# PNEUMONIA DIAGNOSIS APPLICATION

**NAME REGISTRATION NUMBER**

KIPROTICH BRIAN RUTTO N16/03/0577/17

**A software project submitted in partial fulfillment for the requirement of award of Bachelor of Science Degree in Information and communication technology of**

**Laikipia University.**

**FEBRUARY 2022**

# DECLARATION

This software project is my original work, except where otherwise stated and has not been presented for a degree in any other University or any other Award.

**…………………………… …………………..**

**Kiprotich brian rutto Date**

#### N16/03/0577/017

# CERTIFICATION

The undersigned certify that he has read and hereby recommend for Acceptance of Laikipia

University a software project entitled: “PNEUMONIA DIAGNOSIS APPLICATION”

…………………………………… ……………………….

Dr Kingori Mindo Date

Department of Computing and Informatics

Laikipia University

# COPYRIGHT

This software project is copyright material protected under the Berne Convection, the copyright Act 1999 and other international and national enactments in that behalf, on intellectual property. It may not be reproduced by any means in full or in part except for short extracts in fair dealing so for research or private study, critical scholarly review or discourse with acknowledgement, with written permission of the Dean School of Science and Applied Technology on behalf of both the author and Laikipia University.

# DEDICATION

I dedicate this report to the almighty GOD and my loving who parents who supported me in so many ways.

# ACKNOWLEDGEMENT

special thanks and appreciation goes to the almighty GOD for all the blessings in my life. I would like to acknowledge and appreciate several people who were instrumental in ensuring that this project was a success. I can’t be thankful enough to the entire Laikipia university especially my supervisor Dr.Mindo for helping, guiding, and supporting me throughout the project completion in partial fulfillment for the requirement of the award of Bachelor of Science in information communication technology. Finally, I would like to appreciate my parents who provided support in so many ways and my friends for the continuous encouragement.

# ABSTRACT

Pneumonia is a respiratory infection caused by bacteria or viruses; it affects many individuals, especially in developing and underdeveloped nations, where high levels of pollution, unhygienic living conditions, and overcrowding are relatively common, together with inadequate medical infrastructure. Pneumonia causes pleural effusion, a condition in which fluids fill the lung, causing respiratory difficulty. Early diagnosis of pneumonia is crucial to ensure curative treatment and increase survival rates. Chest X-ray imaging is the most frequently used method for diagnosing pneumonia. However, the examination of chest X-rays is a challenging task and is prone to subjective variability. In this project, a computer-aided diagnosis system for automatic pneumonia detection using chest X-ray images is developed. Transfer learning was used to handle the scarcity of available data. The proposed approach was evaluated on two publicly available pneumonia X-ray datasets, provided by Kermany et al. and the Radiological Society of North America (RSNA). The proposed method achieved accuracy rates of 88.81. After the model was fully model was fully developed, I used a library provided by tensorflow called tensorflow js to transform the model from a python code into a native javascript code and json file. A fully functional website application was then implemented to enable anyone to use the system remotely. The web application provides an intuitive user interface that can be used by people with no programming skills.

Table of contents

[PNEUMONIA DIAGNOSIS APPLICATION 1](#_Toc97131842)

[DECLARATION 2](#_Toc97131843)

[CERTIFICATION 2](#_Toc97131844)

[COPYRIGHT 3](#_Toc97131845)

[DEDICATION 4](#_Toc97131846)

[ACKNOWLEDGEMENT 5](#_Toc97131847)

[ABSTRACT 6](#_Toc97131848)

[CHAPTER ONE 8](#_Toc97131849)

[1.1 Background information 9](#_Toc97131850)

[1.2 Problem definition 9](#_Toc97131851)

[1.3 Description of the Current System 9](#_Toc97131852)

[1.3.1 How the current system works 10](#_Toc97131853)

[1.3.2 Weakness of the current system 10](#_Toc97131854)

[1.4 Proposed solution 11](#_Toc97131855)

[1.4.1 Justification 11](#_Toc97131856)

[1.4.2 Objectives: 11](#_Toc97131857)

[CHAPTER TWO 14](#_Toc97131858)

[LITERATURE REVIEW 14](#_Toc97131859)

[2.1 Introduction 14](#_Toc97131860)

[2.2 Case studies of similar systems 14](#_Toc97131861)

[2.3 The Research Gap to be Addressed by proposed System 15](#_Toc97131862)

[CHAPTER THREE 16](#_Toc97131863)

[METHODOLOGY 16](#_Toc97131864)

[3.1 Introduction 16](#_Toc97131865)

[3.2 Software process model adopted 16](#_Toc97131866)

[3.2.1 Strengths 16](#_Toc97131867)

[3.2.2 weaknesses 17](#_Toc97131868)

[3.3 Requirement Gathering Tools 18](#_Toc97131869)

[3.4 System requirement 18](#_Toc97131870)

[3.4.1 Hardware Requirements 18](#_Toc97131871)

[3.5 Software requirements 19](#_Toc97131872)

[CHAPTER FOUR 20](#_Toc97131873)

[SYSTEM ANALYSIS AND DESIGN 20](#_Toc97131874)

[4.2 Requirement analysis 20](#_Toc97131875)

[4.3 Architectural design: 20](#_Toc97131876)

[4.4 System Analysis 21](#_Toc97131877)

[4.4.1 Context diagram 21](#_Toc97131878)

[4.4.2 Domain analysis 21](#_Toc97131879)

[4.4.3 Use case model 22](#_Toc97131880)

[4.5 System Design 22](#_Toc97131881)

[4.5.1 Sequence diagrams 22](#_Toc97131882)

[4.6 Database design 23](#_Toc97131883)

[CHAPTER FIVE 24](#_Toc97131884)

[SYSTEM IMPLEMENTATION AND TESTING 24](#_Toc97131885)

[5.1 Introduction 24](#_Toc97131886)

[5.2 Summary of the modules 25](#_Toc97131887)

[5.2.2 data loading and preprocessing 25](#_Toc97131888)

[5.2.3 model development 25](#_Toc97131889)

[5.2.3 graphical user interface- web application 26](#_Toc97131890)

[5.3 Summary of how the system works 26](#_Toc97131891)

[5.4 Test Regime 29](#_Toc97131892)

[5.4.1 screen shot of selected tests 31](#_Toc97131893)

[5.5 conclusion 31](#_Toc97131894)

[5.6 recommendations 32](#_Toc97131895)

[REFERENCES 33](#_Toc97131896)

[APPENDICES 34](#_Toc97131897)

# 

# CHAPTER ONE

INTRODUCTION

## 1.1 Background information

Pneumonia is an acute pulmonary infection that can be caused by bacteria, viruses, or fungi and infects the lungs, causing inflammation of the air sacs and pleural effusion, a condition in which the lung is filled with fluid. It accounts for more than 15% of deaths in children under the age of five years.

Deep learning is an important artificial intelligence tool, which plays a crucial role in solving many complex computer vision problems [5, 6]. Deep learning models, specifically convolutional neural networks (CNNs), are used extensively for various image classification problems

## 1.2 Problem definition

Pneumonia causes the death of around 700,000 children every year and affects 7% of the global population. Chest X-rays are primarily used for the diagnosis of this disease. However, even for a trained radiologist, it is a challenging task to examine chest X-rays.

However, even for very professional and experienced doctors, the diagnosis of pneumonia through X-ray images is still a tremendous task because X-ray images have similar region information for different diseases, such as lung cancer. Therefore, it is very time-consuming and energy-consuming to diagnose pneumonia through traditional methods and impossible to diagnose whether a patient suffers pneumonia through a standardized process.

## 1.3 Description of the Current System

Currently when a patient is suspected to be suffering from pneumonia, a qualified doctor usually administers a series of tests to ascertain the patient is infected or not.

### 1.3.1 How the current system works

one of the most effective test in taking an x-ray scan of patient chest cavity, a radiologist will the study the image carefully to predict whether the patient is suffering from pneumonia or not.

A radiologist usually looks out for known patterns that are associated with pneumonia. For one to accurately diagnose a patient he or she must be highly experienced.

### 1.3.2 Weakness of the current system

**slow**

Manually studying the x-ray radiograph takes a lot of time, this is because a doctor needs to look for all known patterns and sings of pneumonia infection.

**possibility of misdiagnosis**

several factors such exhaustion can influence radiologist ability to accurately predict a radiograph, such misdiagnosis can have serious consequences on patient’s health.

**availability**

in rural hospitals x-ray equipment are common but getting but qualified radiologist are usually not available, this hampers the process of diagnosing pneumonia.

**tedious**

x-ray patterns are very complicated, it takes a lot of effort to study them, thus it’s a highly repetitive and tedious work.

## 1.4 Proposed solution

A web application that uses computer vision to detect pneumonia on chest x-rays. The application should be accessible form common user devices such as laptops and smartphones.

