COVID-19 PREDICTION BY CHEST X-RAYS USING DEEP LEARNING

PROJECT DEMO =DEMO

INTRODUCTION

Project Overview

The "Covid-19 Detection using Deep Learning with Chest X-rays" project aims to harness the power of deep learning algorithms to facilitate early and accurate diagnosis of Covid-19 by analyzing chest X-ray images. The project addresses the critical need for efficient and rapid screening of potential Covid-19 cases, providing a valuable tool.

Purpose

The purpose of this project is to develop a robust and accurate model for early detection of COVID-19 through analysis of chest X-rays.

LITERATURE REVIEW

Existing Problems

Current methodologies for Covid-19 detection vary in accuracy and efficiency. Some challenges include:
***Limited Dataset Availability*: Many studies face challenges due to the scarcity of diverse and well-annotated datasets for training deep learning models.**
***Interpretability of Models*: Understanding and interpreting the decisions made by deep learning models in the medical domain, especially in the context of chest X-ray analysis, is a critical concern.**

Transferability and Generalization**: Ensuring the model's ability to generalize across different populations, demographics, and imaging equipment is crucial for real-world applicability.**

References

"Deep Learning-based Detection for COVID-19 from Chest X-ray Images"
Authors: Wang et al., Year: 2020
This seminal work explores the application of deep learning techniques for Covid-19 detection using chest X-ray images. The study introduces a convolutional neural network (CNN) architecture and discusses its performance in terms of sensitivity and specificity.

"COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-ray Images"
Authors: Wang et al., Year: 2020
This research introduces COVID-Net, a specialized deep learning model for Covid-19 detection. The study focuses on optimizing model architecture for accuracy and generalization across different datasets.

"Transfer Learning with Convolutional Neural Networks for COVID-19 Detection using Chest X-rays"
Authors: Narin et al., Year: 2020
The paper investigates the effectiveness of transfer learning in training deep neural networks for Covid-19 detection. It assesses the model's ability to adapt to limited datasets and varying imaging conditions.

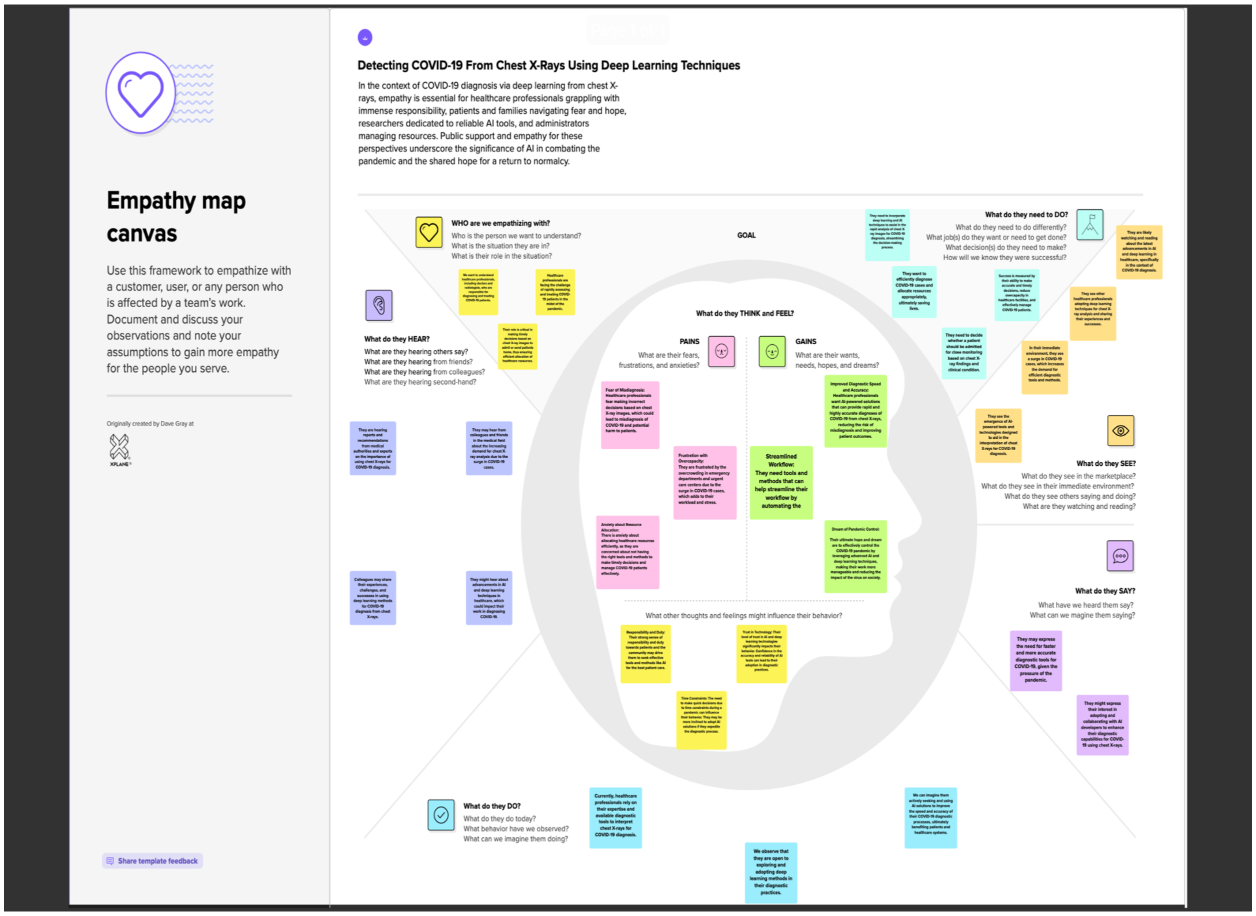
"Explainable Deep Learning Models in Medical Image Analysis"
Authors: Mahmud et al., Year: 2021
Exploring the interpretability of deep learning models, this review discusses methods to make predictions more understandable for medical professionals, an important aspect in the deployment of such systems in clinical settings.

<u>Problem Statement Definition</u>

Building upon the insights from the literature survey, the specific problems targeted by this project include improving the interpretability of deep learning models, enhancing generalization across diverse datasets, and addressing challenges related to the scarcity of annotated Covid-19 chest X-ray images. The project aims to contribute solutions to these issues through the development of an advanced and interpretable deep learning model for accurate Covid-19 detection from chest X-rays.

IDEATION PHASE

Project Empathy Map Link : LINK



Empathy map canvas

Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.

Originally created by Dave Gray at XPLANE

Share template feedback

Detecting COVID-19 From Chest X-Rays Using Deep Learning Techniques

In the context of COVID-19 diagnosis via deep learning from chest X-rays, empathy is essential for healthcare professionals grappling with immense responsibility, patients and families navigating fear and hope, researchers dedicated to reliable AI tools, and administrators managing resources. Public support and empathy for these perspectives underscore the significance of AI in combating the pandemic and the shared hope for a return to normalcy.

WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

- We want to understand healthcare professionals, including doctors and radiologists, who are responsible for diagnosing and treating COVID-19 patients.
- Healthcare professionals are facing the challenge of rapidly assessing and treating COVID-19 patients in the midst of the pandemic.
- Their role is critical in making timely decisions based on chest X-ray images to admit or send patients home, thus ensuring efficient allocation of healthcare resources.

GOAL

What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?

- They need to incorporate deep learning and AI techniques to assist in the rapid analysis of chest X-ray images for COVID-19 diagnosis, streamlining the decision-making process.
- They want to efficiently diagnose COVID-19 cases and allocate resources appropriately, ultimately saving lives.
- Success is measured by their ability to make accurate and timely decisions, reduce overcapacity in healthcare facilities, and effectively manage COVID-19 patients.
- They need to decide whether a patient should be admitted for close monitoring based on chest X-ray findings and clinical condition.

What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?

- They are hearing reports and recommendations from medical authorities and experts on the importance of using chest X-rays for COVID-19 diagnosis.
- They may hear from colleagues and friends in the medical field about the increasing demand for chest X-ray analysis due to the surge in COVID-19 cases.
- Colleagues may share their experiences, challenges, and successes in using deep learning methods for COVID-19 diagnosis from chest X-rays.
- They might hear about advancements in AI and deep learning techniques in healthcare, which could impact their work in diagnosing COVID-19.

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

- Fear of Misdiagnosis: Healthcare professionals fear making incorrect decisions based on chest X-ray images, which could lead to misdiagnosis of COVID-19 and potential harm to patients.
- Frustration with Overcapacity: They are frustrated by the overcrowding in emergency departments and urgent care centers due to the surge in COVID-19 cases, which adds to their workload and stress.
- Anxiety about Resource Allocation: There is anxiety about allocating healthcare resources efficiently, as they are concerned about not having the right tools and methods to make timely decisions and manage COVID-19 patients effectively.

GAINS
What are their wants, needs, hopes, and dreams?

- Improved Diagnostic Speed and Accuracy: Healthcare professionals want AI-powered solutions that can provide rapid and highly accurate diagnoses of COVID-19 from chest X-rays, reducing the risk of misdiagnosis and improving patient outcomes.
- Streamlined Workflow: They need tools and methods that can help streamline their workflow by automating the
- Dream of Pandemic Control: Their ultimate hope and dream are to effectively control the COVID-19 pandemic by leveraging advanced AI and deep learning techniques, making their work more manageable and reducing the impact of the virus on society.

What other thoughts and feelings might influence their behavior?

