete declaration for the development of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be a reference for developing the first version of the system for the developer.

4.1 Introduction

4.1.1 Product Overview

To check whether the patient is infected from covid-19 or not, by taking the x-ray reports, uploading it on the user interface into the trained model and getting the report on that UI. System will also check if the patient is free from covid-19, but still shares some similar report then it will check for pneumonia, and will give the outcome accurately.

4.2 Specific Requirements

4.2.1 External Interface Requirements

4.2.1.1 User Interfaces:

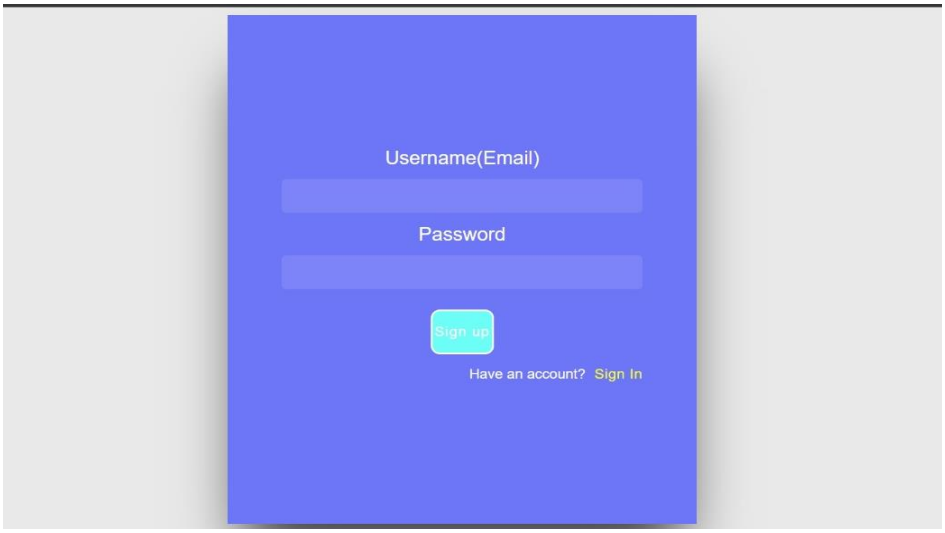


Figure 3: Sign Up Page

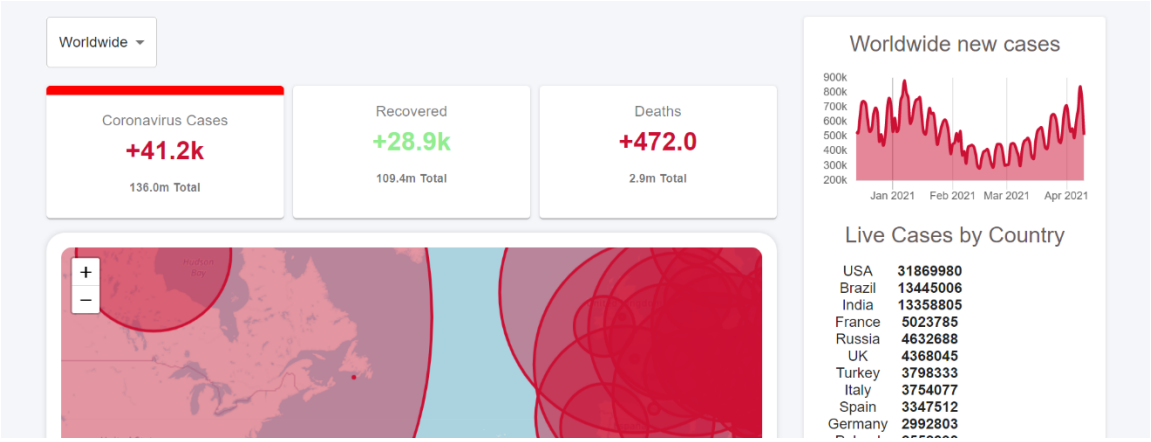


Figure 4: Covid-19 DashBoard

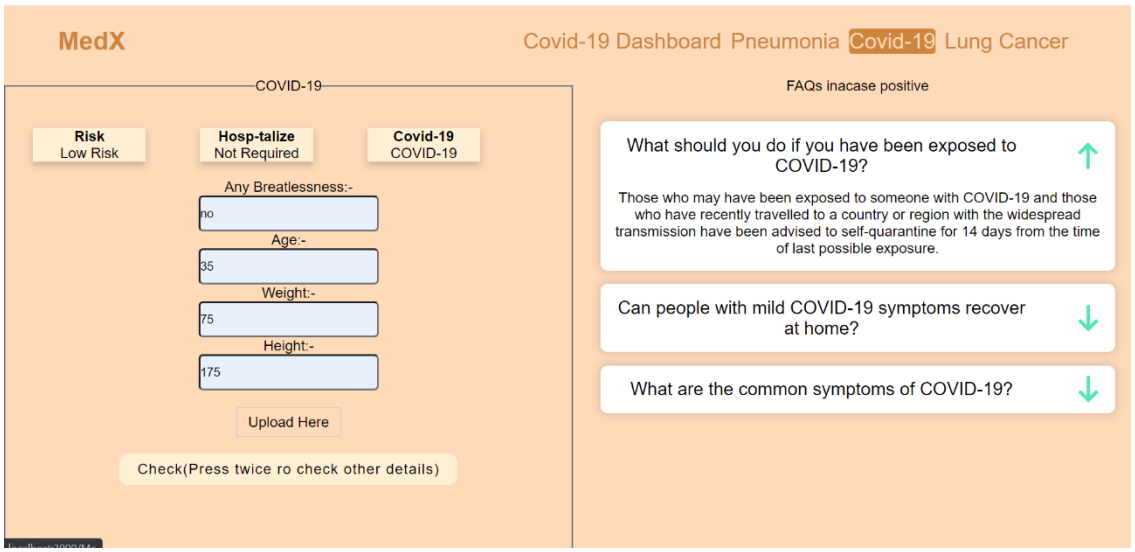


Figure 5: Covid-19 Webpage

4.2.1.2 Hardware Interfaces:

Mouse Drag and Drop of Image , Keyboard for typing various things in various fields of the UI. The backend Hardware services managed with the help of Python, Flask, TensorFlow, CNN.

4.2.1.3 Software Interfaces:

i. Language Used
Front End(User Interface)
ReactJS, HTML, Chart.js, MaterialUI

DataBase
FireBase Authentication, Amazon Web Services s3 Bucket

Backend
Python, Flask, CNN, TensorFlow

4.2.1.4 Communication Protocols:

HTTP, HTTPS

4.2.2 Software Product Features

i)Graphic User Interface for uploading the Xray Image and entering the patient details.

ii) GUI for covid-19 dashboard showing current covid-19 patient all over world and per country along with the recovered and new cases option.

iii.) FAQs on the diseases web-pages.

iv.) Covid-19, Pneumonia, Lung-Cancer web-pages having upload form and FAQ section.

v.) Sign-up and Sign-In page along with forgot password option.

4.2.3 Software System Attributes

(a) Usability: We will be providing a Graphic User Interface for the user to upload their X-ray images. The first GUI will be made with tkinter, and as the database will grow eventually we'll move to Django or Flask.

(b) Performance: The result of whether the person has pneumonia or not or whether the person has covid 19 or not should be displayed within 2 seconds of clicking the button 'Check for disease".

(c) Maintainability: We will be adopting an agile model. We will have a 24 hrs support team for easy maintenance.

(d) Portability: The software will be highly portable.

(e) Error handling: The software will undergo a comprehensive testing procedure to make it error free. We will first perform Unit Test on each module, then followed by Integration Testing and then Big Bang testing. Then we will perform Usability Testing and Acceptance Testing on the UI. Different methods of white box testing will be conducted in the code structure.

(f) Ease of use: The website will be easily navigable with proper text and annotations. We will be providing a Graphic User Interface for the user to upload his or her images.

4.2.4 Database Requirements

A train file containing X-ray images of chest Xray images of normal, pneumonic and Covid-19 positive patients. A test file containing X-ray images of chest Xray images of normal , pneumonic and Covid-19 positive patients. This file will be of around 1 GB and will be on the firebase and amazon s3 buckets.

5 Software Design Document

The purpose of this chapter is to provide documentation which will be used to aid in the development by providing the details for how the web-application should be built. The SDD shows how our application will be structured to satisfy the requirements. It describes the software structure, software components, interfaces, and data necessary for the implementation phase.

5.1 Introduction

5.1.1 Design Overview

We are going to use an object-oriented approach towards development of the application to check whether the patient is infected from covid-19 or pneumonia or lung cancer by taking the x-ray reports, uploading it on the user interface into the trained model and getting the report on that UI. This will have various features like a new user registration page, login and logout, generic pages like home contact us etc, upload images(user friendly interface to upload patient information, automatic resnet-50 interface on chest X-ray images, interface on user friendly reports with upload and delete options.

5.1.2 Requirement Traceability Matrix

	Option Menus	Forms	Intuitive Reports	CNN
Login and Registration	X	X		X
Analysing of X-ray Report			X	X
Report Generation			X	
Storage of Information				X

Table 1: Requirement Traceability Matrix

5.2 System Architectural Design

5.2.1 Chosen System Architecture

We will use client-server architecture for our application as it splits the processing of applications across multiple machines and allows easier sharing of resources from client to servers.

5.2.2 Discussion of Alternative Designs

The alternative is peer-to-peer networking where every computer is “equal” and every computer is both a client and a server but it won’t serve our purpose in the first sense.

5.2.3 System Interface Description

Front end interfaces like forms, login and registration will be done using Reactjs, HTML, CSS, & Flask.

5.3 Detailed Description of Components

5.3.1. Option Menus

Option menus will have COVID-19 cases dashboard, COVID-19 detection dashboard, Pneumonia detection dashboard and Lung Cancer detection dashboard that will have information about the authors of the application.

5.3.2. Login and Registration

Registration page will contain a password and email for one time registration. Login page will have an email and password for login into the application.

5.3.3. Intuitive Reports

Intuitive report about covid-19 or not and details of pneumonia or not will be generated after the patient uploads the X-ray report along with appropriate percentage.