### 1.4.1 Justification

As previously mentioned, pneumonia affects a large number of individuals, especially children, mostly in developing and underdeveloped countries characterized by risk factors such as overcrowding, poor hygienic conditions, and malnutrition, coupled with the unavailability of appropriate medical facilities. Early diagnosis of pneumonia is crucial to cure the disease completely. Examination of X-ray scans is the most common means of diagnosis, but it depends on the interpretative ability of the radiologist and frequently is not agreed upon by the radiologists. Thus, an automatic CAD system with generalizing capability is required to diagnose the disease. To the best of my knowledge most previous methods in the literature focused on developing models and systems to be used by technically skilled people. This work on the other hand aimed to create a simple web application that can be used by ordinary people with no programming skills.

### 1.4.2 Objectives:

|  |  |
| --- | --- |
| GENERAL OBJECTIVES | SPECIFIC OBJECTIVES |
| **patient’s data privacy** | Patient Images are never sent to an external server. Any image uploaded by client stays on client computer or mobile phone. This is because the model is running on client device.  The tech that makes this possible is called Tensorflow.js. It was created by Google |
| **speed** | the web application is able to analyze several images in seconds; this is because the model is running on the client computer or device. |
| **reduce misdiagnosis** | unlike a human being whose predictions can be affected by several factors such as execution and overworking, computer models produce standard prediction every time. |
| **easy to use** | a very simple and intuitive graphical user interface has been provided, this ensures that doctors and radiologist who have no programming skills can use it without need for any further training. |
| **can be use offline** | Once the is model downloaded, it will be cached in memory. The next time client visits this site he or she won’t need to wait but can use the previous cached model, this can enable people to use it even when their computers are offline. |
| **support common devices** | the website application support access from nearly all common user devices such as smartphones, desktop computer and laptops, this ensures that everyone is able to access the site. |

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Introduction

This chapter presents the different research and other literatures both from foreign and

local sources. The literatures of this study come from books, journals, articles, electronic materials and other online sources.

## 2.2 Case studies of similar systems

Pneumonia detection using chest X-rays has been an open problem for many years, the main limitation being the scarcity of publicly available data. Traditional machine learning methods have been explored extensively.

Chandra segmented the lung regions from chest X-ray images and extracted eight statistical characteristics from these regions, which they used to classify them. They implemented five traditional classifiers: multi-layer perceptron (MLP), random forest, sequential minimal optimization (SMO), classification via regression, and logistic regression. They evaluated their method on 412 images and achieved a 95.39% accuracy rate using the MLP classifier.

Kuo et al. [17] used 11 features to detect pneumonia in 185 schizophrenia patients. They applied these features in a large number of regression and classification models, such as decision trees, support vector machines, and logistic regression, and compared the results of the models. They achieved the highest accuracy rate, 94.5%, using a decision tree classifier; the other models fell short by large margins.

Similarly, Yue et al. [18] used 6 features to detect pneumonia in chest CT scan images of 52 patients; the best AUC value they achieved was 97%. However, these methods cannot be generalized and were evaluated on small datasets.

Sharma and Stephen devised simple CNN architectures for the classification of pneumonic chest X-ray images. They used data augmentation to compensate for the scarcity of data. Sharma et al. obtained a 90.68% and Stephen et al. a 93.73% accuracy rate on the dataset provided by Kerman et al. [4], hereafter called the Kermany dataset. Data augmentation, however, provides only a limited amount of new information from which the CNNs can learn and thus may not significantly boost their performance. Rajpukar et al. [14] used the DenseNet-121 CNN model for pneumonia classification but achieved only a 76.8% f1-score for classification. They suspected that the unavailability of patient history was a major cause for the inferior performance of both their deep learning model and the radiologists with which they compared the performance of their method.

## 2.3 The Research Gap to be Addressed by proposed System

The literature above concentrated more on developing models and algorithms to be used by professional machine learning engineers and programmers, this work on the other hand will concentrate implementing systems that can be used by normal users such as doctors and radiologist.

this will be made possible through development of web application that provides graphical interface for users.

# CHAPTER THREE

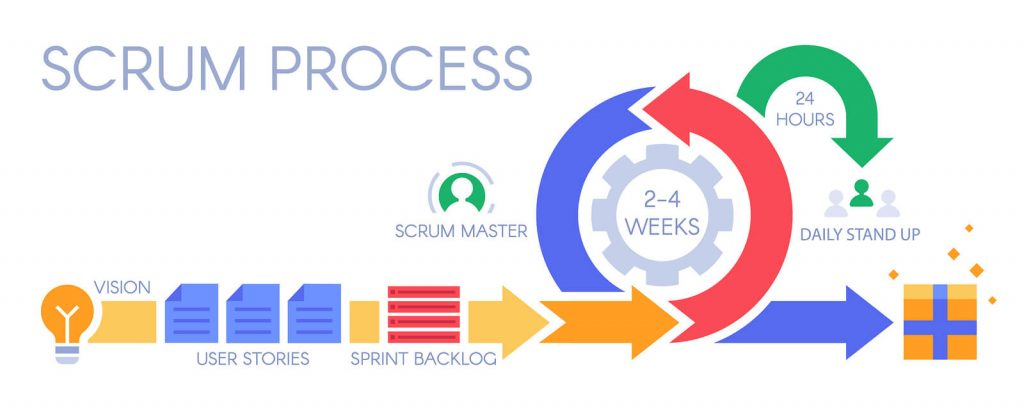
# METHODOLOGY

## 3.1 Introduction

this chapter presents the methodology used to develop

## 3.2 Software process model adopted

scrum software methodology was adopted in this project, scrum is an iterative and incremental agile software development framework for managing product development.



## 3.2.1 Strengths

**More adaptability (and less risk)**

One of the greatest benefits of agile methods is the ability to manage changing priorities. With scrums iterative approach and emphasis on continuous feedback, you can get the data you need during the development process, not after, allowing the team to make more impactful choices based on actual conditions, not just predicted conditions.

And with designated short sprint cycles, clearer project visibility, and regular reporting updates, teams can improve project predictability and reduce risk.

**Greater customer satisfaction**

one of the major benefits of scrum is that with greater customer collaboration comes greater customer satisfaction, scrum project management methodology foregrounds the customer and encourage you to work closely with them, as well as with other stakeholders, to ensure you’re creating something that actually solves their problem.

And because agile projects incorporate regular testing and review with each sprint, you can get their real feedback, in real-time, with each iteration of your working product.

**Happier teams**

scrum teams are more autonomous. That is, they’re often granted a freedom to suggest new ideas, innovate, and problem-solve that can be lacking in traditional project management methodologies. With that kind of responsibility, people are trusted to get the job done and encouraged to see themselves as integral team members who can make a tangible difference to the project’s bottom line.

### 3.2.2 weaknesses

* Scrum often leads to scope creep, due to the lack of a definite end-date
* The chances of project failure are high if individuals aren’t very committed or cooperative
* Adopting the Scrum framework in large teams is challenging
* The framework can be successful only with experienced team members
* Daily meetings sometimes frustrate team members
* If any team member leaves in the middle of a project, it can have a huge negative impact on the project
* Quality is hard to implement until the team goes through an aggressive testing process

## 3.3 Requirement Gathering Tools

#### Use Cases

Use cases provide a walkthrough of the entire product through the eyes of the end-user. This technique will help visualize how the product will actually work.

**Prototyping**

This involve building a model prototype and analyzing the completed, model, additional requirements are then added accordingly.

**Flowcharts**

Flowcharts depict sequential flow and control logic of a related set of activities. They are useful for both technical and non-technical members

## 3.4 System requirement

### 3.4.1 Hardware Requirements

* **RAM:** 1 GB and above
* **Screen size:** compatible with all screen sizes

## 3.5 Software requirements

* Works on Windows 10, XP, Vista, 7, 8, 8.1 & 11
* Works on 64 and 32 bit Windows
* Mac OS X 10.5 or above
* works on all Unix based operating systems
* Android version 2.3.3 or higher
* common browsers such as chrome, opera mini, Microsoft edge, Mozilla, internet explorer, safari and other popular browsers can be used to access the software**.**
* no special software is required to access the software.

# CHAPTER FOUR

# SYSTEM ANALYSIS AND DESIGN

**Introduction**

## 4.2 Requirement analysis

**generalization ability**

the system should be able to generalize well on new data and give accurate predictions

**speed**

the model should be able to analyze data and give results within seconds to avoid wasting radiologist time.

**easy to use**

there should be a graphical user interface that enables people with no programming skills to use the model.

**patient data privacy**

the system should ensure that patients personal data is not tampered with or stolen.