- Responsibility and Duty: Their strong sense of responsibility and duty towards patients and the community may drive them to seek effective tools and methods like AI for the best patient care.
- Trust in Technology: Their level of trust in AI and deep learning technologies significantly impacts their behavior. Confidence in the accuracy and reliability of AI tools can lead to their adoption in diagnostic practices.
- Time Constraints: The need to make quick decisions due to time constraints during a pandemic can influence their behavior. They may be more inclined to adopt AI solutions if they expedite the diagnostic process.

What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?

- They are likely watching and reading about the latest advancements in AI and deep learning in healthcare, specifically in the context of COVID-19 diagnosis.
- They see other healthcare professionals adopting deep learning techniques for chest X-ray analysis and sharing their experiences and successes.
- In their immediate environment, they see a surge in COVID-19 cases, which increases the demand for efficient diagnostic tools and methods.
- They see the emergence of AI-powered tools and technologies designed to aid in the interpretation of chest X-rays for COVID-19 diagnosis.

What do they SAY?
What have we heard them say?
What can we magine them saying?

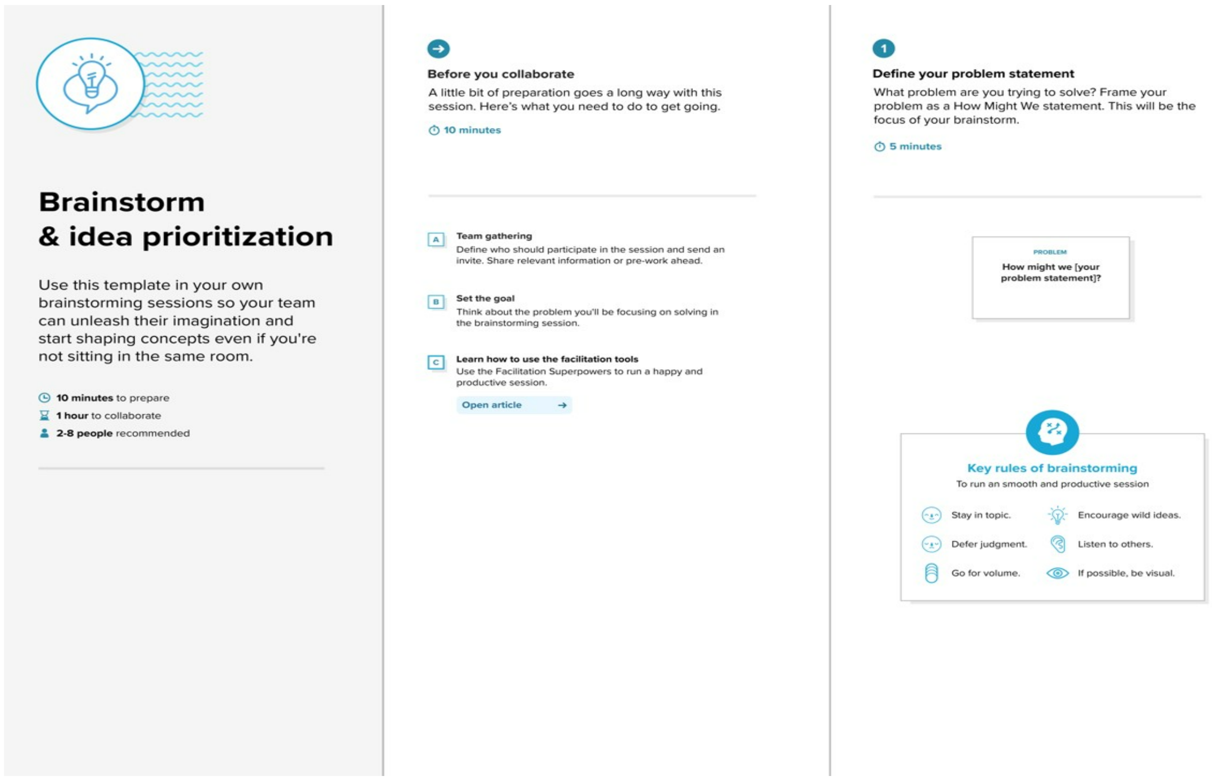
- They may express the need for faster and more accurate diagnostic tools for COVID-19, given the pressure of the pandemic.
- They might express their interest in adopting and collaborating with AI developers to enhance their diagnostic capabilities for COVID-19 using chest X-rays.

What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?

- Currently, healthcare professionals rely on their expertise and available diagnostic tools to interpret chest X-rays for COVID-19 diagnosis.
- We observe that they are open to exploring and adopting deep learning methods in their diagnostic practices.
- We can imagine them actively seeking and using AI solutions to improve the speed and accuracy of their COVID-19 diagnostic processes, ultimately benefiting patients and healthcare systems.

Brainstorm Link : [LINK]()

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm
& idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

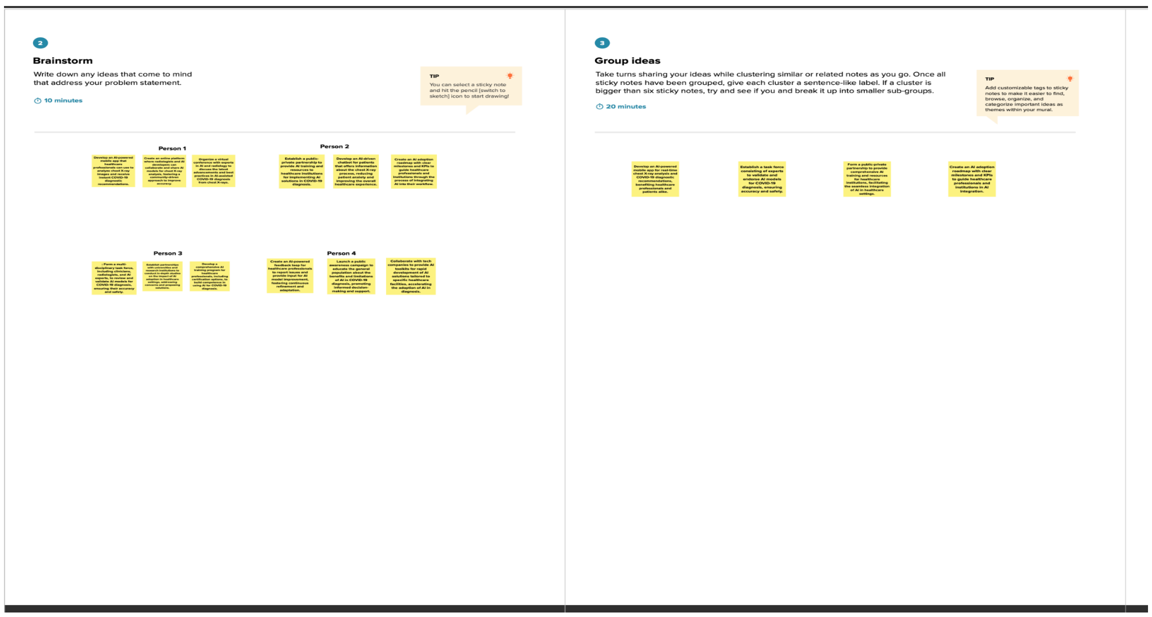
5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

Step-2: Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

- Develop an AI-powered mobile app that healthcare professionals can use to analyze chest X-ray images and receive instant COVID-19 diagnostic recommendations.
- Create an online platform where radiologists and AI developers can collaborate and share AI models for chest X-ray analysis, fostering a community-driven approach to improve accuracy.
- Organize a virtual conference with experts in AI and radiology to discuss the latest advancements and best practices in AI-assisted COVID-19 diagnosis from chest X-rays.

Person 2

- Establish a public-private partnership to provide AI training and resources to healthcare institutions for implementing AI solutions in COVID-19 diagnosis.
- Develop an AI-driven chatbot for patients that offers information about the chest X-ray process, reducing patient anxiety and improving the overall healthcare experience.
- Create an AI adoption roadmap with clear milestones and KPIs to guide healthcare professionals and institutions through the process of integrating AI into their workflow.

Person 3

- : Form a multi-disciplinary task force, including clinicians, radiologists, and AI experts, to review and validate AI models for COVID-19 diagnosis, ensuring their accuracy and safety.
- Establish partnerships with universities and research institutions to conduct in-depth studies on the impact of AI adoption in healthcare settings, addressing concerns and proposing solutions.
- Develop a comprehensive AI training program for healthcare professionals, including certification options, to build competence in using AI for COVID-19 diagnosis.

Person 4

- Create an AI-powered feedback loop for healthcare professionals to report issues and provide input for AI model improvement, fostering continuous refinement and adaptation.
- Launch a public awareness campaign to educate the general population about the benefits and limitations of AI in COVID-19 diagnosis, promoting informed decision-making and support.
- Collaborate with tech companies to provide AI toolkits for rapid development of AI solutions tailored to specific healthcare facilities, accelerating the adoption of AI in diagnosis.