5.4 User Interface Design

5.4.1 Description of User Interface

5.4.1.1 Screen Images

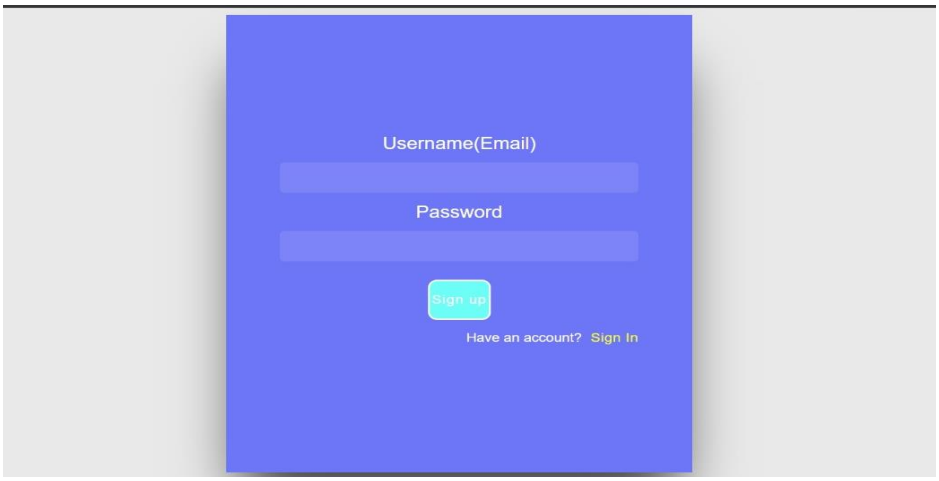


Figure 6: Registration Page

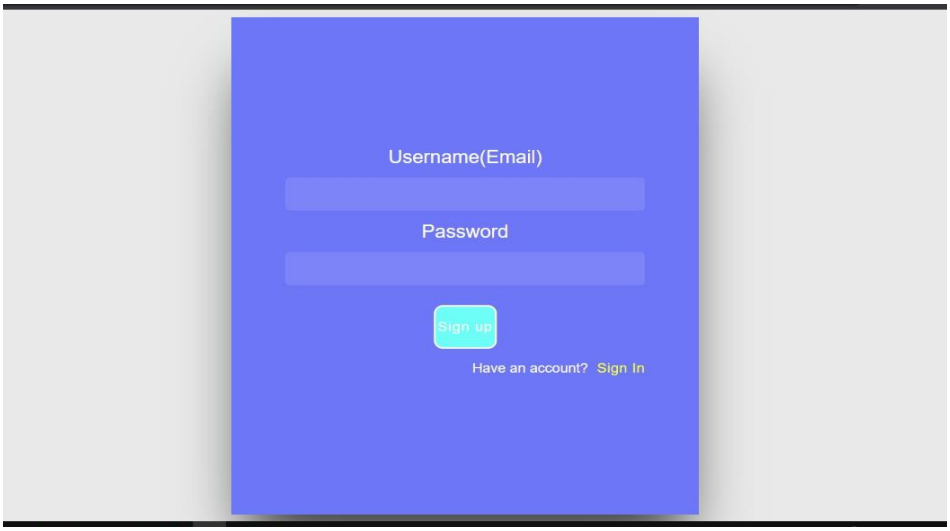


Figure 7: Login Pag

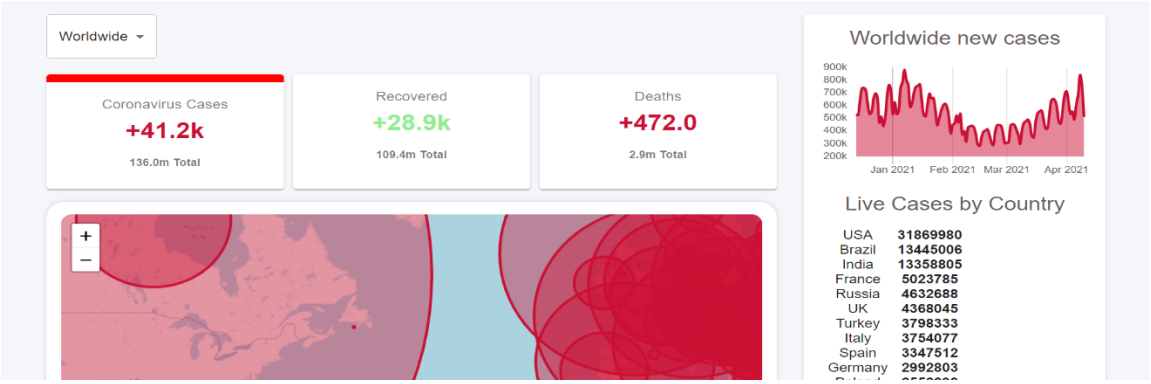


Figure 8: Covid-19 Dashboard

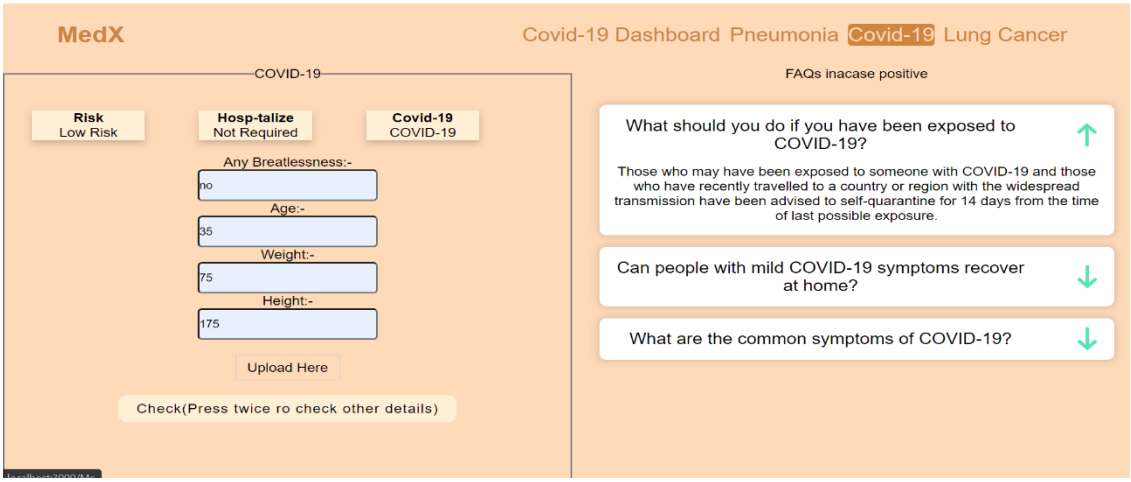


Figure 9: Covid-19 Webpage

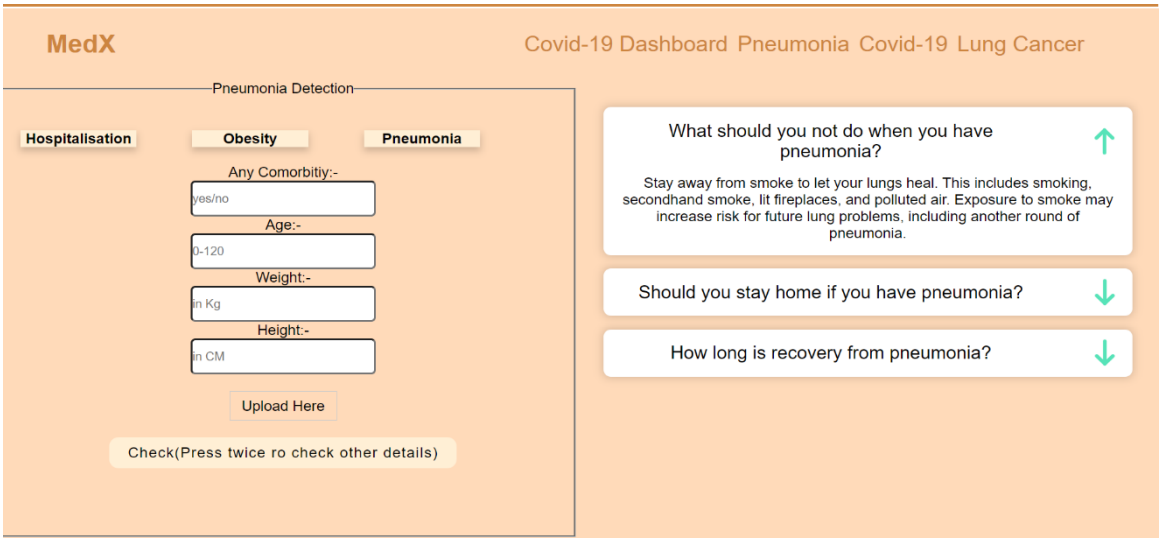


Figure 10: Pneumonia Webpage

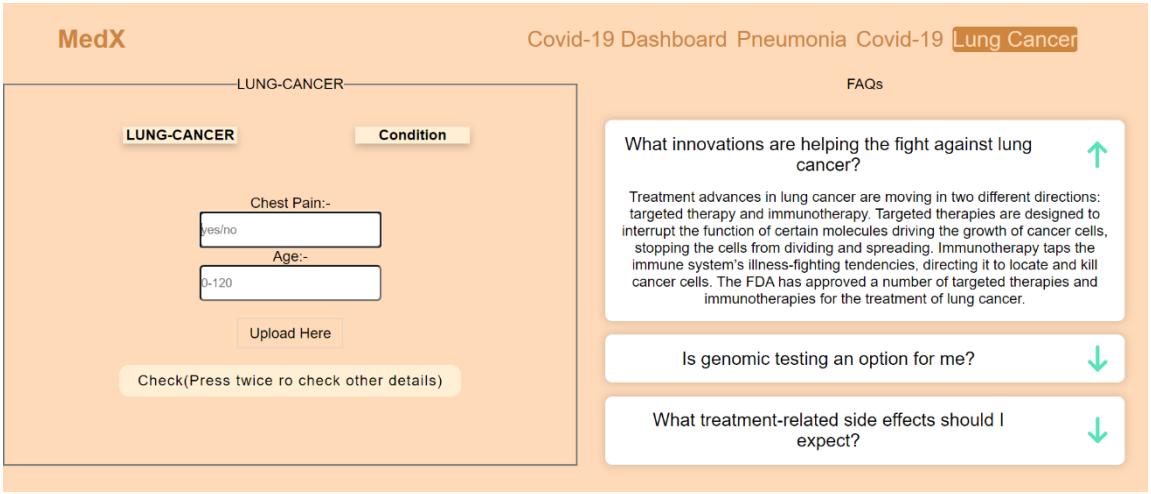


Figure 11: Lung Cancer Webpage

5.4.1.2 Objects and Actions

Objects: Option menus, Registration and Login forms,Upload Data ,View Reports,etc

Actions: Option menus:Will give contact and other information about author of application

Registration and Login forms: Registration page will help users for initial registration into the application. Login page will help user for login to application with credentials

Upload Data: Upload data form will help user to upload X-ray image in to the application

View Report: The report generated using CNN and deep learning models will be shown to the user with the percentage of covid and pneumonia of the user according to the X-ray report uploaded.

6 Implementation

This chapter will notify us about how the project is going to be implemented, what languages, frameworks are going to be used for front-end and back-end, what databases are to be used. In short it will tell us about the tools and techniques to be used for the implementation purpose. The snapshots are attached to the user manual.

6.1 Technologies Used

- Front End
  - Framework
    - React JS : Version
    - Material UI : Version
    - Graph.js : Version
  - Languages
    - HTML 5
    - CSS 3
    - Javascript : Version 6
    - JSX
  - CSS Preprocessor LESS
- Back End
  - Framework
    - Flask: Version 1.1.2
    - Werkzeug(Web Server gateway Interface): Version 1.0.1
    - Tensorflow: Version 2.4.1
    - OpenCV: Version 2
    - Boto: Version 3
  - Language
    - Python : Version 3.8.7
- Database
  - MySQL: Version 5.7
  - MongoDB: Version 3.6

- ❖ Other Tools
  - ➢ Package Dependency Manager
  - ➢ NPM: Version 6.14
  - ➢ Pip: Version 20.2.3
  - ➢ NPX: Version6.14.10

- ❖ IDE
  - ➢ Visual Studio 2017

- ❖ Editors
  - ➢ Atom: Version 2
  - ➢ Sublime Text: Version 3.1

6.2 Implementation

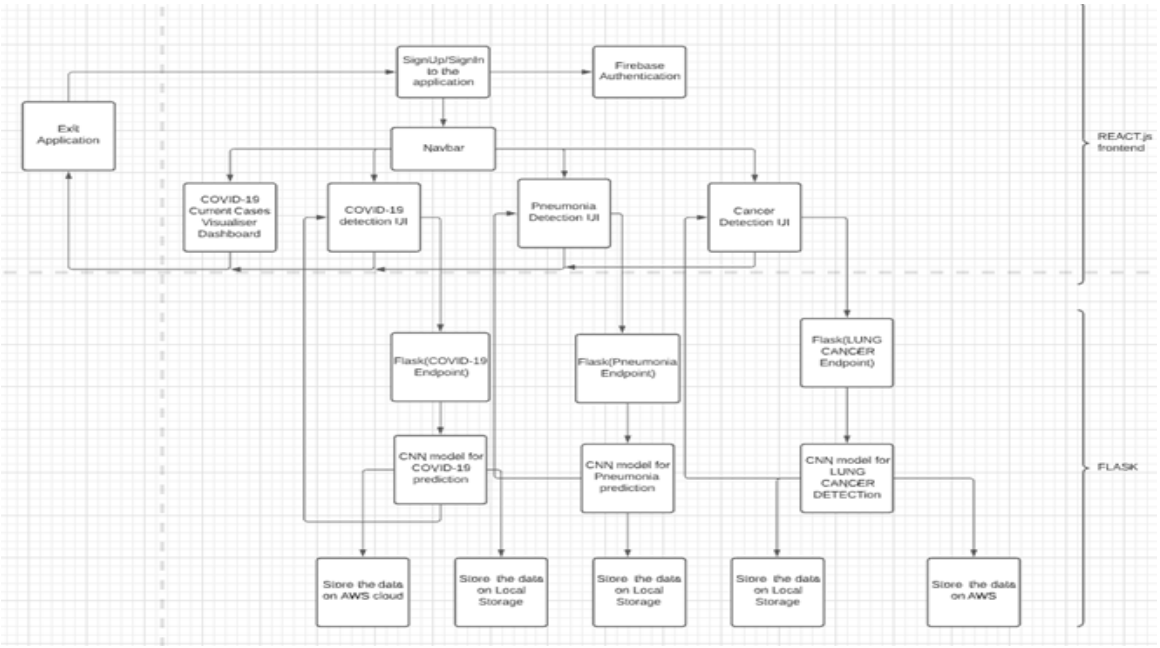


Figure 12: Web-Application Implementation.