**upload more than images at a time**

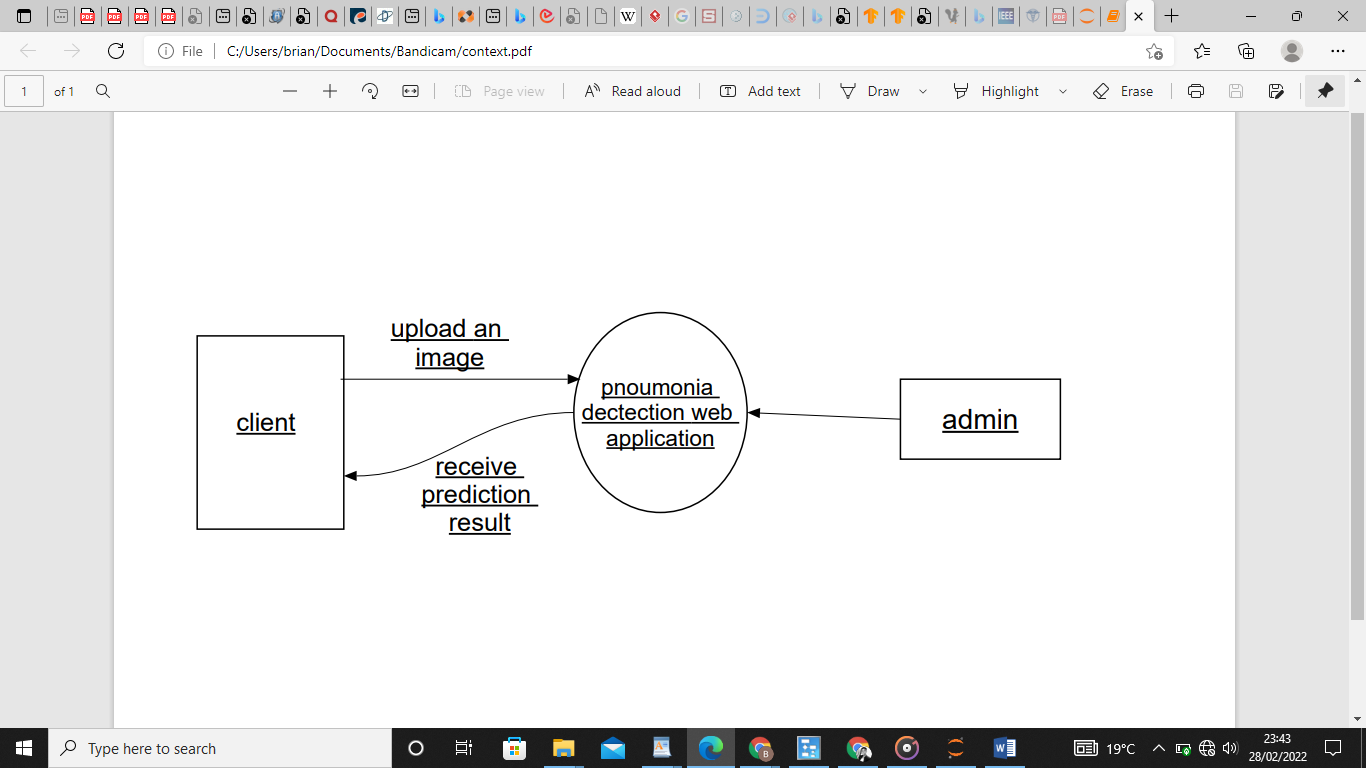
the graphical user interface should enable doctors to upload multiple images at a time.

## 4.3 Architectural design:

**Client server model was used to implement the project**

## 4.4 System Analysis

### 4.4.1 Context diagram

****

### 4.4.2 Domain analysis

Software domain analysis is the identification, analysis, and specification of common requirements from a specific application domain, typically for reuse on multiple projects within that application domain. Using image processing to diagnose pneumonia has been an open problem for many years, several papers that aim to solve this problem has been published.

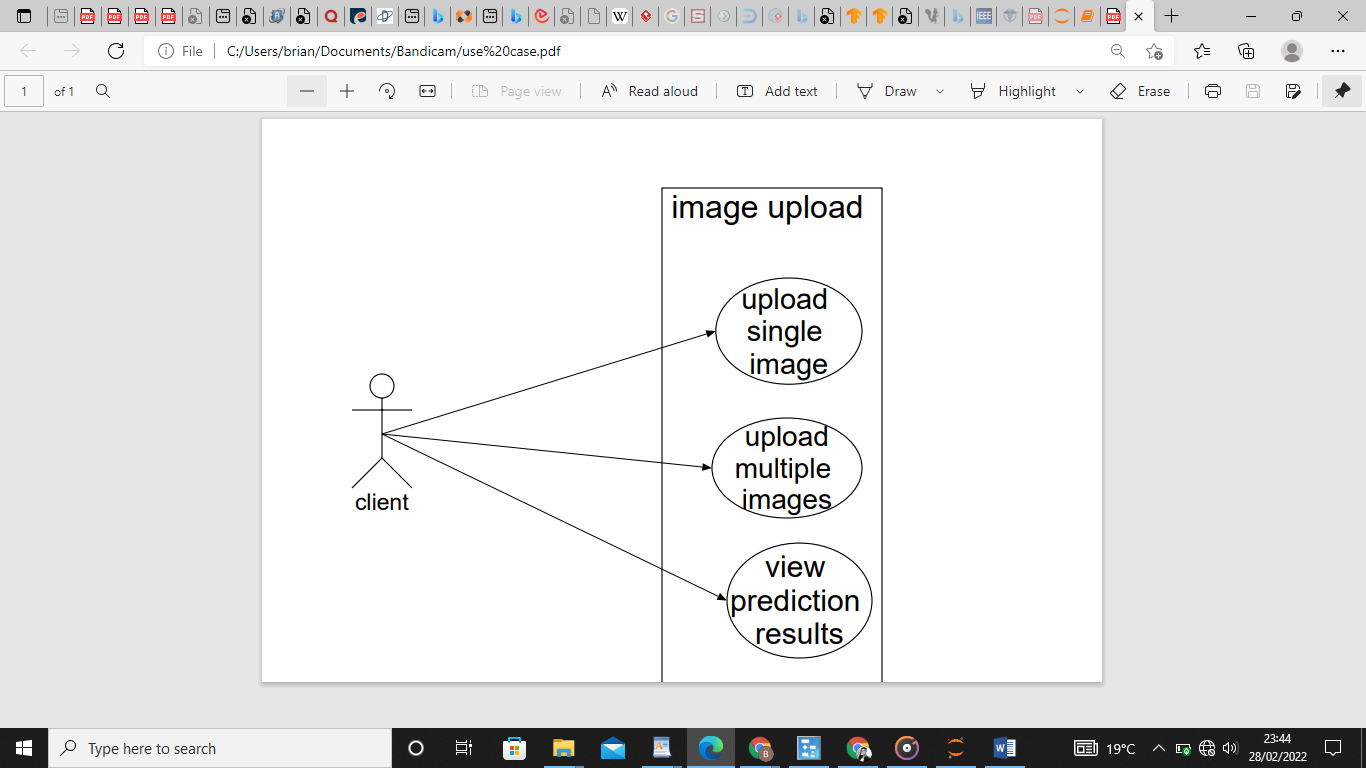
**data preprocessing**-data that will be used to trained CNN models containing several prevent features that can affect the accuracy of the model, this data needs to be cleaned before being used.

**model development**-for application to analyze x-ray images, it needs a trained deep learning model.

**deploying the model-** developed models are in python code, python being an interpreted language is difficult to be embedded in web pages. Available options to deploy the model include; developed desktop application, use python frameworks such as Django and deploy the model using pure JavaScript code.

**uploading the x-ray images-** for x-ray images to be processed by the model they should first be uploaded clients with no programming background should be provided with a graphical user interface that is simple and easy to use.