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

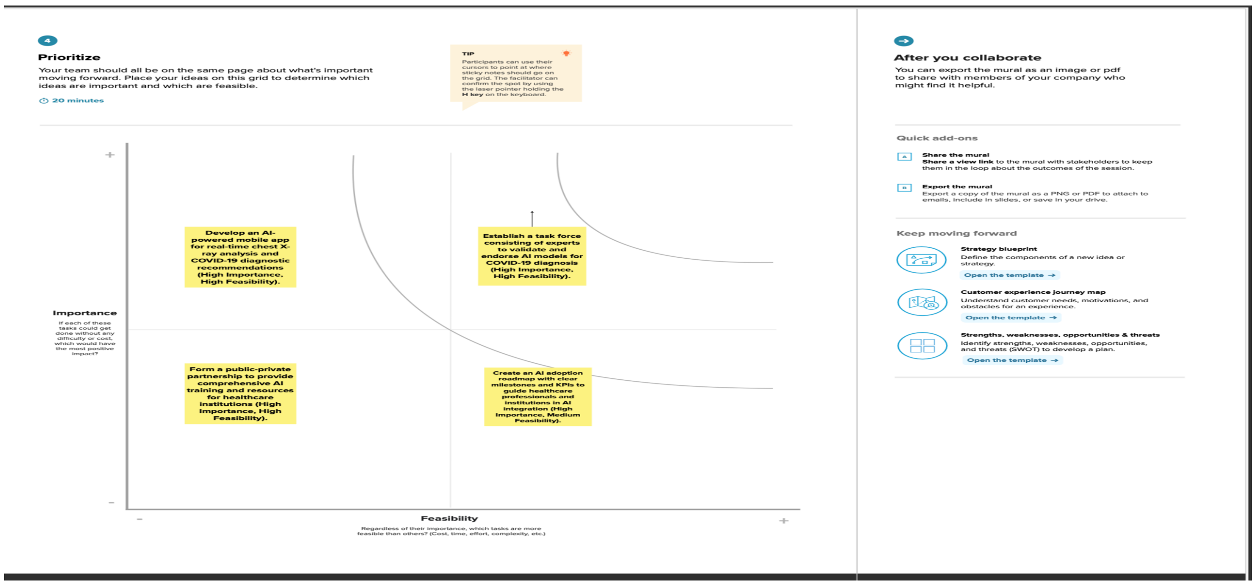
20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

- Develop an AI-powered mobile app for real-time chest X-ray analysis and COVID-19 diagnostic recommendations, benefiting healthcare professionals and patients alike.
- Establish a task force consisting of experts to validate and endorse AI models for COVID-19 diagnosis, ensuring accuracy and safety.
- Form a public-private partnership to provide comprehensive AI training and resources for healthcare institutions, facilitating the seamless integration of AI in healthcare settings.
- Create an AI adoption roadmap with clear milestones and KPIs to guide healthcare professionals and institutions in AI integration.

Step-3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

- Develop an AI-powered mobile app for real-time chest X-ray analysis and COVID-19 diagnostic recommendations (High Importance, High Feasibility).
- Establish a task force consisting of experts to validate and endorse AI models for COVID-19 diagnosis (High Importance, High Feasibility).
- Form a public-private partnership to provide comprehensive AI training and resources for healthcare institutions (High Importance, High Feasibility).
- Create an AI adoption roadmap with clear milestones and KPIs to guide healthcare professionals and institutions in AI integration (High Importance, Medium Feasibility).

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

PROJECT DESIGN PHASE

PROPOSED SOLUTION

Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**To develop a robust and accurate model for early detection of COVID-19 through analysis of chest X-rays.**
2.	Idea / Solution description	**The "Covid-19 Detection using Deep Learning with Chest X-rays" project aims to harness the power of deep learning algorithms to facilitate early and accurate diagnosis of Covid-19 by analyzing chest X-ray images. The project addresses the critical need for efficient and rapid screening of potential Covid-19 cases, providing a valuable tool**.
3.	Novelty / Uniqueness	**Our solution stands out due to its use of cutting-edge machine learning techniques tailored for COVID_19 monitoring, ensuring precise and timely predictions.**
4.	Social Impact / Customer Satisfaction	**The solution contributes to faster and more accurate prediction of covid 19 using chest x-rays and not relying on other tests.**
5.	Business Model (Revenue Model)	**Implement a subscription-based model for healthcare providers, offering premium features and personalized analytics. Additionally, collaborate with medical institutions for enterprise solutions.**
6.	Scalability of the Solution	**The solution is designed to be scalable, accommodating a growing number of users and expanding to different healthcare settings. Cloud-based infrastructure ensures flexibility and scalability.**

Solution Architecture:

Best Tech Solution:

- The "Covid-19 Detection using Deep Learning with Chest X-rays" project aims to harness the power of deep learning algorithms to facilitate early and accurate diagnosis of Covid-19 by analyzing chest X-ray images

Behavior and Aspects:

- The software behavior is dynamic, adapting to changing data patterns. Aspects include data preprocessing, using best image preprocessing algorithms like RESNET-50, and Transfer learning..

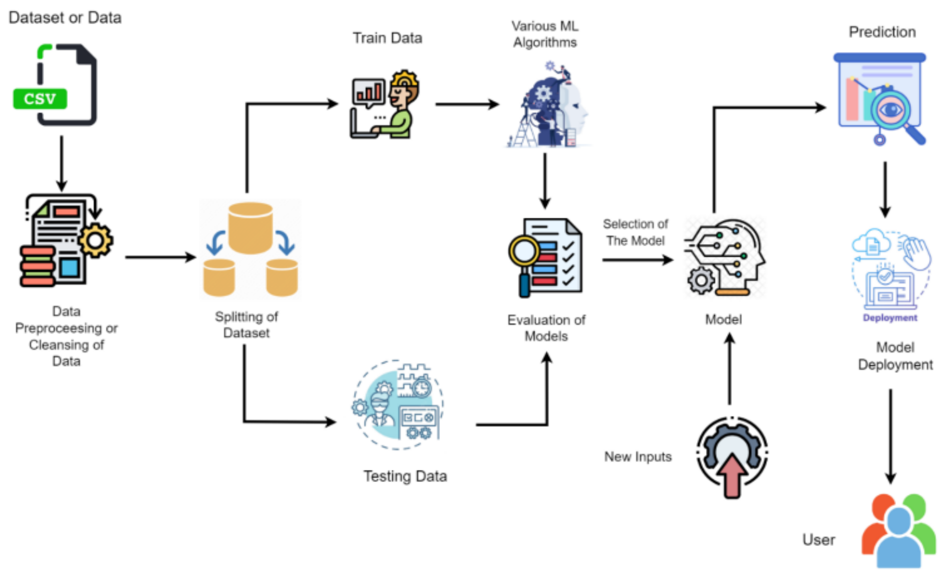
Development Phases:

- Our application undergoes iterative development phases:
    - Data Preprocessing: Cleaning and preparing data for analysis.
    - Model Training: Using RESNET-50.
    - Evaluation: Assessing model accuracy and fine-tuning.
    - Integration: Seamlessly integrating into existing healthcare systems.

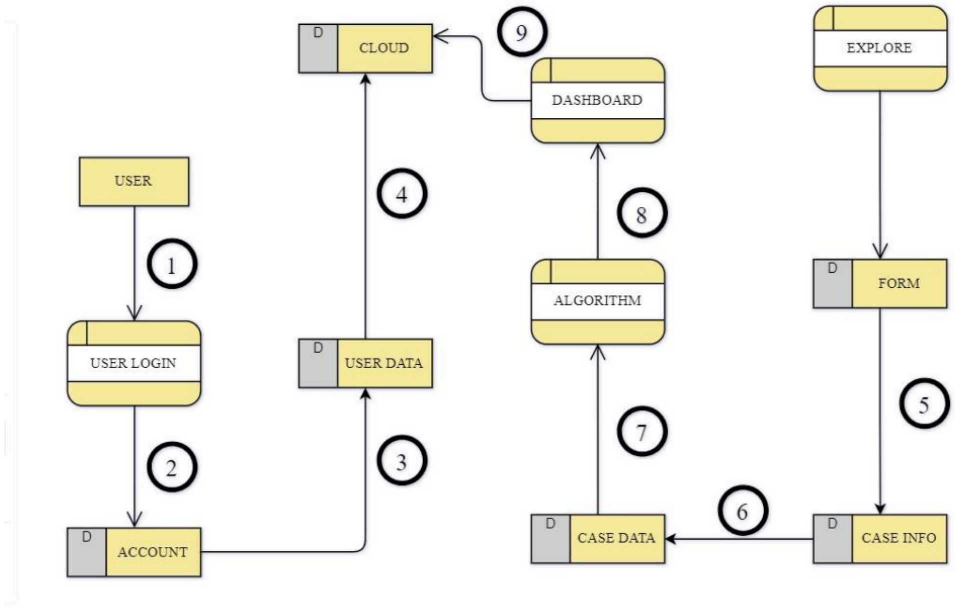
Solution Requirements:

- Key requirements include:
    - Data Quality: High-quality fetal health data.
    - Computational Resources: Adequate resources for model training.
    - Integration: Compatibility with existing healthcare infrastructure.

Solution Architecture Diagram:



DATAFLOW AND USERSTORIES:



1. Users successfully complete the registration process.
2. Within the Web App's Explore section, users are prompted to input specific characteristics.
3. The provided Case Info undergoes a seamless transformation into formatted Case Data.
4. The transformed Case Data serves as input for the FetalAI algorithm.
5. The algorithm processes the data, predicts the score, and presents the results on a dynamic dashboard.

User Stories

Use the below template to list all the user stories for the product.