6.3 Conclusion:

This chapter told us about how the project is going to be implemented, what languages, frameworks are going to be used for front-end and back-end, what databases are to be used. In short it will tell us about the tools and techniques to be used for the implementation purpose. The next chapter tells us about the testing methodologies to be used for the project like it can be white-box testing, black-box testing, etc. and also about the testing tools and environment. It also tells us about which features should and should not be tested.

7 Software Test Document

This chapter tells us about the testing methodologies to be used for the project like it can be white-box testing, black-box testing, etc. and also about the testing tools and environment. It also tells us about which features should and should not be tested.

7.1 Introduction

7.1.1 System Overview

Our Software will check whether the patient is infected from severe Covid-19, Pneumonia or Lung Cancer, by taking the x-ray reports ,uploading it on the user interface into the trained model and getting the report on that UI . System will also check if the patient is free from pneumonia, but still shares some similar report then it will check for covid-19 through infection in the respiratory tract, and will give the outcome accurately. It will have a UI for the person to SignUp if he doesn't have an account and then SignIn. He will upload the images and then get the result of whether the person has that infection or not.

7.1.2 Test Approach

Now there will be 2 modules in our project. First one will be the deep neural network which we will create as the backend of the product. To test the deep neural network model we will split the data set into train and test. The test set of the images will work as the test cases for the deep learning testing. The code for training the cnn and backend will be written in python so PyUnit or PyTest might be used for testing. We will also have manual inspection of the code which is a form of static testing followed by a walkthrough. We will perform unit testing on each module that is the UI and the CNN model followed by integration testing followed by big bang test. Then the entire project will undergo usability testing followed by acceptance testing.

7.2 TEST PLAN

First the documents will be tested. These documents are Software Requirement Specification, Software Design Document, Software Test Document for any grammatical errors or discrepancy. This will be done manually, that is no testing tool will be used in this process. Now there will be 2 modules in our project. First one will be the deep neural network which we will act as the backend of the product. This will consist of 3 versions. The first version will consist of CNN which will detect pneumonia and the second which will detect Covid-19 and pneumonia. The third version will also have a CNN for detecting Lung Cancer. To test the deep neural network model we will split the data set into train and test. The test set of the images will work as the test cases for the deep learning testing. We will also have manual inspection of the code which is a form of static testing followed by a walkthrough. We will perform unit testing on each module that is the UI and the CNN model followed by integration testing followed by big bang test. Then the entire project will undergo usability testing to check whether it is usable by the required audience followed by acceptance testing to compare the given specification with actual specification.

7.2.1 Features to be tested

1). Buttons of the UI

2). SignUp and SignIn form

3).Uploading of the Image

4).Http Request Response between the client and the server

5). Deep Neural Network for its accuracy and performance

6). The CNN model must do good in both the train and test set.

7). The CNN should take in the correct input and display an error message if the image is not in the right format.

7.2.2 Features not to be tested

All features will be tested.

7.2.3 Testing Tools and Environment

1).PyUnit and PyTest for testing the python code

2).Junit for testing the UI

3).Google Webmaster for testing the performance of the UI

4).Jupyter Notebook for testing the Deep Learning model

7.3 TEST CASES

7.3.1 Case-1

7.3.1.1 Purpose
This test case will test the version of the software that is the third version and it will simply consist of the deep neural network and UI and it will be tested for better accuracy in the test set.

7.3.1.2 Inputs
The inputs will consist of images of pneumonia and non-pneumonic patients which will be a part of the test set.

7.3.1.3 Expected Outputs & Pass/Fail criteria
Pass Criteria:- Accuracy of at least 70 percent
Fail Criteria:- Accuracy of less than 70 percent

Also a relative comparison for selection.

7.3.1.4 Test Procedure
The data set will be divided into train and test set and the test set will be used to test the CNN model

7.3.1.5 Final Result
For pneumonia we tested VGG16, VGG19 and ResNet-50. In this we found that the test accuracy was 86.22%, 88.78% and 50% respectively. So ResNet-50 was rejected because it was less than 70% and amongst VGG16 and VGG19 , VGG19 was selected as it had a slightly higher accuracy.

Input:-Validation Images from dataset
Output:- Given accuracy.
Test Case Passed:-Yes

7.3.2 Case-2

7.3.2.1 Purpose
This test case will test the third of the software that is the third version and it will consist of the deep neural network and UI and it will be tested for better accuracy in the test set.

7.3.2.2 Inputs
The inputs will consist of images of pneumonic covid19 and normal patients which will be a part of the test set.

7.3.2.3 Expected Outputs & Pass/Fail criteria
Pass Criteria:- Accuracy of at least 70 percent
Fail Criteria:- Accuracy of less than 70 percent

Also a relative comparison for selection.

7.3.2.4 Test Procedure
The data set will be divided into train and test set and the test set will be used to test the CNN model

7.3.2.5 Final Result
For this 5 novel architecture were tested with following test accuracy

No's of blocks	Without Skip Connection	With Skip Connection
Block 1	91.57%	-
Block 2	96.67%	50%
Block 3	96.67%	51.67%

Table 2: Test Result (a)

No's of blocks	Without Skip Connection	With Skip Connection
Block 1	74.38%.	-
Block 2	71.16%.	*
Block 3	89.92%.	47.3

Table 3: Test Result (b)

Amongst these architecture we selected the third architecture without skip connection is selected as it has both train and test accuracy as best. The model with skip connection architecture is rejected.

Input: Validation Images from dataset
Output: Given accuracy.
Test Case Passed: Yes

7.3.3 Case-3

7.3.3.1 Purpose
This test case will test the version of the software that is the third version and it will simply consist of the deep neural network and UI and it will be tested for better accuracy in the test set.

7.3.3.2 Inputs
The inputs will consist of images of lung cancer and normal patients which will be a part of the test set.

7.3.3.3 Expected Outputs & Pass/Fail criteria
Pass Criteria: Accuracy of at least 70 percent
Fail Criteria: Accuracy of less than 70 percent
Also, a relative comparison for selection.

7.3.3.4 Test Procedure
The data set will be divided into train and test set and the test set will be used to test the CNN model

7.3.3.5 Final Result
In this we designed a novel CNN architecture for which we got a test accuracy of 97 percent.
The test loss was 0.0469.

Input: Validation Images from dataset
Output: Given accuracy.
Test Case Passed: Yes

7.3.4 Case-3

7.3.4.1 Purpose
This test case will test the password fields of the signup or signIn form of the User. For example the length of the password should be greater than 6.

7.3.4.2 Inputs
It will consist of various entered details of incorrect and correct information of the user.

7.3.4.3 Expected Outputs & Pass/Fail criteria

Pass Criteria:- Proper error response if the Entered details are incorrect
Fail Criteria:- Improper response if the Entered details are incorrect

7.3.4.4 Test Procedure

We will be performing this test using manual entering of password with the expected given condition.

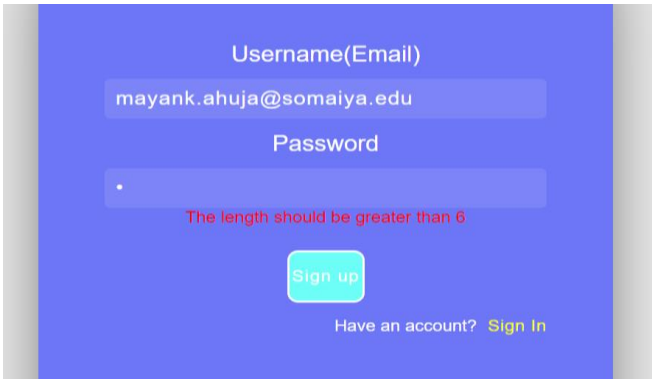


Figure 13: Test Procedure (a)

Input 1:

Email: mayank.ahuja@somaiya.edu

Password:c (length!=6)

Output: Correct error the required field(password field).

Test Case Passed:Yes

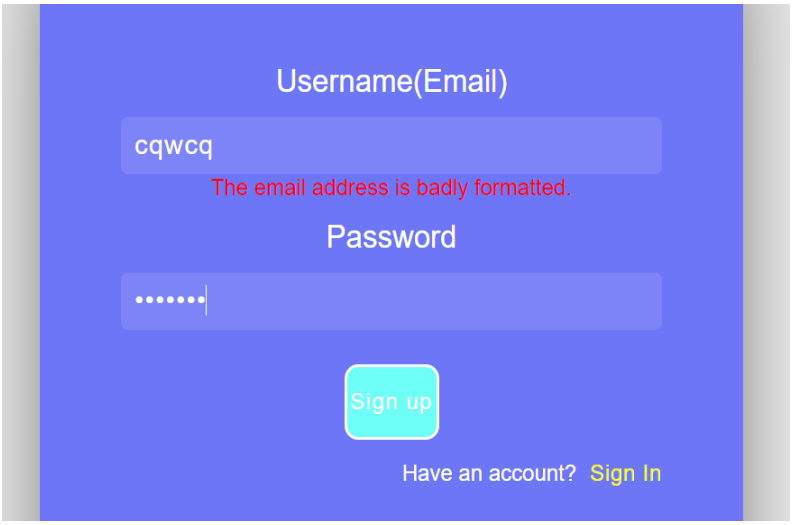


Figure 14: Test Procedure (b)

Input 2:

Email:cqwcq

Password:abcdefg

Output: No error for the required field(password field).

Test Case Passed:Yes

So in this case as we can see the length is 1 which is not allowed. Now as the system is giving the error the test case has passed

7.3.5 Case-4

7.3.5.1 Purpose

This test case will test the email fields of the signup. To be specific it will check for the proper formatting of the email address.

7.3.5.2 Inputs

It will consist of various entered details of incorrect and correct information of the user.

7.3.5.3 Expected Outputs & Pass/Fail criteria

Pass Criteria: Proper error response if the Entered details are incorrect
Fail Criteria: Improper response if the Entered details are incorrect

7.3.5.4 Test Procedure

We will be performing this test using manual entering of email with the expected given condition.