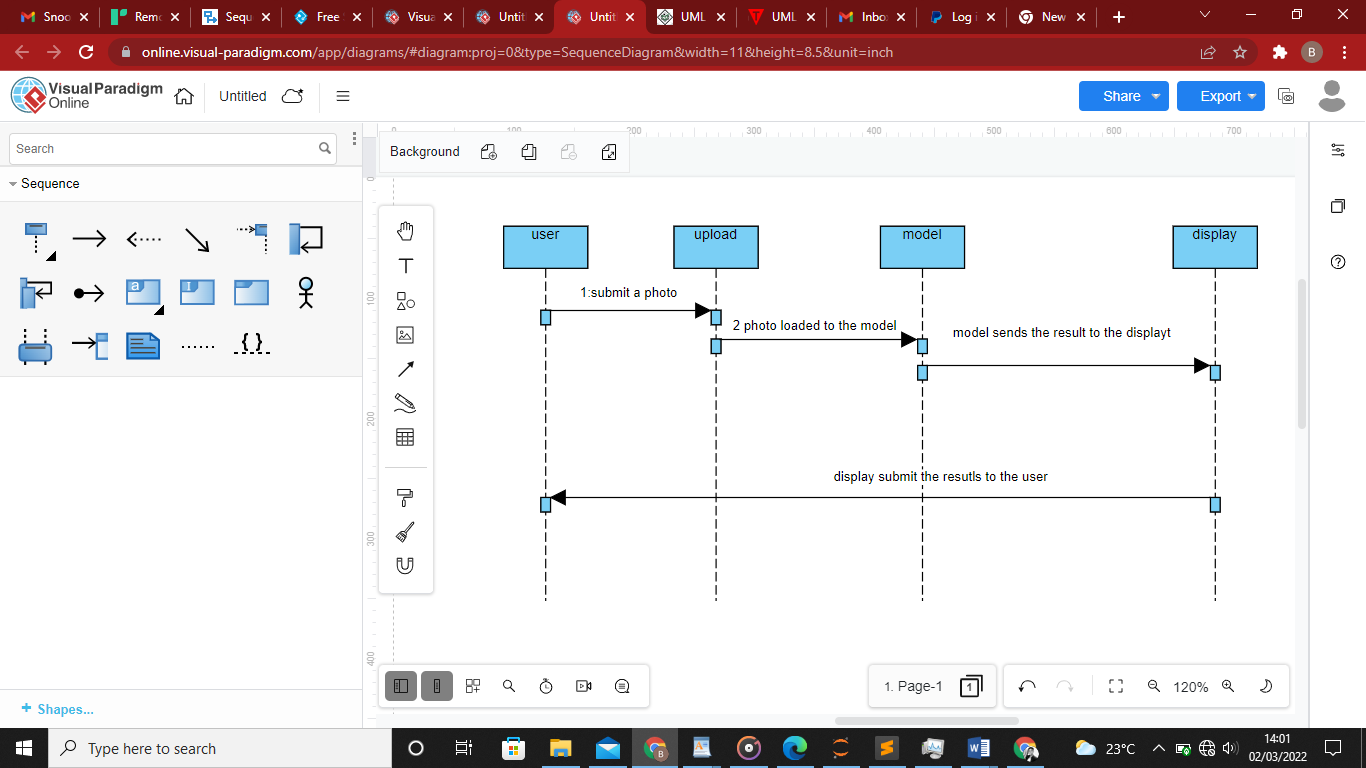
### 4.4.3 Use case model

****

## 4.5 System Design

### 4.5.1 Sequence diagrams

user



## 4.6 Database design

To maintain patient data privacy, the system does not collect any patient personal data, the application does not also require a user to sing in to use therefore database was not necessary.

# CHAPTER FIVE

# SYSTEM IMPLEMENTATION AND TESTING

## 5.1 Introduction

pneumonia analyzer is a project developed to help doctors and radiologist to diagnose pneumonia from patients. The project is not intended to replace the work of radiologist or doctor but it is intended to be used as tool to improve diagnosis speed and reduce misdiagnosis.

the project provides graphical user interface to enable non-skilled users to access it.

**tensorflow api**

Tensorflow is an open source machine learning framework developed by google, tensoflow provides all the required tools to load data, preprocess data, build models and to deploy models.

**tensorflow js**

The AI brain was developed by training CNN network using x-ray images from a hospital, the resulting model is in python code, embedding an interpreted language like python in a website affects the latency of the website, to prevent the website from running slowly, the model was transform into json file using tensorflow js api,tensorflow js is developed and maintained by google.

**JavaScript**

it was used in this project to enable users to upload images and view predictions

**python**

## 5.2 Summary of the modules

### 5.2.2 data loading and preprocessing

The proposed dataset used to evaluate model performance consists of total 5786 X-ray

images and is from the Kaggle competition. In addition, the dataset is organized into

three folders (train, test, val) and contains sub-folders for each image category (Pneumonia/Normal). All chest X-ray images (anterior-posterior) were obtained from patients age one to five years old. In order to illustrate the generalization of different models, the original dataset was randomly split into three folders, provided with train validation, and tested by 70%, 10%, and 20%.

All images were resized to a fixed shape, which in this study is a hyper-parameter

from Ω = {50, 100, 224, 300}.

### 5.2.3 model development

model or kennel is the core module of the project; it consists of a series of tensor flow sequential CNN networks. the models are trained using supervised learning algorithms.

this algorithm learns patterns from already label data. The data consist of over 800 x-ray images labelled by professional radiologist.

The kennel is made up of convolutional layers, pooling layers and activation functions.

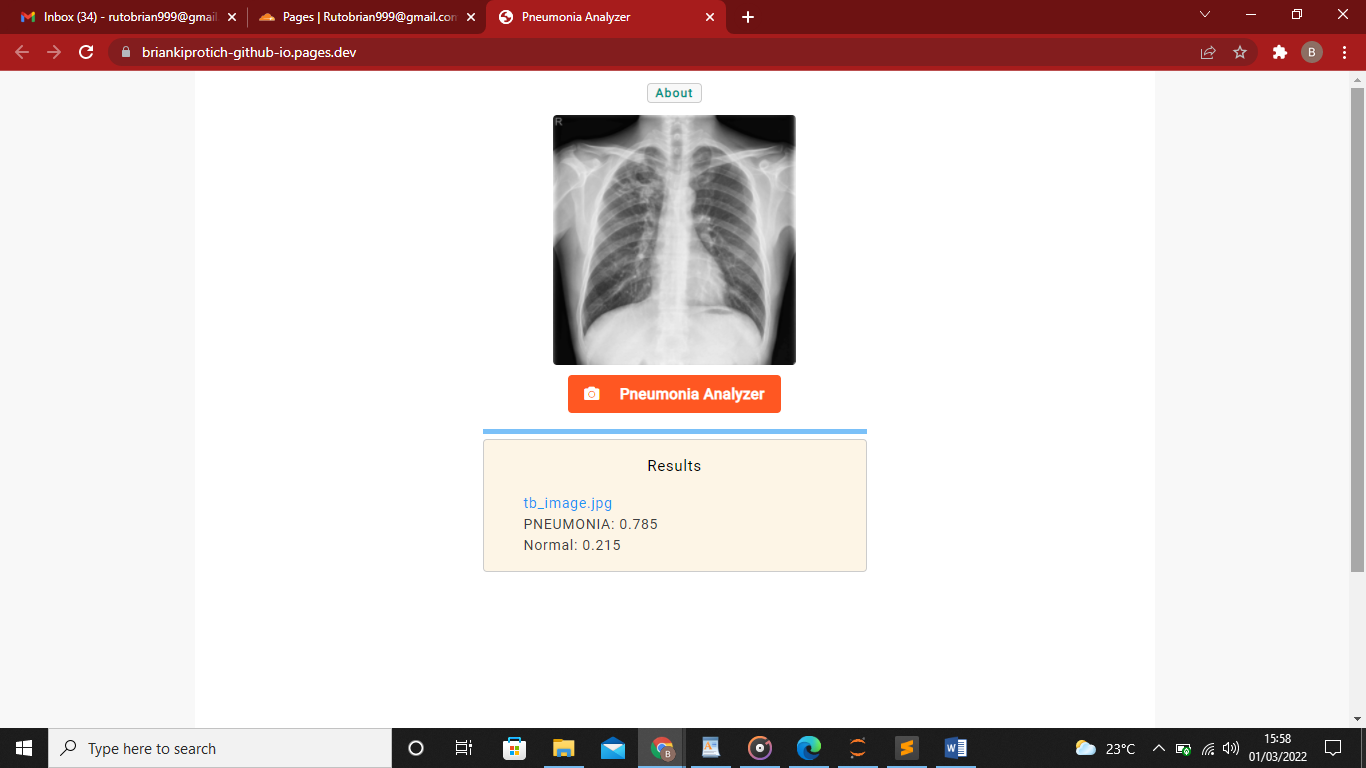
activation function used is enable the model to learn non-linear relationship, in this project soft max function was used.

## 5.2.3 graphical user interface- web application

The model was deployed using a simple web application, this enable anyone with internet access to access it without having to download any software. The application was developed using html, css and JavaScript. It provides an easy to use interface to upload images and view results.