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer	Registration	USN-1	As web user, I can register for the application by entering	Successfully entering email, password, and	High	Sprint-1

			my email, password, and confirming my password.	confirming password leads to account creation.		
		USN-2	As user, I can register for the application through smartbridge internz platform.	Successful registration and access to dashboard using smartbridge internz credentials.	Low	Sprint -2
		USN-4	As a user, I can register for the application through Gmail	Successful registration and access to the FetalAI dashboard through Gmail.	Medium	Sprint -1
	Login	USN-5	As a mobile user, I can log into the FetalAI application by entering my email and password.	Successfully entering a valid email and password leads to login.	High	Sprint -1
	Best Algorithm Finding	USN-7	Trying out all the available algorithms in order to find which one gives the best accuracy rate	Ability to filter and sort predictions based on parameters such as gestational age, health risk levels, and other relevant factors.	High	Sprint-1

	Finding correlations	USN-8	We have a huge number of 21 parameters which can be hectic to handle, hence we shall find correlated columns and eliminate them.	● Access a user-friendly dashboard offering comprehensive insights into health predictions for selected pregnancies. Explore predicted health parameters, receive recommendations, and view relevant data visualizations. Utilize convenient filters and sorting options based on	High	Sprint-1

				parameters like gestationalage, health risk levels, and other relevant factors.		
	Logo requirement	USN-10	Find or design an apt logo for the WebUI	The logo should reflect the essence of our application, conveying a sense of health,and advanced technology. The colors used in the logo should be harmonious with the color scheme of the WebUI, promoting visual consistency.	Medium	Sprint-1
	Defining Description	USN-11	A detailed information about the application, its uses, and its application should be available for the users in order to understand better about the model.	The information should be easily accessible within the application, preferably	Medium	Sprint-2

				through a dedicated section or help center.		
Customer Care Executive	Contact us page	USN-12	In order to allow the users to post further queries, a contact us part of the page must be made available with the details of our team in it and how to contact us.	Include a mechanism for users to provide feedback on the "Contact Us" process, ensuring continuous improvement and addressing any issues promptly.	Medium	Sprint-3
	Back Navigator	USN-14	A button must be provided for the users to return to the predictor_inputs page to start predicting from the model again	Use an intuitive icon or label for the button to clearly convey its purpose, ensuring users understand that clicking it will take them back to the predictor_inputs page.	Low	Sprint-4

REQUIREMENTS

Technical Architecture:

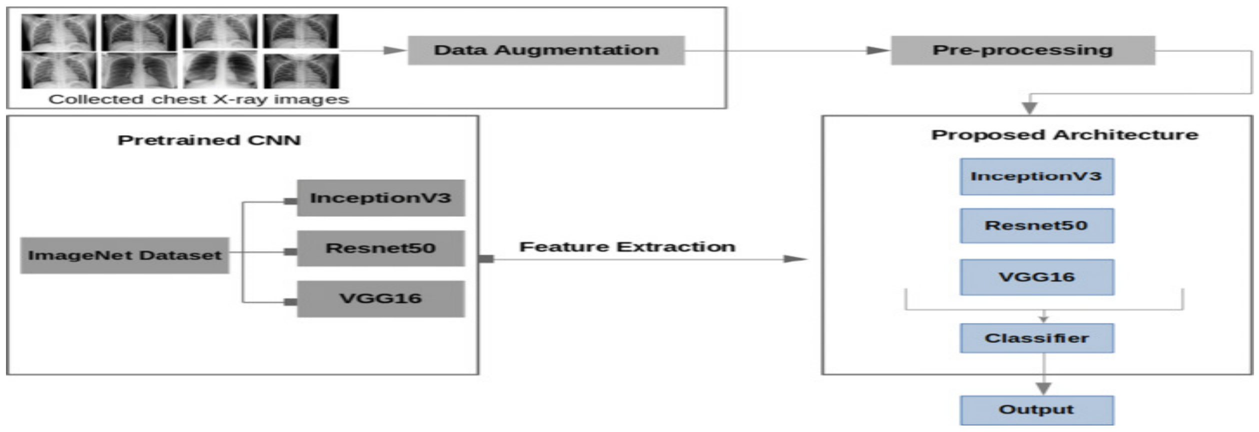


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Collection	Sources of chest X-ray images (hospitals, online databases) - Web scraping tools (if collecting from online sources)	-.
2.	Data PreProcessing	Python libraries (NumPy, OpenCV) for image preprocessing -	Python

		Data cleaning and formatting	
3.	Convolutional Neural Network	Deep learning framework (e.g., TensorFlow, PyTorch) - Model architecture (VGG16, ResNet, custom architecture)	Python Libraries and frameworks
4.	Training and Validation	Training data for the model - Validation data for model evaluation - Data augmentation techniques	Python Libraries and frameworks
5.	Hyperparameter Tuning	Hyperparameter optimization libraries (e.g., Keras Tuner)	Python Libraries and frameworks
6.	Model export	Tensorflow saved model format	Python Libraries and frameworks.
7.	Web Framework Flask	Flask for creating web application ,Python for backend development	Python FLask
8.	API design	Flask REST-ful API design	Flask RESTful
9.	User Interface	Frontend development of user interface	HTML,CSS,JS
10.	Version Control	To track changes and control	Version Control

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Purpose	Automated detection of COVID-19 in chest X-ray images.	Deep learning frameworks (e.g., TensorFlow, PyTorch)
2.	Target	Users Healthcare professionals, researchers, and the general public.	HTML, CSS, JavaScript,
3.	Efficiency	The system should provide results quickly and efficiently.	.Backend optimizations, cloud-based computing (AWS, Google Cloud)

S.No	Characteristics	Description	Technology
4.	Accessibility	Should be accessible online via web or mobile interfaces.	Web and mobile application development
5.	User-Friendly	The interface should be easy to	HTML CSS

		use and understand.	

PROJECT PLANNING $ SCHEDULING

Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Establish the project's foundation	USN-1	-Collect diverse chest X-ray datasets: Research and collect COVID-19 and normal chest X-ray datasets. Preprocess and clean the data: Organize and store data, resize, normalize, handle missing data. - Design a CNN architecture: Choose an appropriate architecture, split data into train, validation, and test sets, implement data augmentation..	10	High	
Sprint-2	Develop and train the initial CNN model	USN-2	- Train the CNN model: Configure model architecture, train on training set, implement early stopping and model checkpointing. - Evaluate model performance: Use validation set, implement evaluation metrics, identify areas for improvement.	10	High	
Sprint-3	Optimize the model's performance and prepare for deployment	USN-3	- Fine-tune model hyperparameters: Perform hyperparameter tuning based on validation results, document the best hyperparameters. - Prepare for deployment: Export the trained model in a suitable format (e.g., TensorFlow SavedModel,	10	High	

			ONNX), create a simple user interface for testing.			
Sprint-4	Finalize the project, conduct testing, and document	USN-4	- Conduct testing and debugging: Test the model with COVID-19 and normal chest X-ray images, address and fix issues, ensure a functional user interface. - Document the project: Create project documentation, prepare a presentation or report for stakeholders.	10	High	
Sprint-1	Deploy the model On flask.	USN-5	- Set up Flask application: Establish a Flask application for deploying the model. - Model Integration: Integrate the trained model into the Flask application. - Create RESTful API: Develop a RESTful API endpoint for model inference. - Test Flask Deployment: Conduct testing and debugging for the Flask deployment. - Document Flask Deployment: Create documentation for the Flask deployment, including setup instructions and API documentation.	10	High	

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	4 Days	2 Nov 2023	5 Nov 2023	10	1 November 2023
Sprint-2	20	4 Days	6 Nov 2023	9 Nov 2023	10	1 November 2023
Sprint-3	20	4	10 Nov 2023	13 Nov 2023	10	1 November 2023

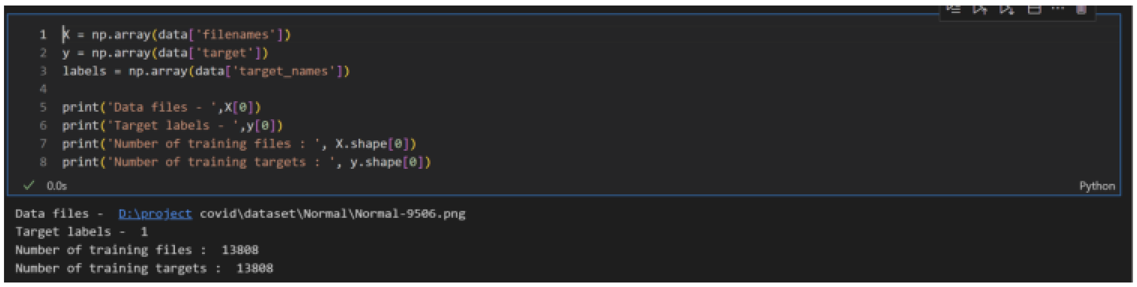
		Days				
Sprint-4	20	4 Days	14 Nov 2023	17 Nov 2023	10	1 November 2023

CODING & SOLUTIONING

1)Libraries-
from imutils
import paths
import matplotlib.pyplot as plt
import numpy as np
import argparse
import cv2
import os
import time
import keras

2) Access to datasets-
We import the data set from Kaggle (https://www.kaggle.com/code/rollanmaratov/covid19-detection-using-tensorflow-from-chest-xray/data). From here we import the chest X rays of both covid and normal patients and import to the model.