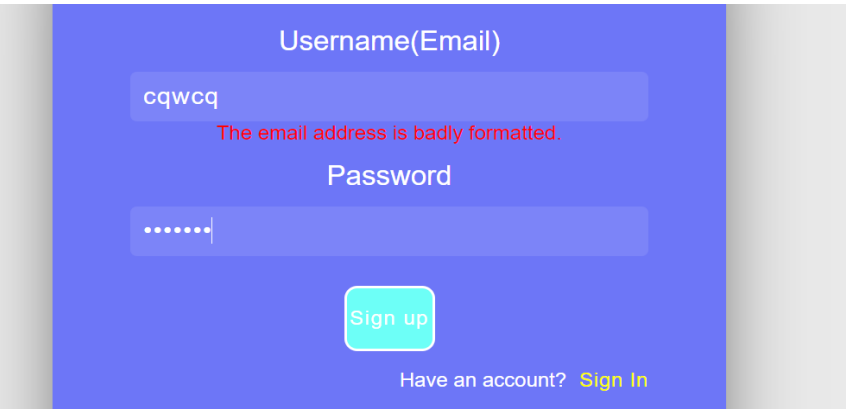


Figure 15: Test Procedure (c)

Input 1:

Email:cqwq

Password:@Ej3O2uw

Output:Error message

Test Case Passed:Yes

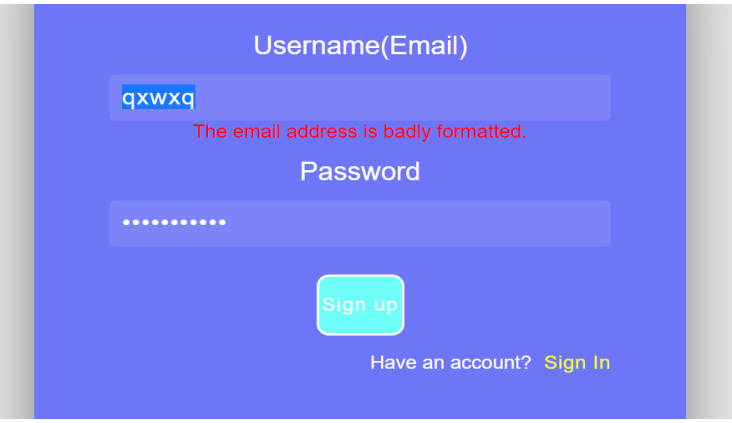


Figure 16: Test Procedure (d)

Input 2: qxwxq
Output: Correct error.
Test Case Passed: Yes

7.3.6 Case-5

3.6.1 Purpose

This test case will test the email fields of the signup. To be specific it will check for the duplicate email address. It will check whether there already exists the entered email address or not.

7.3.6.2 Inputs

It will consist of various entered details of incorrect and correct information of the user.

7.3.6.3 Expected Outputs & Pass/Fail criteria

Pass Criteria: If the Entered details are existing and doesn't allow the signup.
Fail Criteria: If the Entered details are existing and it allows the signup.

7.3.6.4 Test Procedure

We will be performing this test using manual entering of email with the expected given condition.

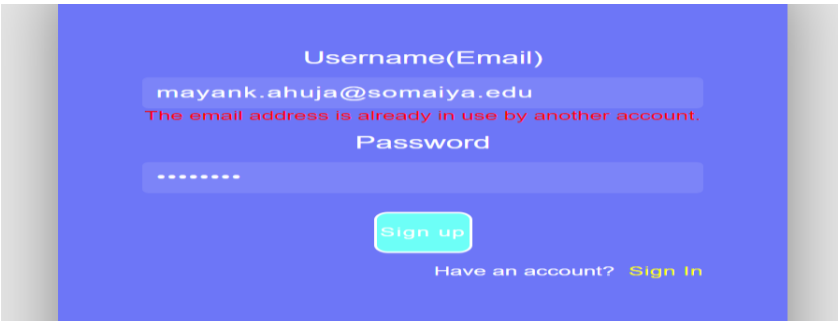
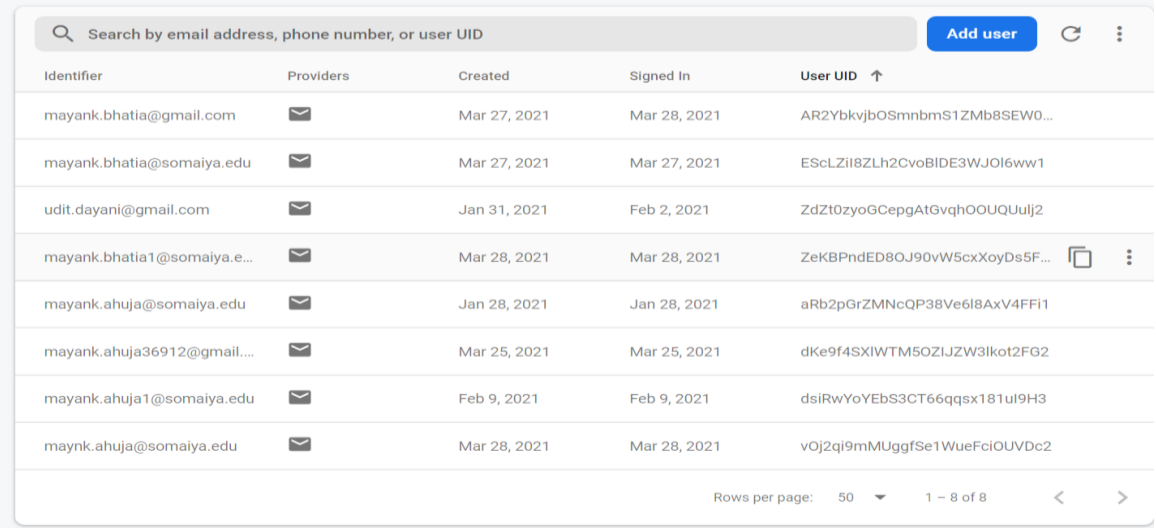


Figure 17: Test Procedure (e)

Trying to sign up with an already existing email address.



Input 1:

Email: mayank.ahuja@somaiya.edu

Password: @Ej3O2uw

Output: Error message

Test Case Passed:-Yes

Input 2:

Email:mayank.bhatia1@somaiya.edu

Password:@Ej3O2uw

Output:Error message

Test Case Passed:Yes

7.3.7 Case-6

3.7.1 Purpose

This test case will test the password fields of the Sign In. To be specific it will check for the password whether it is correct or not.

7.3.7.2 Inputs

It will consist of various entered details of incorrect and correct information of the user.

7.3.7.3 Expected Outputs & Pass/Fail criteria

Pass Criteria: If the entered password is incorrect and it doesn't allows the sign in.
Fail Criteria: If the entered password is incorrect and it allows the sign in.

7.3.7.4 Test Procedure

We will be performing this test using manual entering of password with the expected given condition.

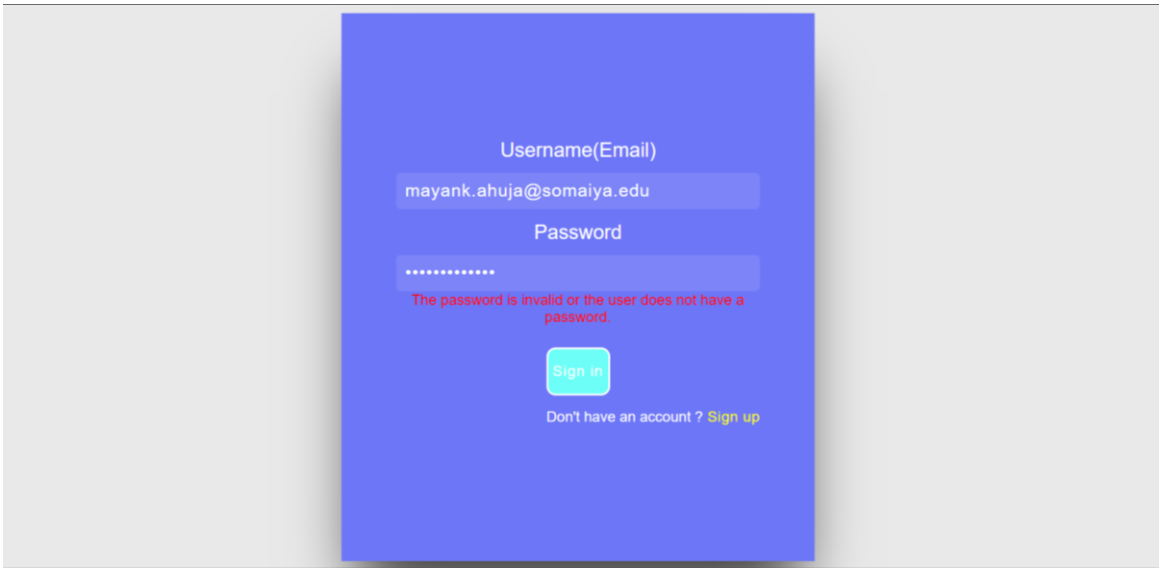


Figure 18: Test Procedure (f)

Input 1:

Email:mayank.ahuja@somaiya.edu
Password:abcdefghijk

Output:Error message
Test Case Passed:Yes

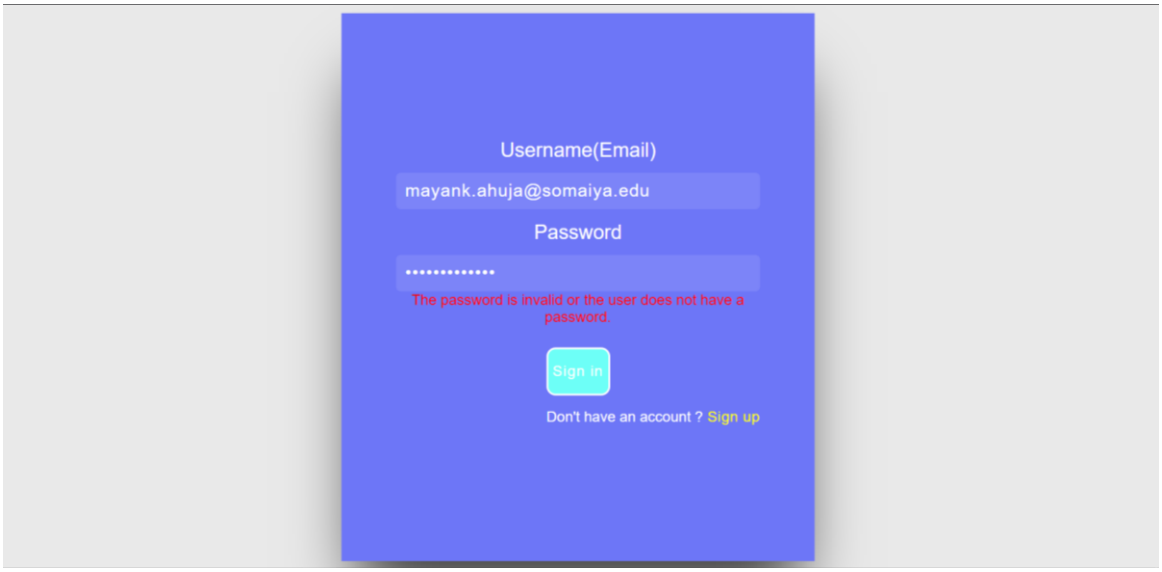


Figure 19: Test Procedure (g)

Input 2:

Email:mayank.ahuja@somaiya.edu
Password:abcdefghijl
Output:Error message
Test Case Passed:Yes

7.3.8 Case-7

7.3.8.1 Purpose

This test case we will test for the http connection request between the client(reactjs) and the server(flask)

7.3.8.2 Inputs

It will consist of an inbuilt dummy call from reactjs client server to the particular flask function.

7.3.8.3 Expected Outputs & Pass/Fail criteria

Pass Criteria: If in the command line we get to see the http request to the flask function then pass.
Fail Criteria: If in the command line we don’t get to see the http request to the flask function then fail.

7.3.8.4 Test Procedure

A dummy call will be made from client to the server to see there exists a communication between the client and the server(flask api).