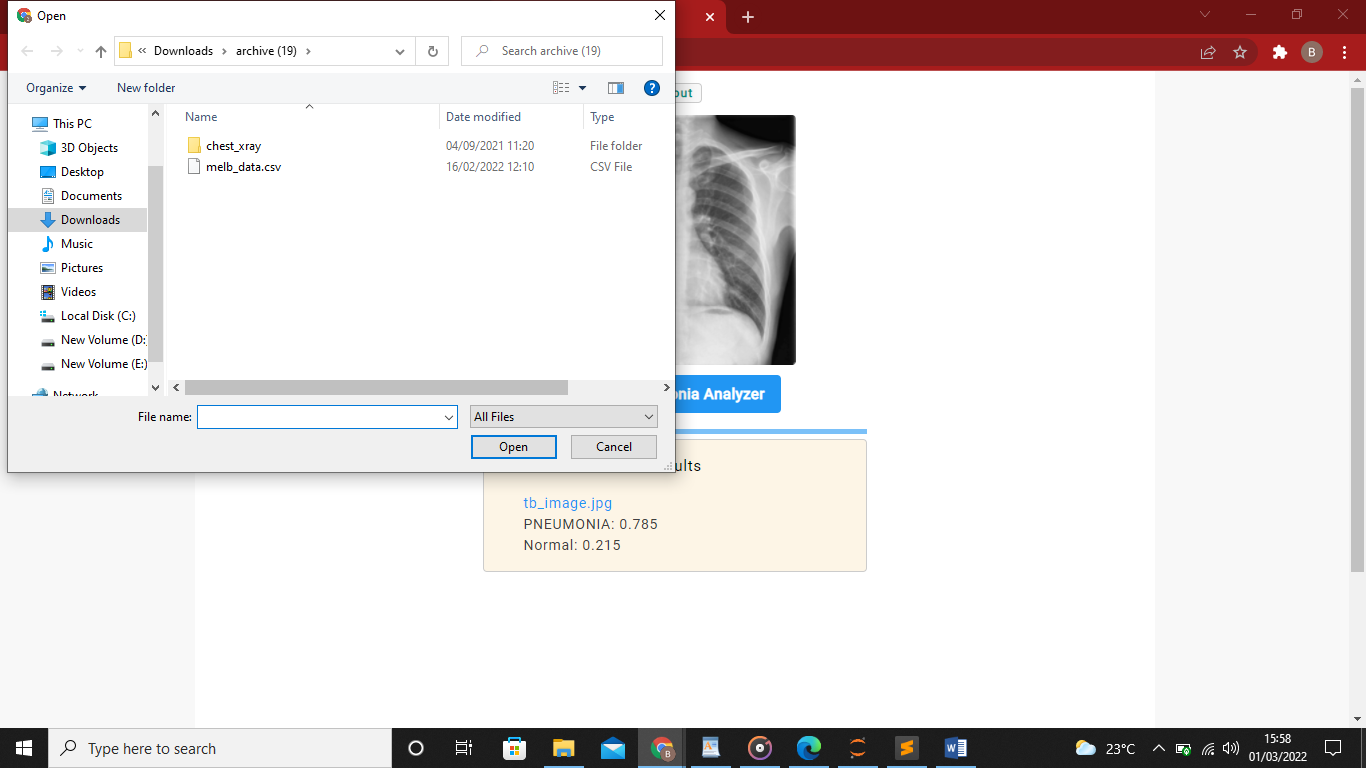
## 5.3 Summary of how the system works

Home page

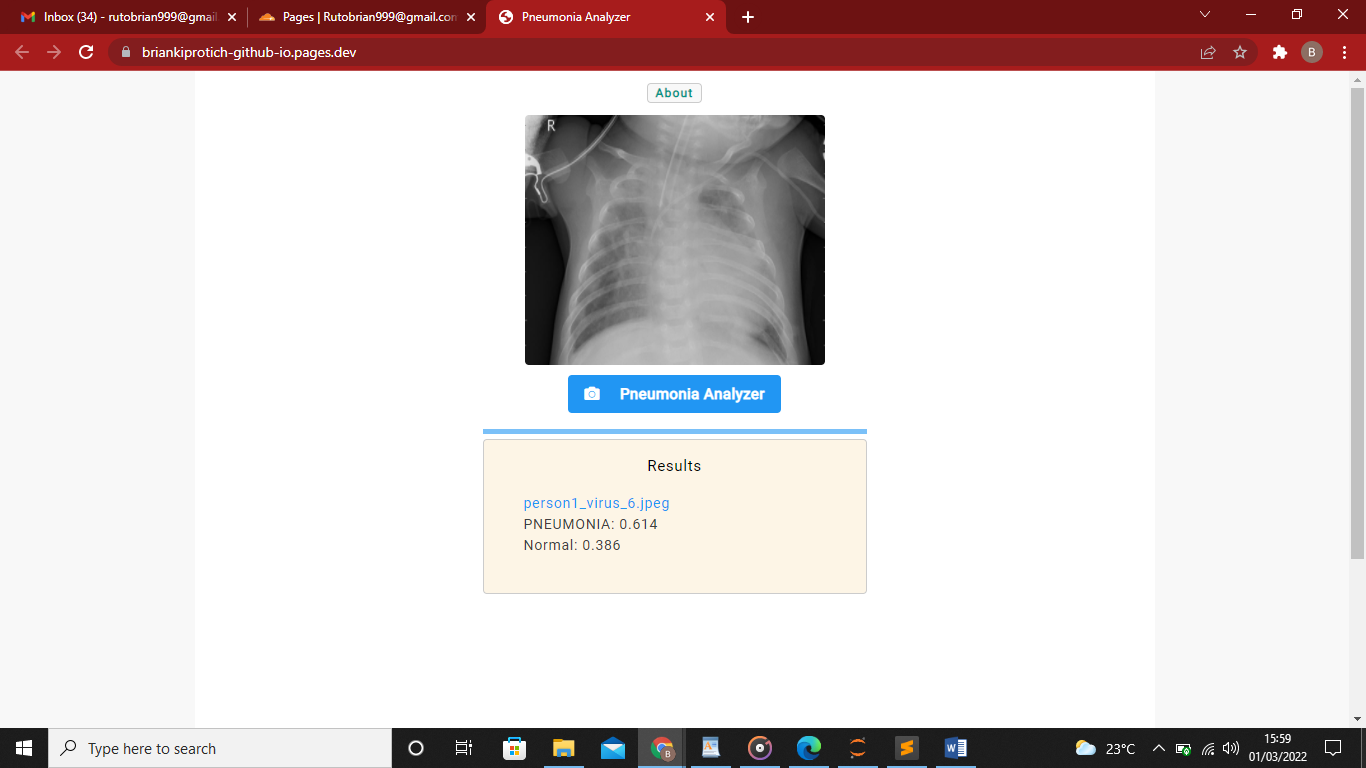


When the application is started, it automatically simulates an image analysis in progress.

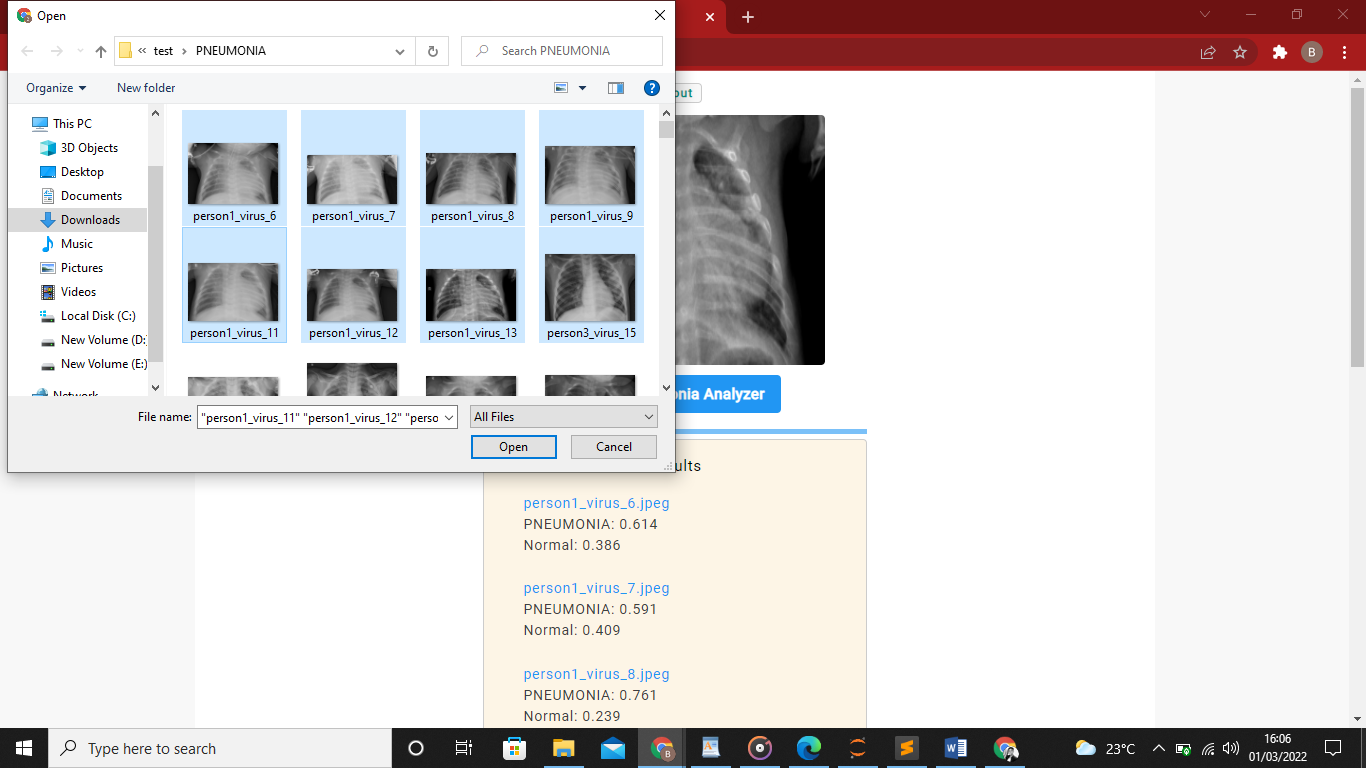
**Uploading a single image (windows 10)**