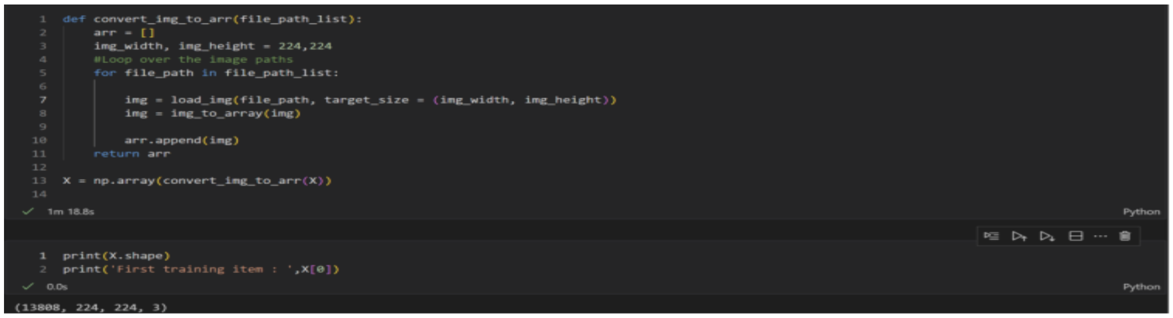
3)Data Preprocessing
Convert the data and labels to Numpy arrays



```
X = np.array(data['filenames'])
y = np.array(data['target'])
labels = np.array(data['target_names'])

print('Data files - ',X[0])
print('Target labels - ',y[0])
print('Number of training files : ', X.shape[0])
print('Number of training targets : ', y.shape[0])
```

```
Data files -  D:\project covid\dataset\Normal\Normal-9506.png
Target labels -  1
Number of training files :  13808
Number of training targets :  13808
```

Convert Images into arrays using keras-



```
def convert_img_to_arr(file_path_list):
    arr = []
    img_width, img_height = 224,224
    #Loop over the image paths
    for file_path in file_path_list:

        img = load_img(file_path, target_size = (img_width, img_height))
        img = img_to_array(img)

        arr.append(img)
    return arr

X = np.array(convert_img_to_arr(X))

```

```
1m 18.8s
```

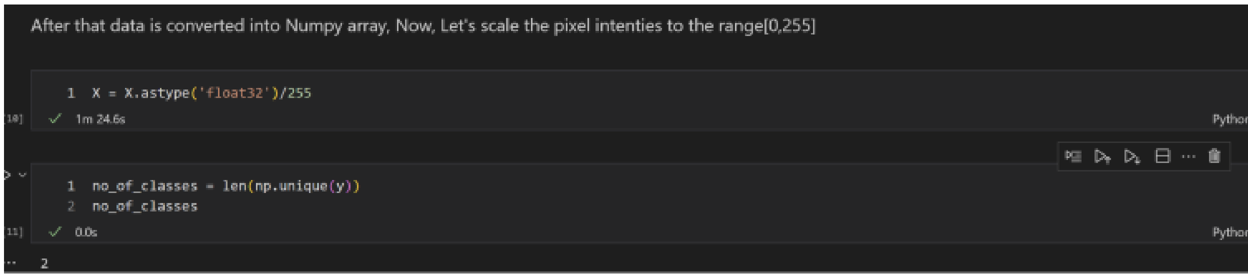
```
print(X.shape)
print('First training item : ',X[0])
```

```
0.0s
(13808, 224, 224, 3)
```

Note that the shape of training data is (13808, 224, 224, 3)

- 13808 is the number of training items or files,
- (224,224) is the target size or image size provided while loading image
- 3 refers to the depth for colored images (RGB channels).

Rescaling it-



After that data is converted into Numpy array, Now, Let's scale the pixel intenties to the range[0,255]

```
X = X.astype('float32')/255
```

```
[10]  1m 24.6s
```

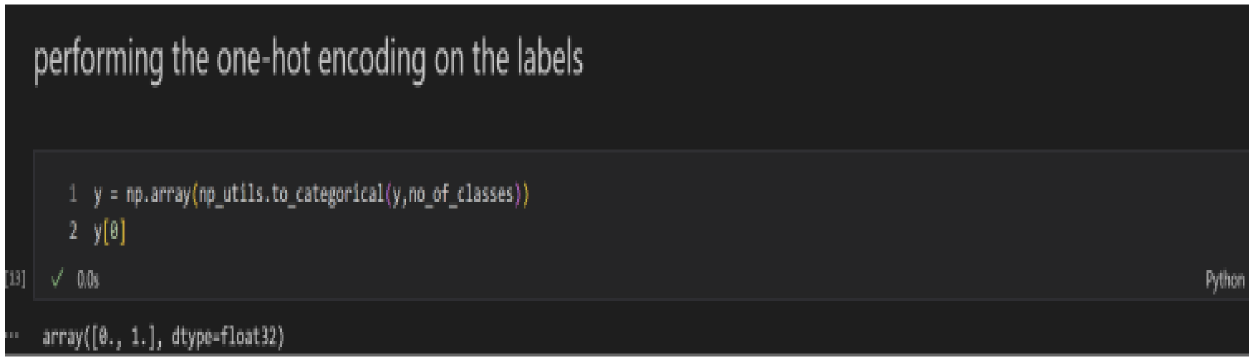
```
no_of_classes = len(np.unique(y))
no_of_classes
```

```
[11]  0.0s
2
```

The "rescale value" is a number that we will use to multiply the data before any other processing. The original images have RGB values between 0-255, but these values are too high for our models to process. Therefore, we aim to have values between 0 and 1 by scaling with a factor of 1/255. This ensures that all values in X will be within the 0-1 range.

So We created our Problem into a Binary Classification Problem,wheather it is infected or not-
0 or 1

Converts a class vector (integers) to binary class matrix by performing the one-hot encoding on the labels


performing the one-hot encoding on the labels

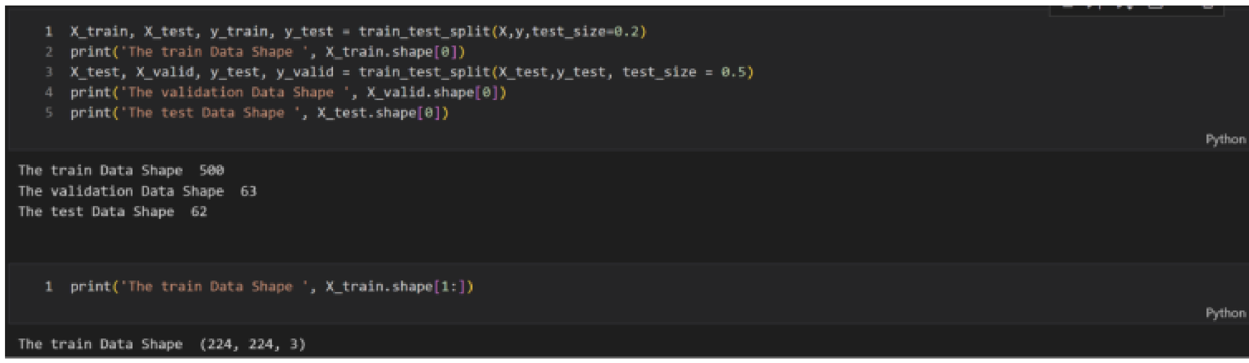
```
y = np.array(np_utils.to_categorical(y,no_of_classes))
y[0]
```

[13] ✓ 0.0s

Python

```
array([0., 1.], dtype=float32)
```

<u>Splitting the data into train, test and validation set</u>
We also split the validation data set cause at the end of model building,we need a set of dataset just for accuracy and validation.



```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2)
print('The train Data Shape ', X_train.shape[0])
X_test, X_valid, y_test, y_valid = train_test_split(X_test,y_test, test_size = 0.5)
print('The validation Data Shape ', X_valid.shape[0])
print('The test Data Shape ', X_test.shape[0])
```

Python

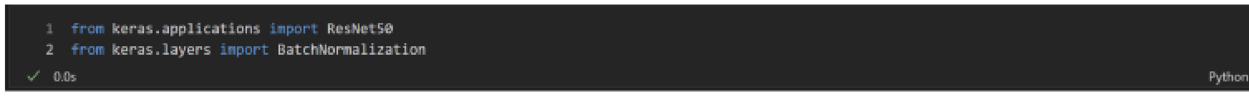
```
The train Data Shape  500
The validation Data Shape  63
The test Data Shape  62
```

```
print('The train Data Shape ', X_train.shape[1:])
```

Python

```
The train Data Shape  (224, 224, 3)
```

4)CNN implementation with Resnet50



```
from keras.applications import ResNet50
from keras.layers import BatchNormalization
```

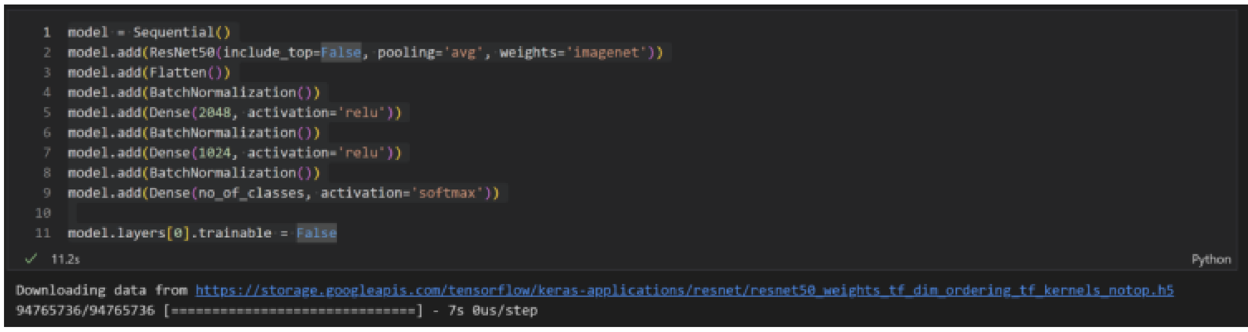
✓ 0.0s

Python

-Now we create a neural network model using the Sequential API in Keras. It starts with a pre-trained ResNet50 model as the first layer, followed by flatten, batch normalization, and fully connected layers.