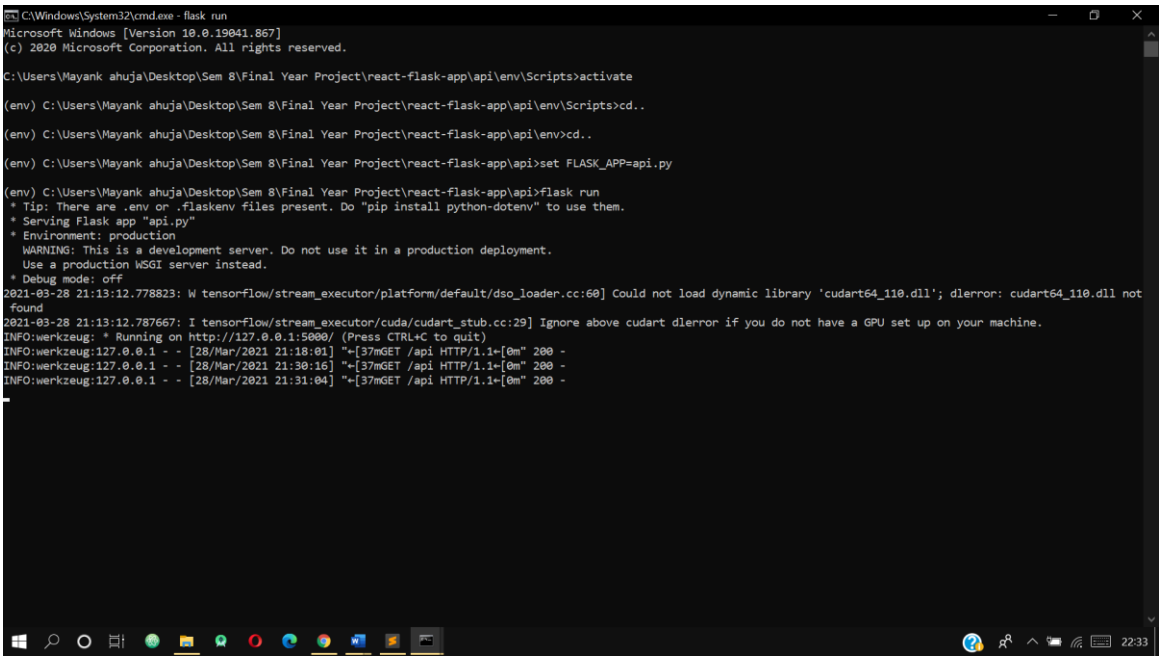


Figure 20: Test Procedure (h)

INFO:werkzeug:127.0.0.1 - - [28/Mar/2021 21:31:04] "←[37mGET /api HTTP/1.1←[0m" 200 – As we can see the request is successfully interpreted by the flask function running in the python virtual environment.

Input: GET request
Output: 200 successful
Test Case Passed: Yes

7.3.9 Case-8

7.3.9.1 Purpose

This test case we will test for validity of the input in the pneumonia form field

7.3.9.2 Inputs

It will consist of inputs entered into the form as well as uploading the file for detection of pneumonia

7.3.9.3 Expected Outputs & Pass/Fail criteria

Pass Criteria: If we see an error message when wrong or incomplete information is entered or wrong file format is uploaded.
Fail Criteria: If we see an error message when wrong or incomplete information is entered or wrong file format is uploaded.

7.3.9.4 Test Procedure

We enter different details and upload images of different file formats to check for errors. Age(0-120), Weight(0-250), Height(0-230)

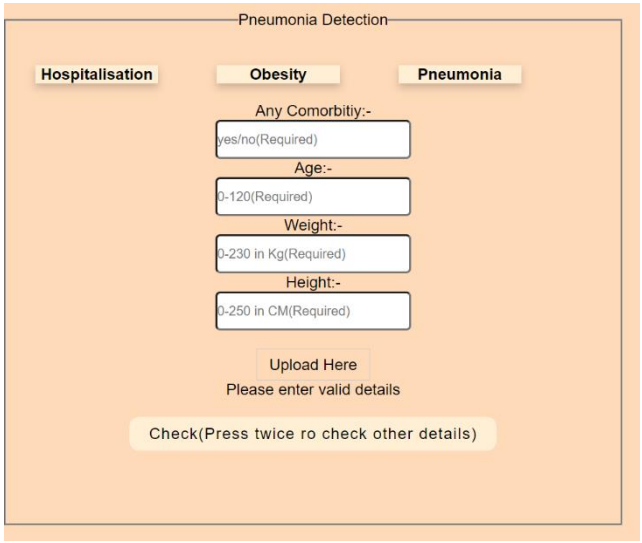


Figure 21: Test Procedure (i)

Input: no information
Output: Please enter valid details.
Test Case Passed: Yes

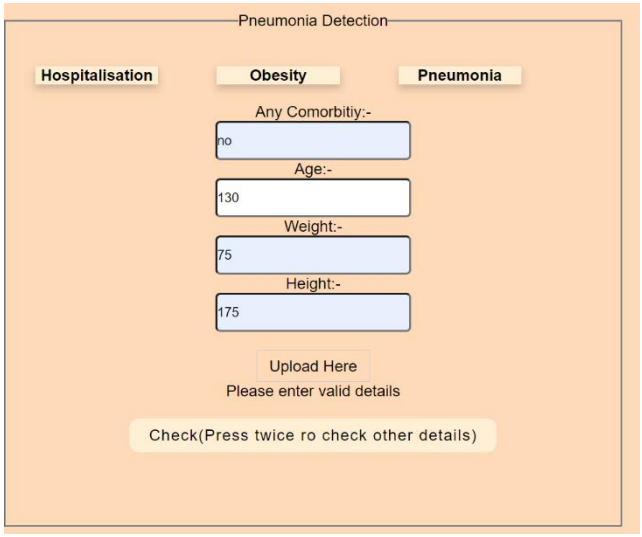


Figure 22: Test Procedure (j)

Input: Wrong information entered (Age>120)
Output: Please enter valid details.
Test Case Passed: Yes

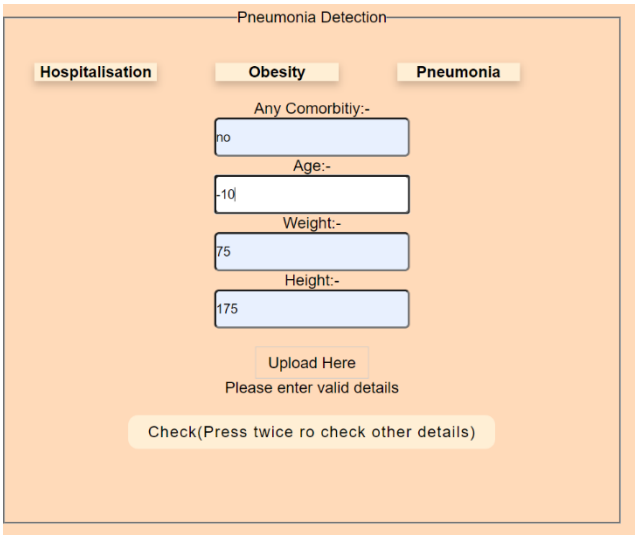


Figure 23: Test Procedure (k)

Input: Wrong information entered (Age<0)
Output: Please enter valid details.
Test Case Passed: Yes

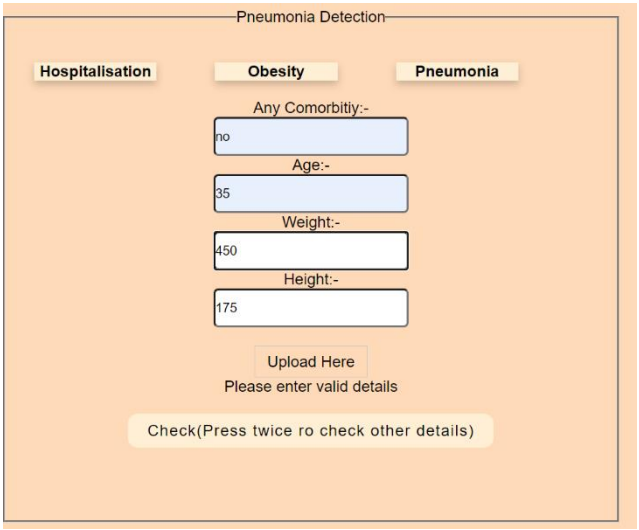


Figure 24: Test Procedure (l)

Input:- Wrong information entered (Weight>442)
Output:- Please enter valid details.
Test Case Passed:-Yes

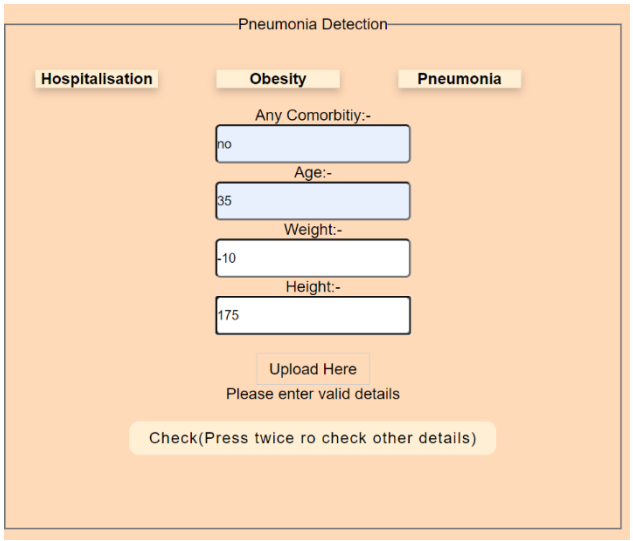


Figure 25: Test Procedure (m)

Input:- Wrong information entered (Weight<0)
Output:- Please enter valid details.
Test Case Passed:-Yes

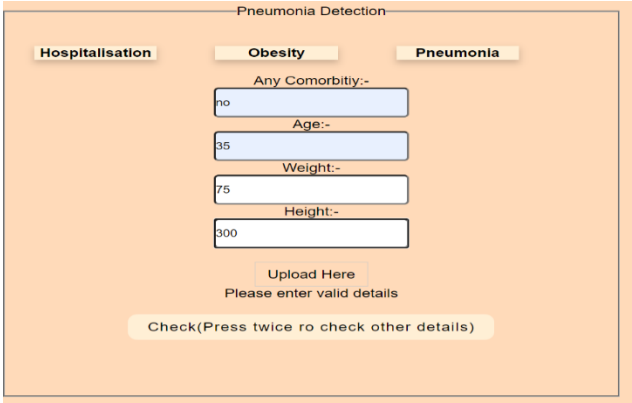


Figure 26: Test Procedure (n)

Input:- Wrong information entered (Height>270)
Output:- Please enter valid details.
Test Case Passed:-Yes

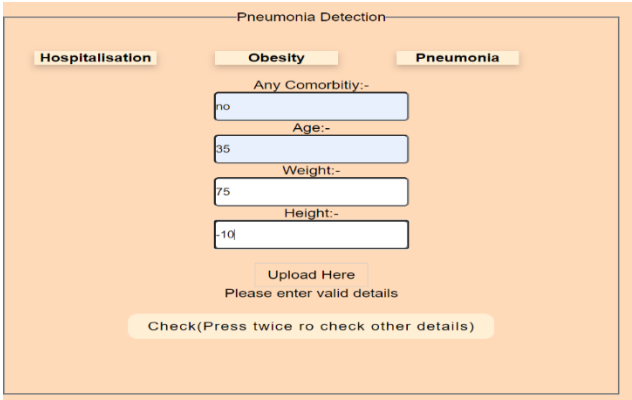


Figure 27: Test Procedure (o)

Input:- Wrong information entered (Height<0)
Output:- Please enter valid details.
Test Case Passed:-Yes

7.4 Conclusion

This chapter described the testing methodologies to be used for the project like it can be white-box testing, black-box testing, etc. and also about the testing tools and environment. It also tells us about which features should and should not be tested. The next chapter deals with the conclusion and the future scope of the project. The conclusion concludes that the specified requirements have been implemented. The future scope deals with the new features which are/can be implemented in future which will enhance the application more.

8 Conclusion and Future Work

This section summarizes the results of the project as a whole and sets up a topic for discussion as to what can be inferred from it. The conclusion concludes that the specified requirements have been implemented. The future scope deals with the new features which are/can be implemented in future with enhanced technology which will help the application to be more productive.

8.1 Conclusion

This project is an interactive web-application for everyone who experiences the symptoms of covid-19, pneumonia, or lung-cancer and can easily be diagnosed with this web-app by uploading their lungs x-ray and get their in-depth report in a second which also suggests the immediate action needed along with the FAQs. We tested various Neural Networks in order to provide the best accuracy, and then chose VGG-19 as our neuron, which provided 97% accuracy for covid-19; 90% for pneumonia & 99% for lung-cancer. Thus, it fulfils the need of an e-pathology lab which is able to detect various lung related diseases using chest X-ray images. The User Interface design is simple, and it is very easy for the user to understand the application. Each member was assigned one or more roles and everyone has successfully completed their responsibilities.

8.2 Future Work

Various disease detection can be added to make it a complete e-pathology lab and able to replace the pathology labs.

Severity level can be added if any disease is detected to give the in-depth knowledge of danger.