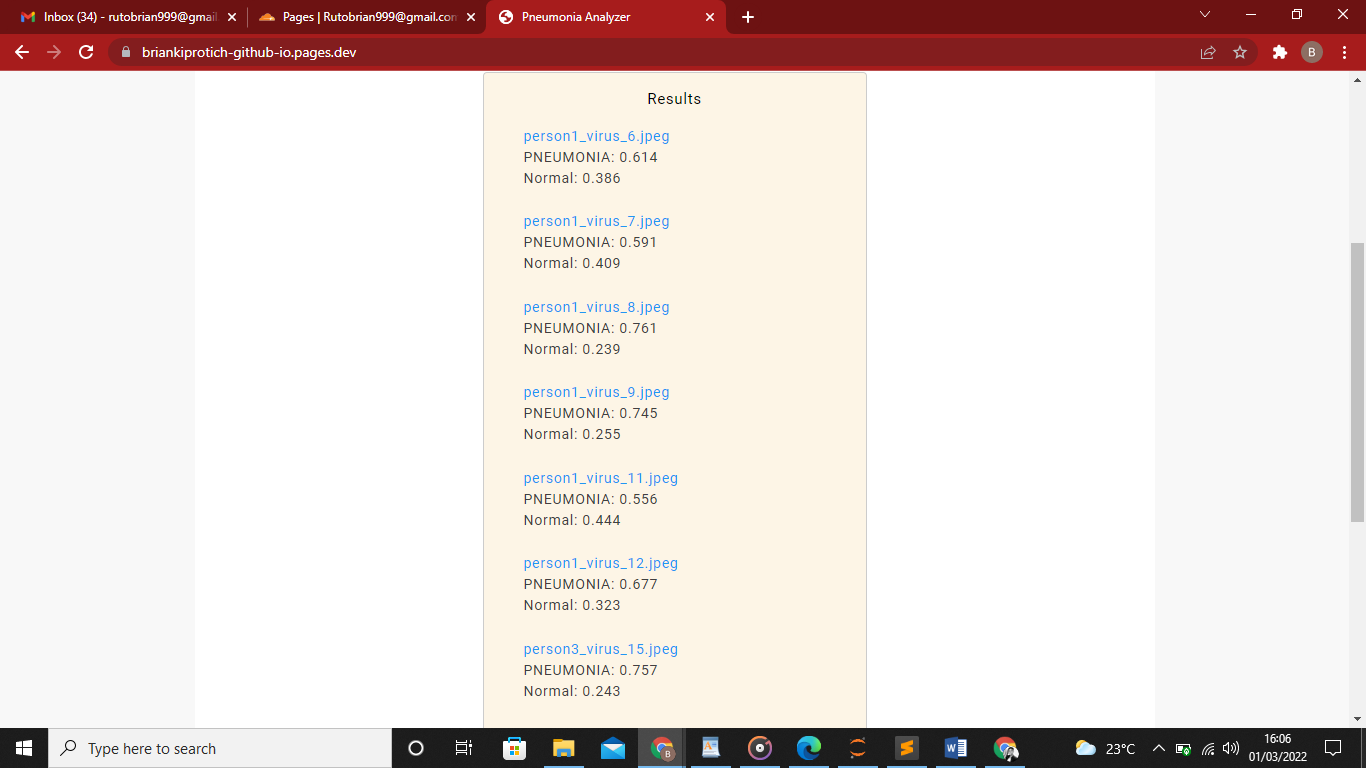
**Viewing results**



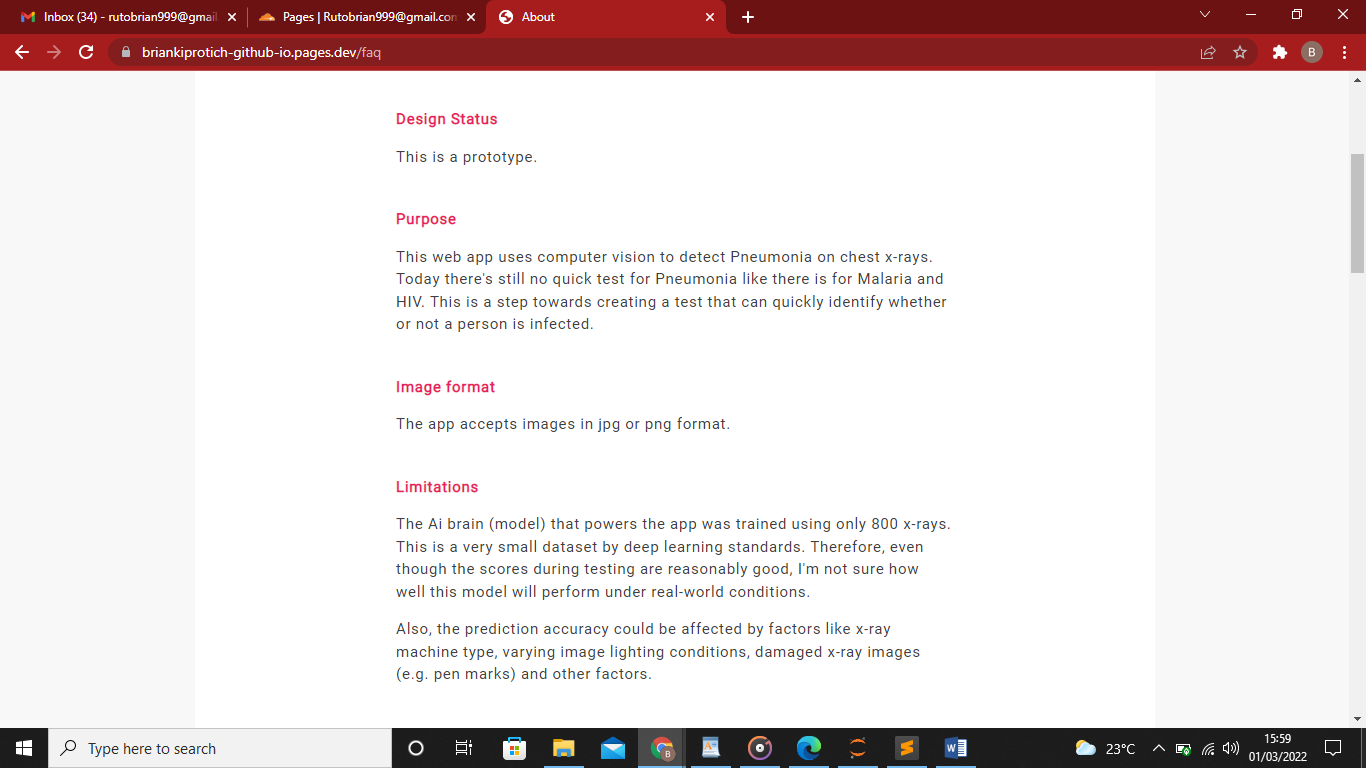
**Uploading multiple images**



**Viewing results**



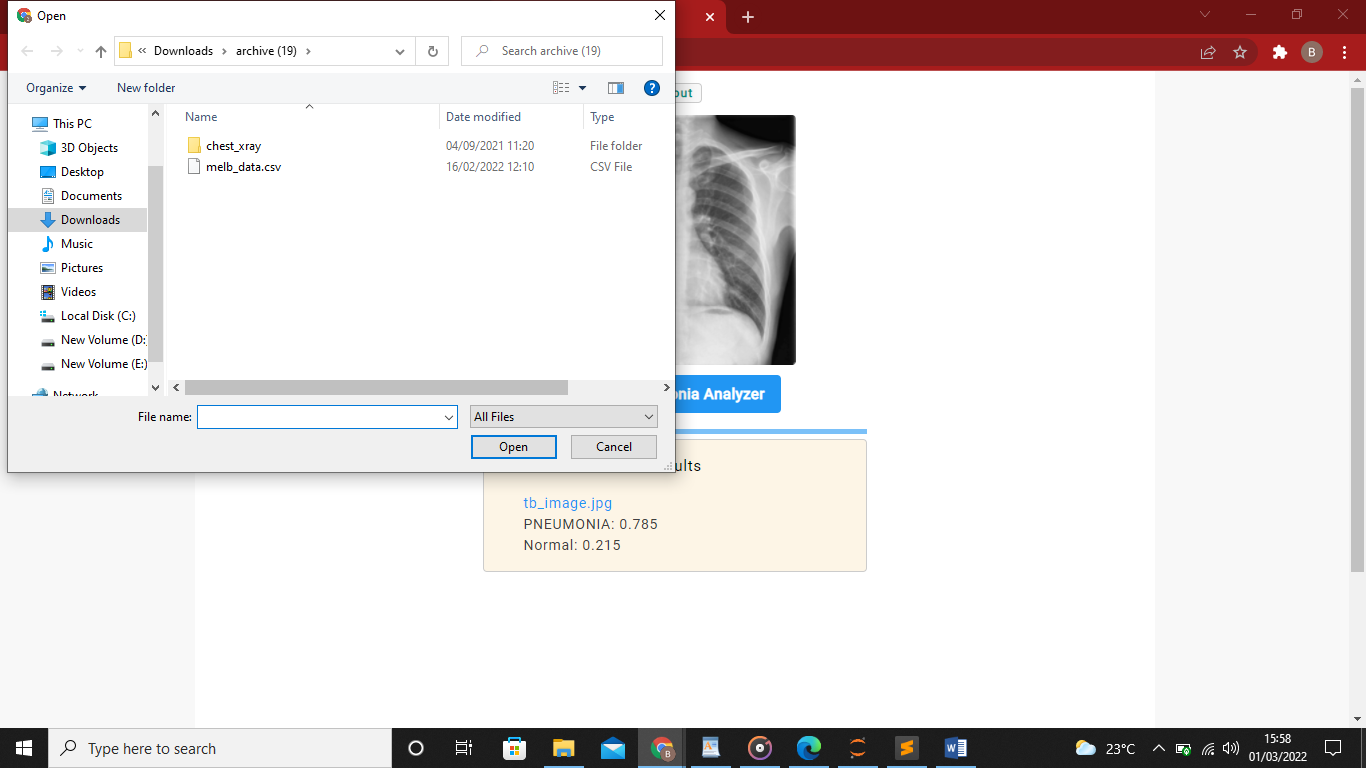
**Learning how to interpret results in about section**



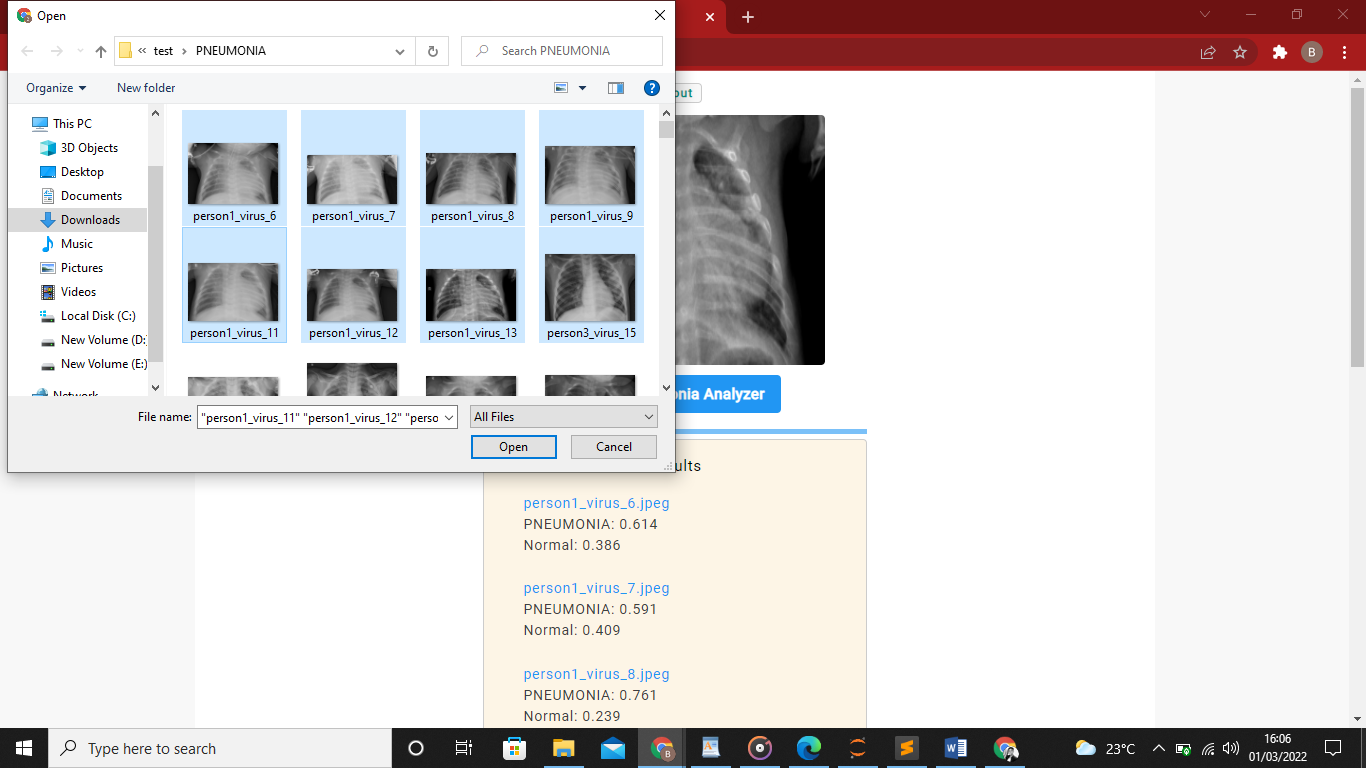
## 5.4 Test Regime

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test id | description Test | Test data | Expected results | Actual |
| 1 | Upload a single image | Sample x-ray image | Open file explorer in windows, windows, android.  Clicking a single image should enable user to upload image.  Analyze the image and display results immediately. | As expected |
| 2 | Upload multiple images |  | Open file explorer in windows, windows, android.  Enable user to select as many images as possible.  Analyze images and display the result in a list. | As expected |

### 5.4.1 screen shot of selected tests



uploading a single image



uploading multiple images

## 5.5 conclusion

This study describes a CNN-based model aiming to diagnose pneumonia on a chest X-ray image. The model has been deployed as web application and the access link is provided below, the source code for this project can be access from GitHub. The project is open source and can be used by anyone. Thank you for reading

## 5.6 recommendations

The web application is able to analyze several images within seconds and achieved accuracy of over 88 percent, therefore i recommend this project to radiologist and doctors who wishes to save time.