-The last layer has the number of units equal to the number of classes in the problem, with a softmax activation function.

-The first layer's weights are frozen to preserve the pre-trained weights. This architecture allows for efficient feature extraction and classification.



```
model = Sequential()
model.add(ResNet50(include_top=False, pooling='avg', weights='imagenet'))
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(2048, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(1024, activation='relu'))
model.add(BatchNormalization())
model.add(Dense(no_of_classes, activation='softmax'))

model.layers[0].trainable = False
```

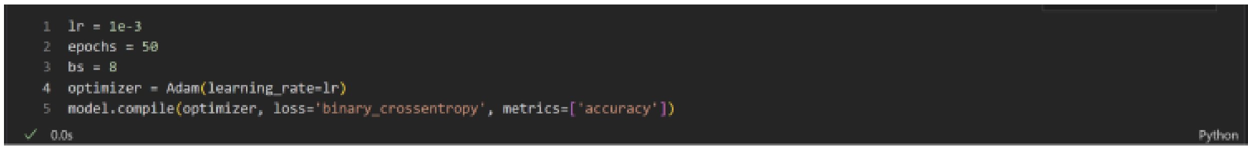
```
✓ 11.2s                                                                Python
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 7s 0us/step
```

5)Modelling

So For compiling the model we will use Adam Optimizer

for using the Adam optimizer with a specified learning rate to navigate through the parameter space during training. The learning rate is considered a crucial hyperparameter, and adjusting it can impact the convergence and stability of the optimization process, especially when dealing with potential challenges like getting stuck in local minima.

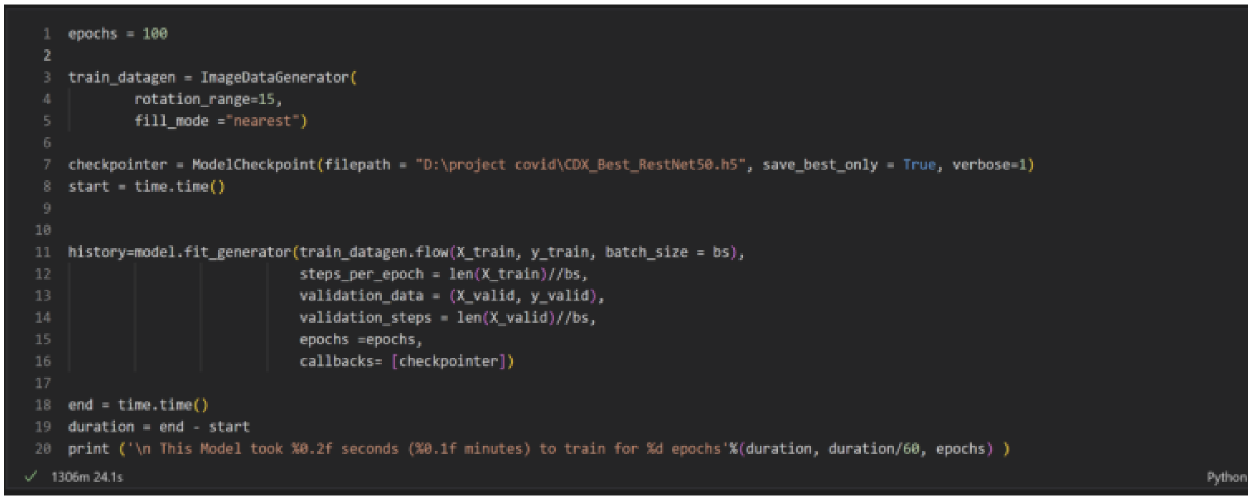
So we employ the Adam optimizer to guide the model toward the global minimum during training. If the training process encounters local minima, the Adam optimizer is expected to facilitate escaping from them and progressing towards the global minimum. Additionally, I'll define the learning rate for the optimizer, specifically setting it at 1e-3 (0.001).



```
lr = 1e-3
epochs = 50
bs = 8
optimizer = Adam(learning_rate=lr)
model.compile(optimizer, loss='binary_crossentropy', metrics=['accuracy'])
```

```
✓ 0.0s                                                                 Python
```

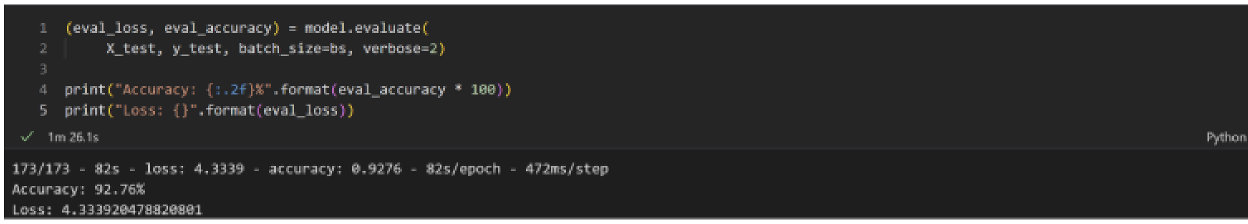
Using model.fit_generator as I am using ImageDataGenerator to pass data to the model.



```
epochs = 100

train_datagen = ImageDataGenerator(
        rotation_range=15,
        fill_mode ="nearest")

checkpointer = ModelCheckpoint(filepath = "D:\project covid\CDX_Best_RestNet50.h5", save_best_only = True, verbose=1)
start = time.time()


history=model.fit_generator(train_datagen.flow(X_train, y_train, batch_size = bs),
                            steps_per_epoch = len(X_train)//bs,
                            validation_data = (X_valid, y_valid),
                            validation_steps = len(X_valid)//bs,
                            epochs =epochs,
                            callbacks= [checkpointer])

end = time.time()
duration = end - start
print ('\n This Model took %0.2f seconds (%0.1f minutes) to train for %d epochs'%(duration, duration/60, epochs) )
```

✓ 1306m 24.1s Python

6) Evaluation

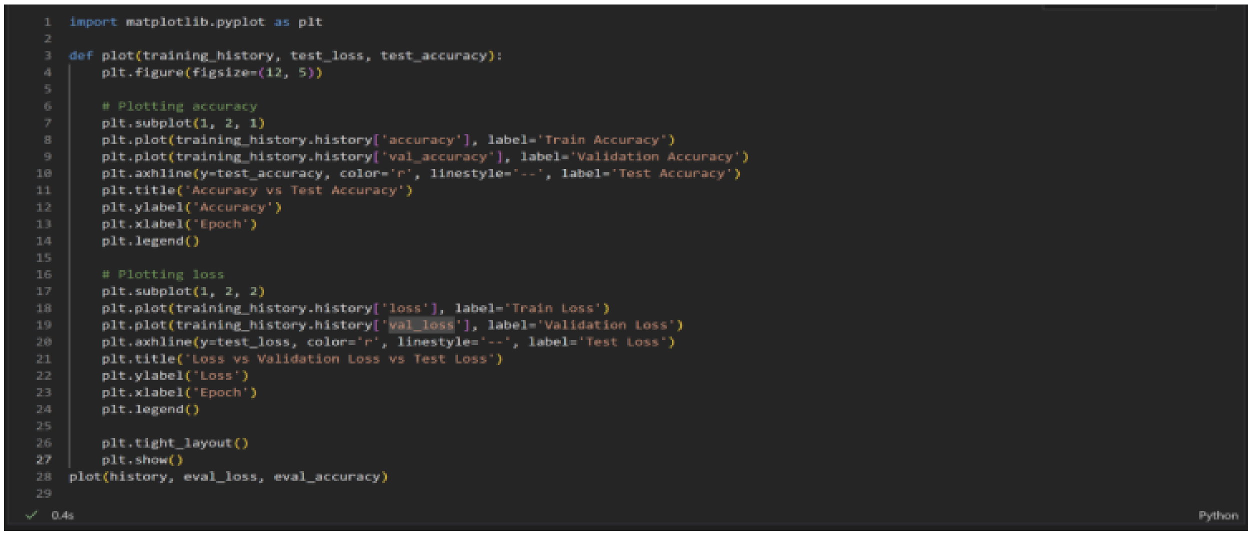


```
(eval_loss, eval_accuracy) = model.evaluate(
     X_test, y_test, batch_size=bs, verbose=2)

print("Accuracy: {:.2f}%".format(eval_accuracy * 100))
print("Loss: {}".format(eval_loss))
```