Near-by best available doctor’s / hospitals list could be added if disease is detected.

Best home-remedies / medicines can be suggested.

A ‘Chat with Doctors’ feature can be introduced.

9. References

[1] Muhammad Farooq and Abdul Hafeez et al., “COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs” Arxiv, Mar 2020, arXiv:2003.14395

[2] Ioannis D. Apostolopoulos Tzani A Mpesiana et al., “ Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks.” Arxiv, Mar 2020, arXiv:2003.11617

[3] Joseph Paul Cohen Lan Dao , Paul Morrison , Karsten Roth , Yoshua Bengio , Beiyi Shen, MD Almas Abbasi, MD Mahsa Hoshmand-Kochi, MD Marzyeh Ghassemi , Haifang Li , Tim Q Duong. et al., “Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning” NCBI. Jul 2020 arXiv:2005.11856

[4] Linda Wang , Zhong Qiu Lin , and Alexander Wong et al., “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images.” NCBI, Nov 2020 arXiv:2003.09871

[5]Muhammad Farooq and Abdul Hafeez et al., “COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs” Arxiv, Mar 2020, arXiv:2003.14395

[6] Ioannis D. Apostolopoulos Tzani A Mpesiana et al., “ Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks.” Arxiv, Mar 2020, arXiv:2003.11617

[7] Joseph Paul Cohen Lan Dao , Paul Morrison , Karsten Roth , Yoshua Bengio , Beiyi Shen, MD Almas Abbasi, MD Mahsa Hoshmand-Kochi, MD Marzyeh Ghassemi , Haifang Li , Tim Q Duong. et al., “Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning” NCBI. Jul 2020 arXiv:2005.11856

[8] Linda Wang , Zhong Qiu Lin , and Alexander Wong et al., “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images.” NCBI, Nov 2020 arXiv:2003.09871

[9] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[10] Maria de la Iglesia Vayá, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier Barber, Domingo Orozco-Beltrán, Francisco García-García, Marisa Caparrós, Germán González, Jose María Salinas et al. “BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients.” Arxiv, Jun 2020 arXiv:2006.01174

[11] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong Marzyeh Ghassemi et al. “COVID-19 Image Data Collection: Prospective Predictions are the Future.” Arxiv, Jun 2020, arXiv:2006.1198

[12] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization.” Arxiv,Oct 2016, arXiv:1610.02391

10. Appendices

10.1 Research Paper Publication Certificates:

SOMAIYA VIDYAVIHAR

K J Somaiya Institute of Engineering & Information Technology

An Autonomous Institute affliated to Universirty of Mumbai

Accredited by NAAC and NBA, Approved by AICTE, New Delhi

ORGANISES

4TH INTERNATIONAL CONFERENCE ON ADVANCES IN SCIENCE AND TECHNOLOGY (ICAST-2021)

CERTIFICATE

This is to certify that Mayank Ahuja

has participated/presented a paper titled

Covid-19 Detection Using Convolution Neural Networks

Published at SSRN- Elsevier Online Conference Publication Portal for 4th International Conference on "Advances in Science and Technology" (ICAST-2021) organized by K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai-22 on 7th/8th of May, 2021 in online mode.

Dr. Sunita Patil
Convenor - ICAST 2021,
Vice-Principal

Dr. Suresh Ukarande
Chairperson - ICAST 2021,
Principal

SOMAIYA VIDYAVIHAR

K J Somaiya Institute of Engineering & Information Technology

An Autonomous Institute affliated to Universirty of Mumbai

Accredited by NAAC and NBA, Approved by AICTE, New Delhi

ORGANISES

4TH INTERNATIONAL CONFERENCE ON ADVANCES IN SCIENCE AND TECHNOLOGY (ICAST-2021)

CERTIFICATE

This is to certify that Deep Chothani

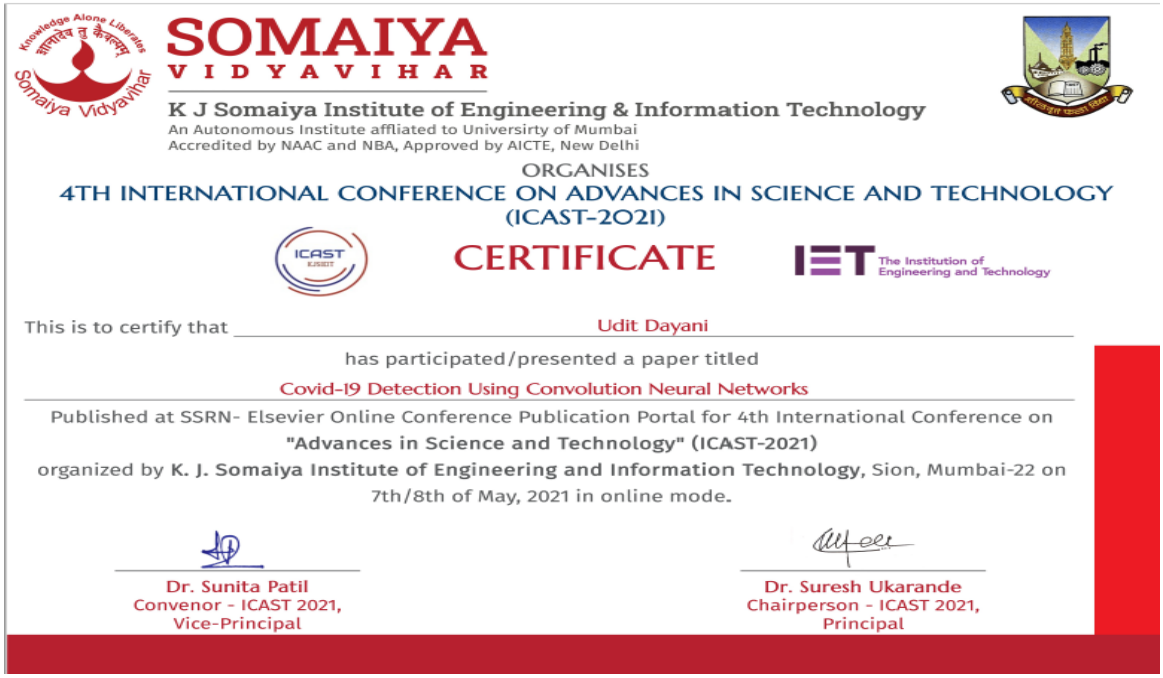
has participated/presented a paper titled

Covid-19 Detection Using Convolution Neural Networks

Published at SSRN- Elsevier Online Conference Publication Portal for 4th International Conference on "Advances in Science and Technology" (ICAST-2021) organized by K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai-22 on 7th/8th of May, 2021 in online mode.

Dr. Sunita Patil
Convenor - ICAST 2021,
Vice-Principal

Dr. Suresh Ukarande
Chairperson - ICAST 2021,
Principal



SOMAIYA VIDYAVIHAR

K J Somaiya Institute of Engineering & Information Technology

An Autonomous Institute affliated to Universirty of Mumbai

Accredited by NAAC and NBA, Approved by AICTE, New Delhi

ORGANISES

4TH INTERNATIONAL CONFERENCE ON ADVANCES IN SCIENCE AND TECHNOLOGY (ICAST-2021)

CERTIFICATE

This is to certify that Udit Dayani

has participated/presented a paper titled

Covid-19 Detection Using Convolution Neural Networks

Published at SSRN- Elsevier Online Conference Publication Portal for 4th International Conference on "Advances in Science and Technology" (ICAST-2021) organized by K. J. Somaiya Institute of Engineering and Information Technology, Sion, Mumbai-22 on 7th/8th of May, 2021 in online mode.

Dr. Sunita Patil
Convenor - ICAST 2021,
Vice-Principal

Dr. Suresh Ukarande
Chairperson - ICAST 2021,
Principal

10.2 Published Research paper:

Covid-19 Detection Using Convolution Neural Networks

Prof. Era Johri #1, Mayank Ahuja #1, Deep Chothani #1, Udit Dayani #1

(erajohri@somaiya.edu) (mayank.ahuja@somaiya.edu) (deep.chothani@somaiya.edu) (udit.dayani@somaiya.edu)

Assistant Professor, students, Department of Information Technology

K.J. Somaiya College of Engineering, Vidyavihar.

Abstract— Abstract—Spread of a deadly and a highly contagious virus named COVID-19 has become conspicuous in the last couple of years. This disease has opened a new arena for scientists and researchers. A new challenge has entered the society to diagnose the disease at a very fast pace along with accuracy to help the medical practitioners. The superannuated approach taken in the medical history is the lung X- RAY. The amelioration in the image processing using convolution neural networks will help us to identify the damage caused by any such diseases which can be detected using chest X-rays. In this paper, we start developing convolutional neural networks of various architectures and compare their validation accuracy, validation loss, train accuracy and test loss over a determined number of epochs by using chest X-ray reports. The architecture with the best validation accuracy along with a good train accuracy can be used for the detection of COVID-19. Postulation like transfer learning will be implemented in our future models to increase the accuracy of the model and heat maps to find the affected region.

Keywords: coronavirus, covid-19, pneumonia, Lung Cancer, diagnostic centre, CNN, neural networks, Tensorflow, blood tests, x-ray (roentgenogram), deep learning, disease predictor, image processing, epoch(time).

1. Introduction

Detection of Covid19 through blood/swab tests is very costly as well as time consuming as compared to the detection by neural network as a trained neural network returns the result within seconds since when the image is fed into the input layer. In this proposed system, we instantly can analyse the area of damage to the lungs and affirm that the person is infected with CoVID-19 or not. Because the ratio of doctor to patient is not optimal, therefore, there is a possibility of critical cases left unentertained. This project can also be advantageous in remote areas of India where there is likelihood of X-Ray machines and difficult to deliver the test. Our system of neural networks will automate the process of identifying the infected cases more efficiently.

Despite the fact that there are various ways for directing COVID-19 tests, there are sure challenges and disadvantages related to them. Detection of COVID-19 through swab tests is exorbitant. Investigating blood/swab tests require some human labour when contrasted with the proposed framework where the outcomes are produced within fraction of seconds subsequent to transferring the report. In far off zones of India it is exceptionally hard to recognize the quantity of cases because of absence of clinical facilities.

Convolutional neural networks are a type of neural networks which are used prominently for object detection with the help of kernels. They come under supervised learning and consist of multiple convolutional blocks. Convolutional neural networks are being used in various fields of medicine, computer science, finance etc. But the most prominent applications of CNN models are in healthcare.
They are used for detection of various diseases like skin cancer, tumor in the brain or tuberculosis in lungs. In recent years

convolutional neural networks are used in pneumonia detection in patients.

We will also analyse the region affected by the help of Grad Cam technique. This is used to detect as to what part of the image the conv-net detects the object, that is to say as to which part of the do we find the object under study.

2. Literature Survey

Analysing roentgenogram reports using Convolution Neural Networks has been used for several decades to explore and visualize fractures or abnormalities in body organs and diagnose sensitive parts like bones, chest, teeth, skull etc. roentgenogram images can be analysed by various methods like multilayer probabilistic method, learning vector quantization and generalized regression networks. These are better in classification but less efficient in terms of accuracy and time of computation. Since both time and accuracy is important for the detection of such malignant disease, we use a deep learning model based on CNN architecture that has the power of extracting different level peculiarity from the images.

These are better in classification but less efficient in terms of precision and time of computation. Since both time and accuracy is important for the detection of such malignant disease, we use a deep learning model based on CNN architecture that has the strength of extracting different peculiarities from the images in a particular dataset.