# REFERENCES

* Shalev & Shai.(2017).Understanding Machine Learning. London, England: Cambridge University Press
* Francois.(2019)deep learning with python. New York: oel publishers.
* Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras
* Gulli, A., & Pal, S. (2017). *Deep learning with Keras*. Packt Publishing Ltd.
* Abadi, Mart&#x27;in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning. In *12th $USENIX$ Symposium on Operating Systems Design and Implementation ($OSDI$ 16)* (pp. 265–283)

# APPENDICES

**Appendix 1: project schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S/NO. TASK | DESCRIPTION | START | END | REMARKS |
| 1 | Concept Writing | 01/10/2021 | 10/10/2021 | APPROVED |
| 2 | Proposal writing & Defense | 13/10/2021 | 18/10/2021 | APPROVED |
| 3 | Requirement Analysis | 19/10/2021 | 25/10/2021 |  |
| 4 | System Analysis And Design | 25/10/2021 | 6/11/2021 |  |
| 5 | System Implementation | 12/11/2021 | 27/12/2021 |  |
| 6 | System Testing | 28/12/2021 | 5/01/2022 |  |
| 7 | Documentation | 5/01/2022 | 15/01/2022 |  |
| 8 | PRSENTATION | 5/02/2022 | 5/02/2022 |  |

**Appendix 2: budget**

|  |  |  |
| --- | --- | --- |
| ITEM/NO | ITEM | AMOUNT(ksh) |
|  | **printing** | **600** |
|  | **photocopy** | **600** |
|  | **binding** | **1200** |
|  | **Domain name and hosting** | **1200** |
|  | **Other expenses** | **1000** |
|  | **total** | **4500** |