✓ 1m 26.1s Python

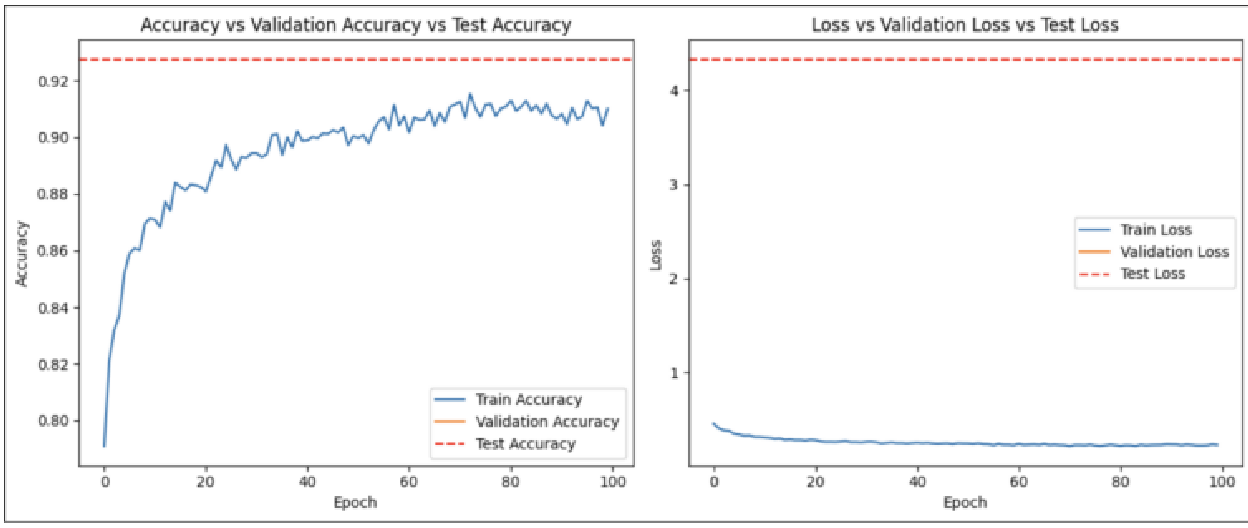
```
173/173 - 82s - loss: 4.3339 - accuracy: 0.9276 - 82s/epoch - 472ms/step
Accuracy: 92.76%
Loss: 4.333920478820801
```

7)Accuracy and loss graph

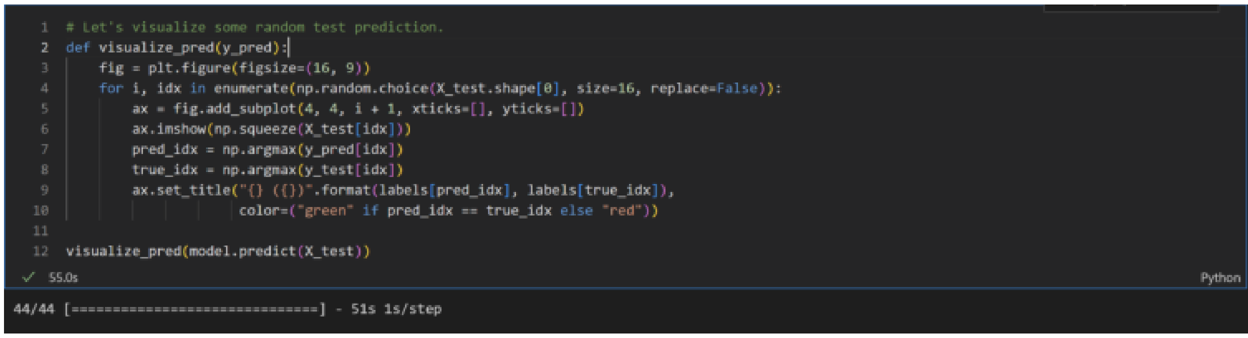


```
import matplotlib.pyplot as plt

def plot(training_history, test_loss, test_accuracy):
    plt.figure(figsize=(12, 5))

    # Plotting accuracy
    plt.subplot(1, 2, 1)
    plt.plot(training_history.history['accuracy'], label='Train Accuracy')
    plt.plot(training_history.history['val_accuracy'], label='Validation Accuracy')
    plt.axhline(y=test_accuracy, color='r', linestyle='--', label='Test Accuracy')
    plt.title('Accuracy vs Test Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend()

    # Plotting loss
    plt.subplot(1, 2, 2)
    plt.plot(training_history.history['loss'], label='Train Loss')
    plt.plot(training_history.history['val_loss'], label='Validation Loss')
    plt.axhline(y=test_loss, color='r', linestyle='--', label='Test Loss')
    plt.title('Loss vs Validation Loss vs Test Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend()

    plt.tight_layout()
    plt.show()
plot(history, eval_loss, eval_accuracy)

```

✓ 0.4s Python



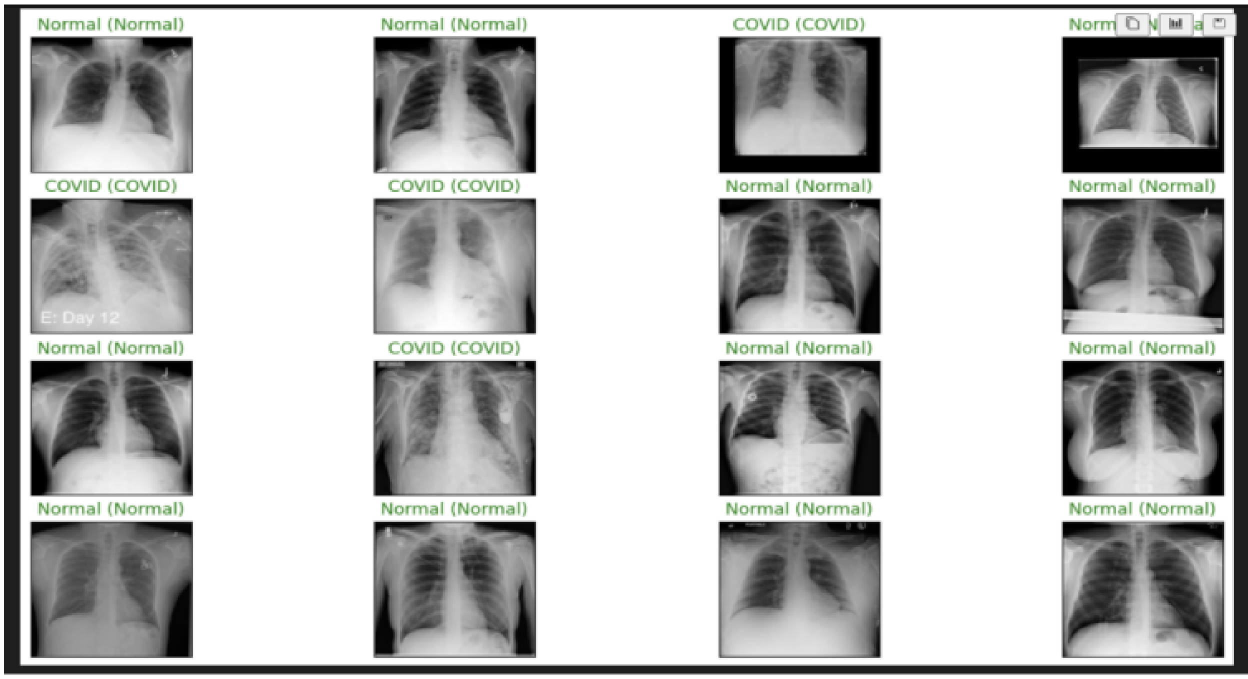
8)Prediction



```
# Let's visualize some random test prediction.
def visualize_pred(y_pred):
    fig = plt.figure(figsize=(16, 9))
    for i, idx in enumerate(np.random.choice(X_test.shape[0], size=16, replace=False)):
        ax = fig.add_subplot(4, 4, i + 1, xticks=[], yticks=[])
        ax.imshow(np.squeeze(X_test[idx]))
        pred_idx = np.argmax(y_pred[idx])
        true_idx = np.argmax(y_test[idx])
        ax.set_title("{} ({})".format(labels[pred_idx], labels[true_idx]),
                     color=("green" if pred_idx == true_idx else "red"))

visualize_pred(model.predict(X_test))
```

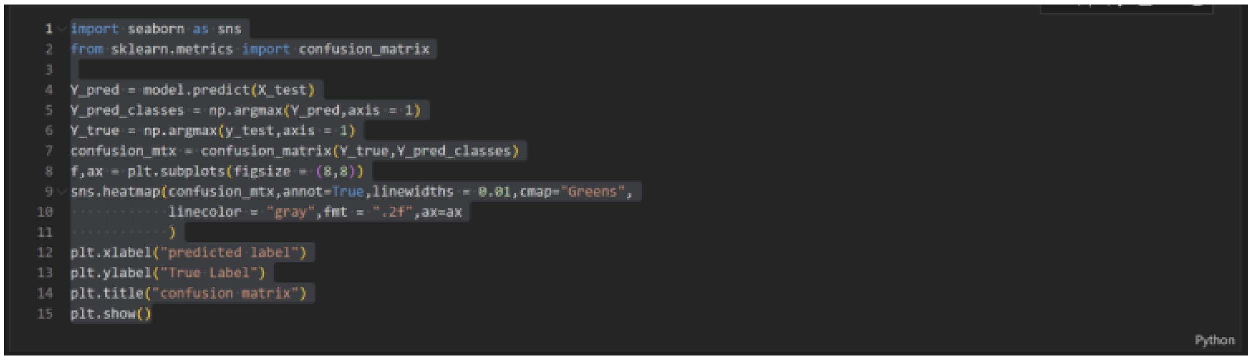
```
✓ 55.0s                                                          Python
44/44 [==============================] - 51s 1s/step
```

a visualization function called visualize_pred has been implemented to showcase random test predictions. The function generates a 4x4 grid of subplots, each displaying an image from the test set. The titles of the subplots indicate the model's predictions and true labels. Correct predictions are labeled in green, while incorrect ones are labeled in red. This insightful visualization provides a quick and visual assessment of the model's performance on the test data. The function is applied to the predictions obtained from the model using model.predict(X_test)