A Deep Learning Framework for Screening of COVID-19 from Radiographs by Muhammad Farooq1 and Abdul Hafeez2, where they have used ResNet50 to detect whether the person has COVID-19, pneumonia or not. There is a balance between the performance and the speed of training it. Also it has strived to burgeon the accuracy to about ninety percent. ResNet50 is a residual neural network which provides a higher accuracy as it allows a greater depth in the neural network.[1]

Automatic detection from roentgenogram images utilizing transfer learning with convolutional neural networks. In that proposal, it is observed that transfer learning techniques are used to train the neural network in order to evaluate the performance of Convolutional Neural Networks for classification of images where various neural networks like VGG19, Mobile Net, Inception and Xception were used. 10-fold-cross-validation was performed in the training and evaluation process. Automatic detection and extraction of features from roentgenogram using transfer learning had major effects in COVID-19 diagnosis.[2]

Joseph Paul Cohen, Lan Dao, Paul Morrison, Karsten Roth, Yoshua Bengio, Beiyi Shen, Almas Abbasi, Mahsa Hoshmand-Kochi, Marzyeh Ghassemi, Haifang Li, Tim Q Duong wrote about the different approaches of deep learning to the detection of COVID-19 pneumonia and presented a severity score prediction model using frontal Chest X-ray images [3]. The DenseNet model was used from TorchXRay library for training as these models are widely used in prediction of Pneumonia along with Geographic extent and Opacity score as the metrics for accuracy. For the prediction of extent of lung involvement, a linear regression algorithm was performed.

A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images, the researchers have

calculated the efficiency and sensitivity of a novel neural network which was created by Darwin.ai. This neural net is named COVID-19-net and the dataset which it works on is COVID-19x [4]. However, it is not ready for industrial use but the features of COVID-19x neural network are very promising for long run industrial use.

Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun have worked on a residual neural network which is relatively deeper than VGG - 19 and VGG-16. This neural network makes use of skip connection to get a deeper network with better accuracy which is not possible in normal CNNs.

There are two versions of ResNets one with Convolution layer first followed by batch normalization and activation function and one with convolution layer after batch normalization and activation function. [5]

This data is the chest X-ray images of people having COVID-19. The chest X-RAY were both posterior anterior as well as anterior posterior. The data is on the link dataset link. It consists of both CT and CX scans. This dataset is used in the later versions of this research paper.

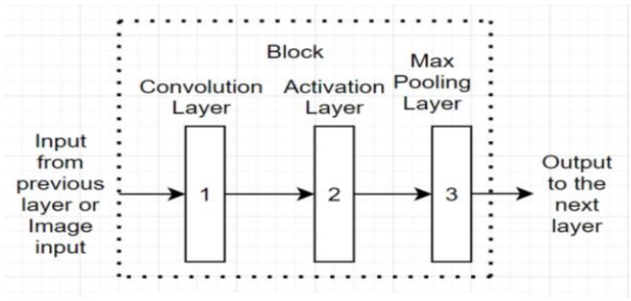
Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong, Marzyeh Ghassemi have worked on a dataset as well. This dataset consists of X-RAY images of patients having different types coronaviruses like SARS, MERS and COVID-

We consider only a part this dataset that has X-RAYS of patients that are COVID-19 positive. Although their dataset is much lesser but it also has more details as to which patient has what gender, or what type of coronavirus he had ie, SARS, MERS or SARS COVID-19. This dataset is used for the training of our models.[7]

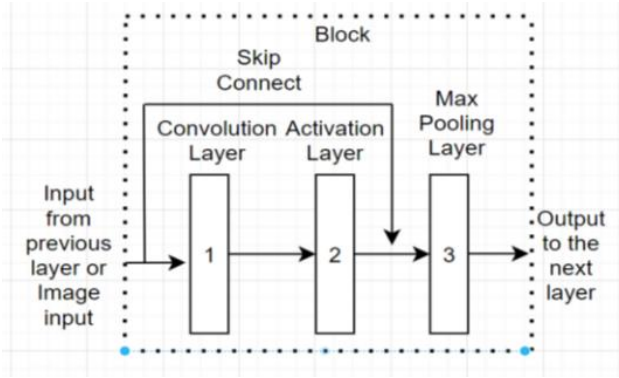
3. Methodology

A). Architecture of a block

The cardinal goal of this research is to proffer a system which will automate the process of COVID-19 detection through the chest roentgenogram reports. Here, a block is defined as a combination of convolution layer, followed by activation layer and then a max pooling layer. The skip connection helps to train a very large neural network efficiently without overfitting. This concept of skip connection is taken from residual neural networks like ResNet-50 wherein we skip a block of neural network and then add the value after that block. But in this case the matrix of the values which are input into the convolution layer of the block is added with the matrix of values output from the activation layer of that block as shown in the diagram.



Without Skip Connection



With Skip Connection

The skip connection here, involves skipping the convolution layer and the activation layer, such that the input for the max pooling layer is the original input which goes in the convolution layer along with the output of the activation layer.

B). Dataset

The dataset which we have taken into account is given in this link which consists of 224 images belonging to 2 classes that are COVID-19 positive and Normal. This dataset is that of research paper [7] discussed above in the literature review. We make use of train and validation generators with rescale value of 1./255, sheer range of 0.2, zoom range of 0.2 and the images are flipped horizontally. Wherein the target size is 224 by 224, batch size is 32 and the class mode is binary. We consider first 20 epochs for our data. The dataset described in paper [7] having X-RAY images of COVID-19 positive patients is merged with a dataset having X-RAY images of COVID-19 negative patients as this is a binary classification problem which has 2 classes and is hosted on the given link below discussed in methodology. The dataset having X-RAY images of COVID-19 negative (NORMAL) patients is taken from Kaggle dataset on this link. We have seen to it that the number of images in both the classes i.e. COVID-19 positive and COVID-19 negative (NORMAL) is same.

C). Hyper-parameters

The performance will be measured in terms of train accuracy, train loss, validation accuracy and validation loss of 1, 2 and 3 block convolutional neural networks with and without skip connection on the COVID-19 dataset. The skip connection for 1-block convolutional neural networks cannot be considered due to shortcoming of the dimensions of the matrix as we cannot add 2 matrix with different dimensions. The values from the output of the final block will flatten and will be fed into a dense layer of artificial neural network which has 16 neurons and has a dropout of 0.5 which will then have a relu activation layer. The values from the output of this layer will be fed into another artificial neural network layer having a single neuron with a sigmoid activation function. We compile this model with Adam's optimiser and the loss is calculated using binary cross entropy. The model is trained with 20 epochs and at the end of the 20th epoch the value of the train accuracy, train loss, validation accuracy and validation loss of 1, 2 and 3 block convolutional neural networks with and without skip connection on the COVID-19 dataset will be chronicled

In the future research these tailored CNN models will be built for detection of such diseases in patients based on the roentgenogram [CX] of the patients which will be compared with the pre-existing CNN models like VGG-19 and VGG-16 discussed in paper [2] and then the best model will be used for the application which could be used instead of the ongoing tedious and costly procedures. In the future we will also be generating the heatmap using Grad-Cam to find out as to which part of the lung is affected. Grad-Cam is the technique used for the construction of the heat map, to

detect which part of lungs is infected from the disease as it helps to detect the important parts of the image using the gradient of the classification score determined by the network.[8]

This focuses on the different techniques used in order to achieve the maximum accuracy viz. by taking different numbers of blocks in neural network layers and comparing its accuracy with the skip and without skip connection models.

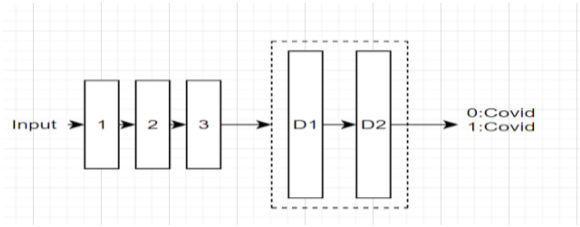


Fig 4.1.1.1(a)

1-Convolutional layer
2-Activation layer
3-MaxPooling layer
D1-Dense layer[16 neurons activation function-'relu']
D2-Dense layer[1 neuron activation function-'sigmoid']

Fig 4.1.1.2

In this architecture the validation accuracy was found to be 91.57% whereas the train accuracy was found to be 74.38% at the 20th epoch. As we can see in the diagram for train loss and the validation loss became stable after 5 to 10 epoch. The validation loss and test loss was found to be 0.362 and 0.5449 respectively at the end of 20th epoch.

4.Results and Discussions

Different Block Results with and without Skip Connection:

4.1.1: 1 Block Without Skip Connection

1 - Block [Convolution layer, Activation layer and MaxPooling layer]

4.1.2: 1 Block With Skip Connection

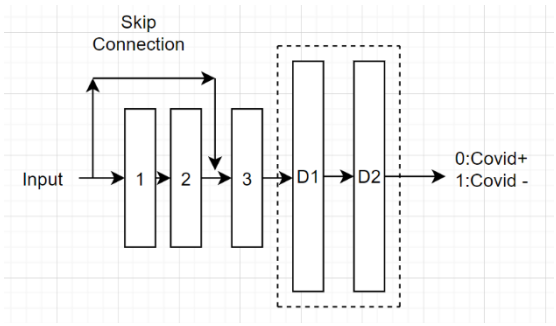
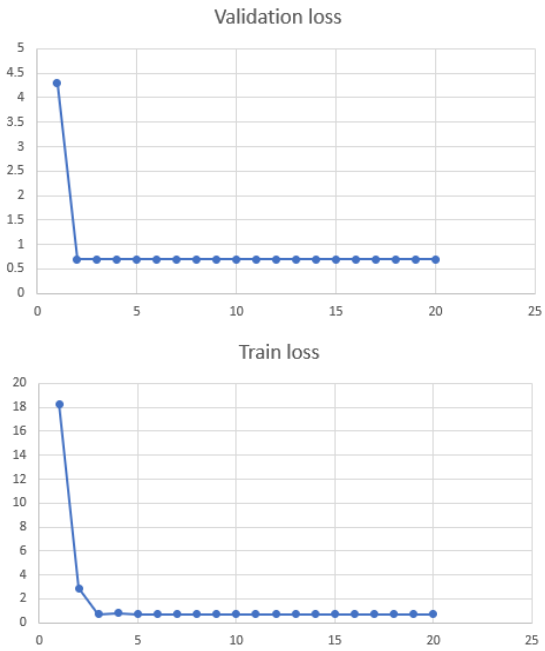


Fig 4.1.1.1(a)



In this architecture the validation accuracy was found to be 50% whereas the train accuracy was found to be 49% at the 20th epoch.

4.2.1: 2 Blocks Without Skip Connection

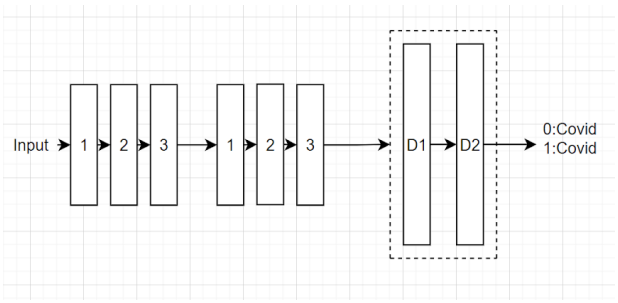


Fig 4.2.1 (a)

1-Convolutional layer
2-Activation layer
3-MaxPooling layer
D1-Dense layer[16 neurons activation function-'relu']
D2-Dense layer[1 neuron activation function-'sigmoid']

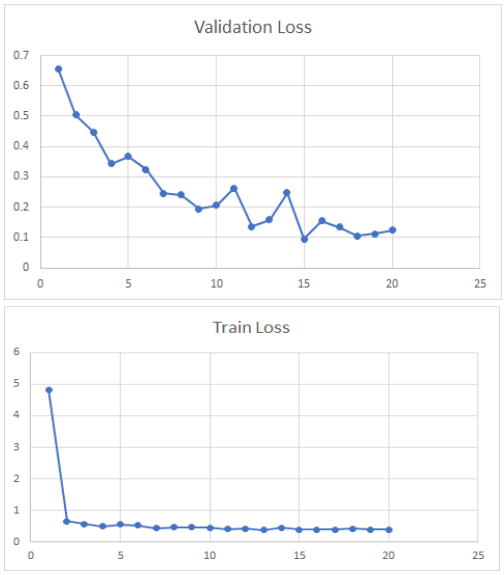


Fig 4.2.1 (b)

In this architecture the validation accuracy was found to be 96.67% whereas the train accuracy was found to be 71.16% at the 20th epoch. As we can see in the diagram, the train loss became stable after the 2nd epoch. The validation loss and test loss was found to be 0.1238 and 0.3997 respectively at the end of 20th epoch.