**Appendix 4: system code**

**JavaScript code**

**1. LOAD THE MODEL IMMEDIATELY WHEN THE PAGE LOADS**

**// Define 2 helper functions**

**function simulateClick(tabID) {**

**document.getElementById(tabID).click();**

**}**

**function predictOnLoad() {**

**// Simulate a click on the predict button**

**setTimeout(simulateClick.bind(null,'predict-button'), 500);**

**}**

**// LOAD THE MODEL**

**let model;**

**(async function () {**

**model = await tf.loadModel('https://briankiprotich-github-io.pages.dev/model\_tb\_1/model.json');**

**$("#selected-image").attr("src", "https://briankiprotich-github-io.pages.dev/assets/tb\_image.jpg");**

**// Hide the model loading spinner**

**// This line of html gets hidden:**

**// <div class="progress-bar">Ai is Loading...</div>**

**$('.progress-bar').hide();**

**// Simulate a click on the predict button.**

**// Make a prediction on the default front page image.**

**predictOnLoad();**

**})();**

**//######################################################################**

**// ### 2. MAKE A PREDICTION ON THE FRONT PAGE IMAGE WHEN THE PAGE LOADS**

**//######################################################################**

**// The model images have size 96x9**

**// This code is triggered when the predict button is clicked i.e.**

**// we simulate a click on the predict button.**

**$("#predict-button").click(async function () {**

**let image = undefined;**

**image = $('#selected-image').get(0);**

**// Pre-process the image**

**let tensor = tf.fromPixels(image)**

**.resizeNearestNeighbor([96,96]) // change the image size here**

**.toFloat()**

**.div(tf.scalar(255.0))**

**.expandDims();**

**// Pass the tensor to the model and call predict on it.**

**// Predict returns a tensor.**

**// data() loads the values of the output tensor and returns**

**// a promise of a typed array when the computation is complete.**

**// Notice the await and async keywords are used together.**

**// TARGET\_CLASSES is defined in the target\_clssses.js file.**

**// There's no need to load this file because it was imported in index.html**

**let predictions = await model.predict(tensor).data();**

**let top5 = Array.from(predictions)**

**.map(function (p, i) { // this is Array.map**

**return {**

**probability: p,**

**className: TARGET\_CLASSES[i]**

**};**

**}).sort(function (a, b) {**

**return b.probability - a.probability;**

**}).slice(0, 3);**

**// Append the file name to the prediction list**

**var file\_name = 'tb\_image.jpg';**

**$("#prediction-list").append(`<li class="w3-text-blue fname-font" style="list-style-type:none;">${file\_name}</li>`);**

**//$("#prediction-list").empty();**

**top5.forEach(function (p) {**

**// ist-style-type:none removes the numbers.**

**// https://www.w3schools.com/html/html\_lists.asp**

**$("#prediction-list").append(`<li style="list-style-type:none;">${p.className}: ${p.probability.toFixed(3)}</li>`);**

**});**

**});**

**//######################################################################**

**// ### 3. READ THE IMAGES THAT THE USER SELECTS**

**// Then direct the code execution to app\_batch\_prediction\_code.js**

**//######################################################################**

**// This listens for a change. It fires when the user submits images.**

**$("#image-selector").change(async function () {**

**// the FileReader reads one image at a time**

**fileList = $("#image-selector").prop('files');**

**//$("#prediction-list").empty();**

**// Start predicting**

**// This function is in the app\_batch\_prediction\_code.js file.**

**model\_processArray(fileList);**

**});**

**import numpy as np**

**import os,sys**

**import matplotlib.pyplot as plt**

**import tensorflow as tf**

**from PIL import Image**

**from tensorflow.keras.preprocessing.image import ImageDataGenerator,load\_img**

**from tensorflow.keras.models import Sequential**

**from tensorflow import keras**

**import pandas as pd**

**from tensorflow.keras import layers**

**import itertools**

**#the next thing is to find directories where the data has been stored,**

**train\_folder='C:/Users/brian/Pictures/chest\_xray/train/'**

**test\_folder='C:/Users/brian/Pictures/chest\_xray/test/'**

**val\_folder='C:/Users/brian/Pictures/chest\_xray/val/'**

**print(len(os.listdir(test\_folder)))**

**print(train\_folder)**

**print(os.listdir(train\_folder))**

**random\_pic=np.random.randint(0,len(os.listdir(nomal\_pic)))**

**print(random\_pic)**

**rand=os.listdir(nomal\_pic)[random\_pic]**

**normaladdr=nomal\_pic+rand**

**print(normaladdr)**

**random\_pic\_sic=np.random.randint(0,len(os.listdir(pneumonia\_pic)))**

**print(random\_pic\_sic)**

**rando=os.listdir(pneumonia\_pic)[random\_pic\_sic]**

**sicaddress=pneumonia\_pic+rando**

**print(rando)**

**#now how do we select a rondom picture and print it at screen**

**sicpic=Image.open(sicaddress)**

**normalpic=Image.open(normaladdr)**

**fig1=plt.figure(figsize=(10,7))**

**plt.imshow(normalpic,cmap='gray',vmin=0,vmax=255)#to show an imagwe as gray use>>> cmap='gray',vmin=0,vmax=25**

**fig2=plt.figure(figsize=(10,7))**

**plt.imshow(sicpic,cmap='gray',vmin=0,vmax=255)#to show an imagwe as gray use>>> cmap='gray',vmin=0,vmax=255**

**num\_of\_test\_samples=600**

**bach\_size=32**

**train\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True)**

**test\_datagen=ImageDataGenerator(rescale=1./255)**

**train=train\_datagen.flow\_from\_directory('C:/Users/brian/Pictures/chest\_xray/train/',target\_size=(64,64),batch\_size=32,class\_mode='binary')**

**test=test\_datagen.flow\_from\_directory('C:/Users/brian/Pictures/chest\_xray/test/',target\_size=(64,64),batch\_size=32,class\_mode='binary')**

**val=train\_datagen.flow\_from\_directory('C:/Users/brian/Pictures/chest\_xray/val/',target\_size=(64,64),batch\_size=32,class\_mode='binary')**

**model=Sequential([**

**# layers.experimental.preprocessing.Rescaling(1./255,input\_shape=(64,64,3)),**

**layers.Conv2D(32,3,activation='relu'),**

**layers.MaxPooling2D(pool\_size=(2,2)),**

**layers.Conv2D(32,3,activation='relu'),**

**layers.MaxPooling2D(pool\_size=(2,2)),**

**layers.Flatten(),**

**layers.Dense(128,activation='relu'),**

**layers.Dense(1,activation='sigmoid')**

**])**

**model.compile(optimizer='adam',loss='binary\_crossentropy',metrics=['accuracy'])**

**history=model.fit(train,validation\_data=val,epochs=10,steps\_per\_epoch=5,)**

**!mkdir -p saved\_model**

**model.save('saved\_model/my\_model')**

**import tensorflowjs as tfjs**

**my\_mode='C:/Users/brian/seaborn/saved\_model/my\_model'**

**def importModel(modelPath):**

**model = keras.models.load\_model(modelPath)**

**tfjs.converters.save\_keras\_model(model, "tfjsmodel")**

**importModel(my\_mode)**

**y\_pred=model.predict(test,100)**

**y\_pred=np.argmax(y\_pred,axis=1)**

**test\_accu=model.evaluate(test)**

**history.history**

**#nomal\_pic=train\_folder+'NORMAL/'**

**plt.plot(history.history['accuracy'])**

**plt.plot(history.history['val\_accuracy'])**

**plt.plot(history.history['loss'])**

**plt.plot(history.history['val\_loss'])**

**plt.title('model accuracy')**

**plt.ylabel('accuracy')**

**plt.xlabel('epoch')**

**plt.legend(['accuracy','val\_accuracy','loss','val\_loss'],loc='upper left')**

**plt.show()**

**// After the model loads we want to make a prediction on the default image.**

**// Thus, the user will see predictions when the page is first loaded.**

**function simulateClick(tabID) {**

**document.getElementById(tabID).click();**

**}**

**function predictOnLoad() {**

**// Simulate a click on the predict button**

**setTimeout(simulateClick.bind(null,'predict-button'), 500);**

**};**

**$("#image-selector").change(function () {**

**let reader = new FileReader();**

**reader.onload = function () {**

**let dataURL = reader.result;**

**$("#selected-image").attr("src", dataURL);**

**$("#prediction-list").empty();**

**}**

**let file = $("#image-selector").prop('files')[0];**

**reader.readAsDataURL(file);**

**// Simulate a click on the predict button**

**// This introduces a 0.5 second delay before the click.**

**// Without this long delay the model loads but may not automatically**

**// predict.**

**setTimeout(simulateClick.bind(null,'predict-button'), 500);**

**});**

**let model;**

**(async function () {**

**model = await tf.loadModel('https://briankiprotich-github-io.pages.dev/model\_tb\_1/model.json');**

**$("#selected-image").attr("src", "https://briankiprotich-github-io.pages.dev/assets/tb.jpg")**

**// Hide the model loading spinner**

**$('.progress-bar').hide();**

**// Simulate a click on the predict button**

**predictOnLoad();**