4.2.2: 2 Blocks With Skip Connection

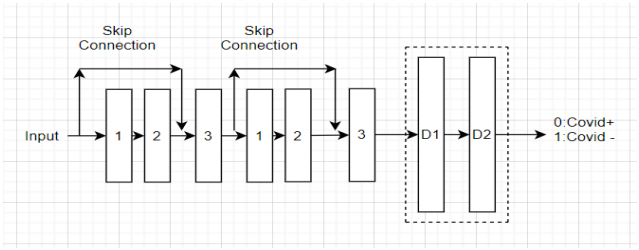


Fig 4.2.2 (a)

1-Convolutional layer
2-Activation layer
3-MaxPooling layer
D1-Dense layer[16 neurons activation function-'relu']
D2-Dense layer[1 neuron activation function-'sigmoid

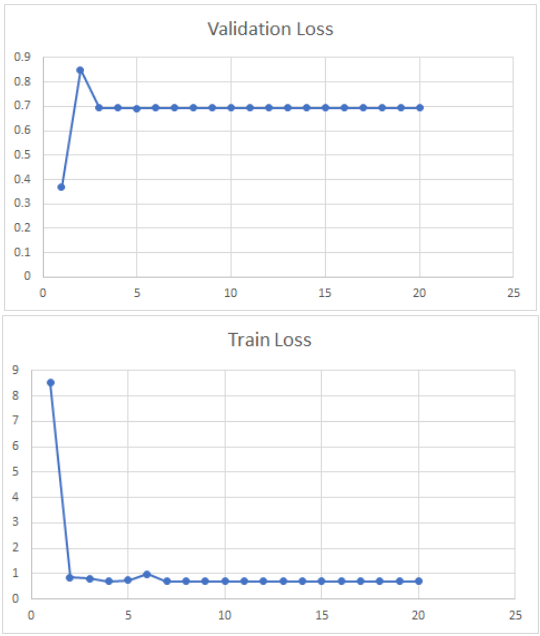


Fig 4.2.2 (b)

In this architecture the validation accuracy was found to be 50% whereas the train accuracy was found to be 47.38% at the 20th epoch. As we can see in the diagram, the train loss and the validation loss became stable roughly after the 3rd epoch. The validation loss and test loss was found to be 0.6931 and 0.6931 respectively at the end of 20th epoch.

4.3.1: 3 Blocks Without Skip Connection

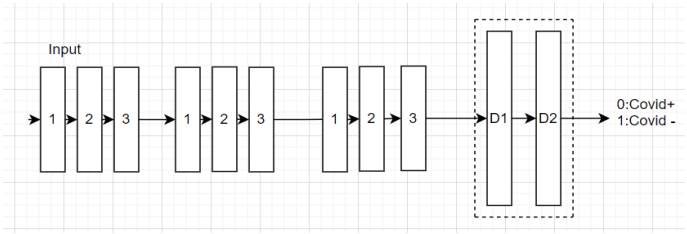


Fig 4.3.1 (a)

1-Convolutional layer
2-Activation layer
3-MaxPooling layer
D1-Dense layer[16 neurons activation function-'relu']
D2-Dense layer[1 neuron activation function-'sigmoid']

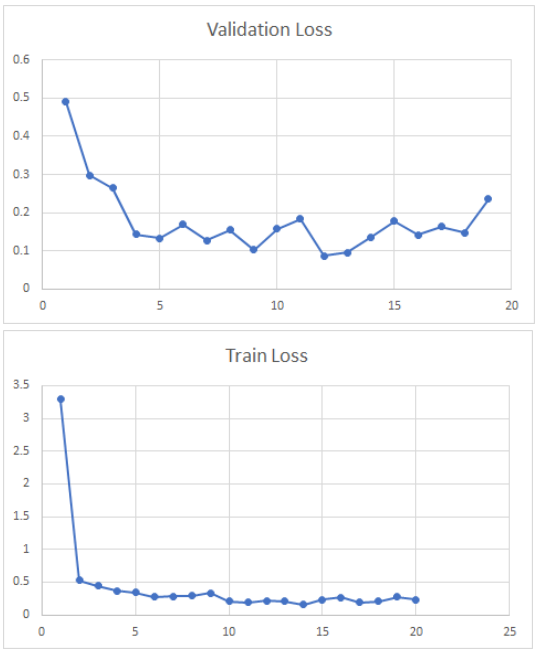


Fig 4.3.1(b)

In this architecture the validation accuracy was found to be 96.67% whereas the train accuracy was found to be 89.92% at the 20th epoch. As we can see in the diagram, the train loss and the validation loss kept changing till the very end unlike other results where it becomes stable after the first 5 epochs. The validation loss and test loss was found to be 0.1027 and 0.2359 respectively at the end of 20th epoch.

4.3.2: 3 Blocks With Skip Connection

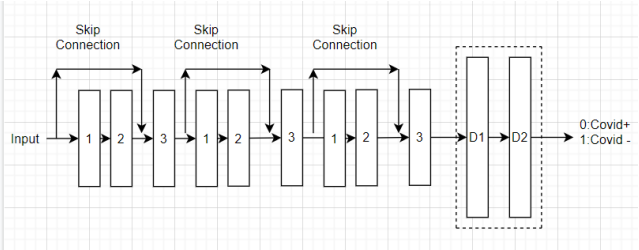
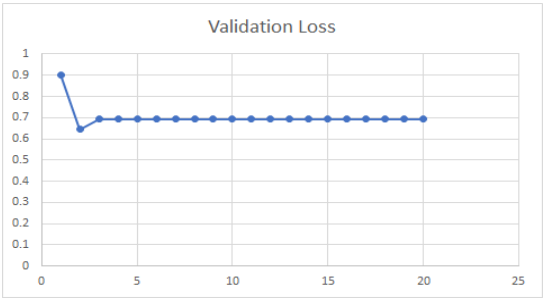


Fig 4.3.2 (a)

1-Convolutional layer
2-Activation layer
3-MaxPooling layer
D1-Dense layer[16 neurons activation function-'relu']
D2-Dense layer[1 neuron activation function-'sigmoid']



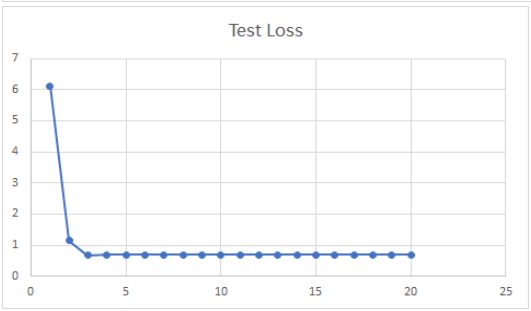


Fig 4.3.2 (b)

In this architecture the validation accuracy was found to be 51.67% whereas the train accuracy was found to be 47.3% at the 20th epoch. As we can see in the diagram, the train loss and the validation loss became stable roughly after the 3rd epoch. The validation loss and test loss was found to be 0.6931 and 0.6932 respectively at the end of 20th epoch.

The following table shows how the accuracy and the sensitivity kept increasing on further adding the network blocks, further comparing between the better approach by testing with and without the skip connection:

A Comparison of the models having different number of blocks with skip connection reveals that block 3 gives the best accuracy among these three of 51.67% and having train accuracy of 47.3% and validation loss and test loss of 0.6931 and of 0.6932 respectively.

No. of Blocks	With Skip Connection			
	Validation Accuracy	Train Accuracy	Validation Loss	Train Loss
Block 1	50%	49%	0.6932	0.6932
Block 2	50%	47.38%	0.6931	0.6931
Block 3	51.67%	47.3%	0.6931	0.6932

Table 1: With Skip Connection

Here, different models can be compared where we have not skipped the connection and can see that block 2 and 3 gives the best validation accuracy of 96.67% but the train accuracy of block 3 which is 89.2%, is better than block 2 and it is observed that it is continuously increasing on the increment of the blocks.

No. of Blocks	Without Skip Connection

	Validation Accuracy	Train Accuracy	Validation Loss	Train Loss
Block1	91.57%	74.38%	0.3762	0.5449
Block 2	96.67%	71.16%	0.1238	0.3997
Block 3	96.67%	89.2%	0.1027	0.2359

Table 2: Without Skip Connection

We saw the huge difference between the models with skip connection and without skipping the connection, so we can conclude that we should not prefer to skip the connection in order to get better validation and train accuracy.

4. DISCUSSION

In this paper a block is defined as a combination of 3 layers; convolution layer, activation layer and max pooling layer. There are (3*3) 64 filters or kernels for the convolution layer. The activation layer is that of the relu activation function and in the case of max pooling layer the pool size for max pooling is of (3*3). There are 2 dense layers for every architecture displayed above wherein the first dense layer will have 16 neurons with 'relu' activation function and the second dense layer will have 1 neuron with 'sigmoid' activation function. The first layer has a dropout of 50 percent to prevent overfitting using Adam's optimizer.

In the without skip connection case, the trained and tested model having one, two and three blocks followed by 2 dense layers, it has been observed that the train accuracy in the last three to be increasing from around 70% for 1 block and 2 blocks to about 90% in layer 3. So, in the case of train accuracy the architecture with 3 blocks has the best train accuracy. The validation accuracy also increases from a range of values from 91.67% to 96.67% for 2 and 3 blocks, whereas the test loss and validation loss declines with the increase in the number of blocks.

Now in the case of skip connection where we train and test the model having two and three blocks followed by 2 dense layers, the validation accuracy is around 50% for 2 blocks and 51.64% for 3 blocks. There is a slight increase in the validation accuracy by increasing the layers in the with skip connection architecture. The train accuracy for 2 blocks is 51.84% and for 3 blocks is 47.3%. The validation loss and the training loss remains constant at around 0.6932 and 0.6931.

5. CONCLUSION AND FUTURE WORK

As we can see from the data for a shorter architecture of CNN having less number of blocks perform better if there are no skip connections. With skip connection we get a validation accuracy of about 50 percent but without the skip connection we have a validation accuracy of above 90 percent. So, we can conclude that model without the skip connection having 3 blocks has the greatest test and validation accuracy and can be used for the detection of COVID-19, however a further study is needed to check the effect of using batch normalization after each convolution layer on the accuracy of model of the particular CNN architecture having skip connection and a further study can also reveal the influence of transfer learning and the relation between the number of layers and accuracy, to check if the accuracy keeps increasing on adding more layers.

6. References

[1] Muhammad Farooq and Abdul Hafeez et al., “COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs” Arxiv, Mar 2020, arXiv:2003.14395

[2] Ioannis D. Apostolopoulos Tzani A Mpesiana et al., “ Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks.” Arxiv, Mar 2020, arXiv:2003.11617

[3] Joseph Paul Cohen Lan Dao , Paul Morrison , Karsten Roth , Yoshua Bengio , Beiyi Shen, MD Almas Abbasi, MD Mahsa Hoshmand-Kochi, MD Marzyeh Ghassemi , Haifang Li , Tim Q Duong. et al., “Predicting COVID-19 Pneumonia Severity on Chest X-ray with Deep Learning” NCBI. Jul 2020 arXiv:2005.11856

[4] Linda Wang , Zhong Qiu Lin , and Alexander Wong et al., “COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images.” NCBI, Nov 2020 arXiv:2003.09871

[5] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[6] Maria de la Iglesia Vayá, Jose Manuel Saborit, Joaquim Angel Montell, Antonio Pertusa, Aurelia Bustos, Miguel Cazorla, Joaquin Galant, Xavier Barber, Domingo Orozco-Beltrán, Francisco García-García, Marisa Caparrós, Germán González, Jose María Salinas et al. “BIMCV COVID-19+: a large annotated dataset of RX and CT images from COVID-19 patients.” Arxiv, Jun 2020 arXiv:2006.01174

[7] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong Marzyeh Ghassemi et al. “COVID-19 Image Data Collection: Prospective Predictions are the Future.” Arxiv, Jun 2020, arXiv:2006.1198

[8] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization.” Arxiv,Oct 2016, arXiv:1610.02391