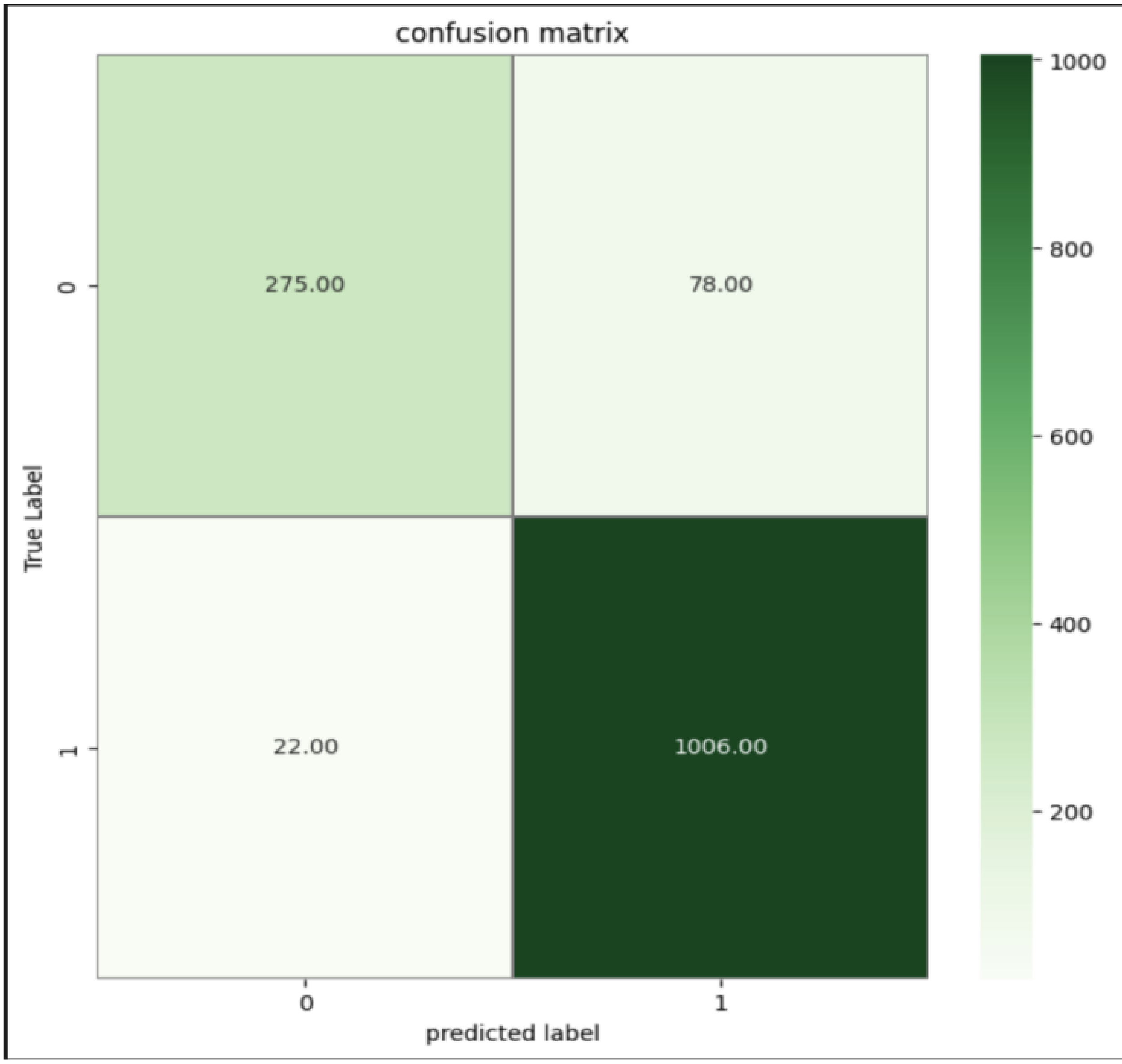
PERFORMANCE TESTING

Confusion Matrix-

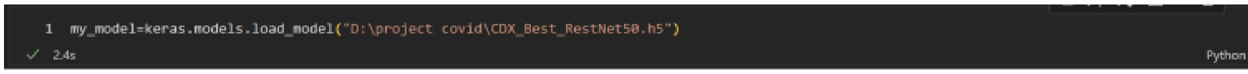


```python
import seaborn as sns
from sklearn.metrics import confusion_matrix

Y_pred = model.predict(X_test)
Y_pred_classes = np.argmax(Y_pred,axis = 1)
Y_true = np.argmax(y_test,axis = 1)
confusion_mtx = confusion_matrix(Y_true,Y_pred_classes)
f,ax = plt.subplots(figsize = (8,8))
sns.heatmap(confusion_mtx,annot=True,linewidths = 0.01,cmap="Greens",
            linecolor = "gray",fmt = ".2f",ax=ax
            )
plt.xlabel("predicted label")
plt.ylabel("True Label")
plt.title("confusion matrix")
plt.show()
```

Python



Finally uploading the model



```
my_model=keras.models.load_model("D:\project covid\CDX_Best_RestNet50.h5")
✓ 2.4s                                                                Python
```

ADVANTAGES AND DISADVANTAGES

10.1 Advantages

10.1.1 Early Detection
One of the primary advantages of the project is its potential to enable early detection of Covid-19 cases. By leveraging deep learning on chest X-ray images, the system can identify patterns indicative of infection at an early stage, allowing for prompt medical intervention.

10.1.2 Automation and Efficiency
The automated nature of the deep learning model streamlines the diagnostic process, reducing the burden on healthcare professionals. This can lead to quicker turnaround times in diagnosis, especially during periods of high demand, such as a pandemic.

10.1.3 Resource Optimization
Efficient identification of Covid-19 cases through automated analysis of chest X-rays can contribute to resource optimization in healthcare settings. By assisting in prioritizing and allocating resources, the project may enhance the overall management of the pandemic.

10.1.4 Scalability
The deep learning model can be scaled for widespread use, potentially benefiting healthcare facilities with varying capacities. The scalability of the solution makes it adaptable to different healthcare settings and resource availability.

10.1.5 Continuous Learning and Improvement
The project can be designed to incorporate continuous learning mechanisms, allowing the model to improve over time with additional data. This adaptability enhances the model's accuracy and reliability as more information becomes available.

10.2 Disadvantages
10.2.1 Dataset Limitations
One significant challenge is the availability of diverse and well-annotated datasets. Limited datasets can impact the model's ability to generalize across different populations and conditions, potentially leading to biased or less accurate predictions.

10.2.2 Interpretability
The inherent complexity of deep learning models, especially convolutional neural networks, may result in a lack of interpretability. Healthcare professionals may find it challenging to understand the rationale behind the model's predictions, raising concerns about trust and acceptance.

10.2.3 Ethical Considerations

The deployment of AI in healthcare raises ethical concerns related to privacy, informed consent, and the responsible use of technology. Ensuring that the project adheres to ethical guidelines and regulations is crucial to its success and acceptance.

10.2.4 False Positives and Negatives

No model is perfect, and there is a risk of false positives and false negatives. Misdiagnosing individuals can have serious consequences, and managing the trade-off between sensitivity and specificity is a critical consideration in the development of the deep learning model.

10.2.5 Resource Requirements

Implementing and maintaining a deep learning solution requires significant computational resources. Healthcare facilities with limited resources may face challenges in adopting and sustaining the technology, impacting its accessibility in certain settings.

FLASK DEPLOYMENT

CODE:

```
from flask import Flask, render_template, request
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
from PIL import Image
import io

app = Flask(__name__)

model_path = "CDX_Best_RestNet50.h5"
model = load_model(model_path)


def preprocess_image(img):
        img = img.convert("RGB")  # Convert to RGB in case the image is not in RGB format
        img = img.resize((224, 224))  # Resize image to match model's expected sizing
        img_array = image.img_to_array(img)
        img_array = np.expand_dims(img_array, axis=0)
        return img_array


def predict(image):
        try:
        processed_img = preprocess_image(image)
        predictions = model.predict(processed_img)
        covid_probability = predictions[0][0]
        if covid_probability > 0.5:  # Adjust threshold as needed
        return "COVID Positive"
        else:
        return "COVID Negative"
        except Exception as e:
        print(str(e))
        return "Error occurred while processing the image"


@app.route("/", methods=["GET", "POST"])
def upload_file():
        result = None
        if request.method == "POST":
        file = request.files["file"]

        if file.filename == "":
        return render_template("index.html", result="No selected file")

        try:
        img = Image.open(io.BytesIO(file.read()))
        img.verify()  # Verify if the file is an image
        result = predict(img)
        except Exception as e:
        print(str(e))
        return render_template(
        "index.html", result="Invalid image file or unreadable image"
        )

        return render_template("index.html", result=result)


if __name__ == "__main__":
        app.run(debug=True, port=8080)
```

Working:

The Flask code defines a web application with a single route ("/") handling both GET and POST requests. For GET requests, it renders an HTML template ("index.html") with no initial result displayed. Upon a POST request triggered by an uploaded file, the app verifies its validity, processes it as an image, and invokes a function to predict COVID-19 positivity. The predicted result ("COVID Positive" or "COVID Negative") is then displayed on the webpage. The app uses

the Flask framework to create an interface for users to upload images for COVID-19 detection using a pre-trained ResNet50 model, offering a streamlined prediction experience through a web browser.

preprocess_image(img): This function takes an image as input, converts it to RGB format (if not already in that format), resizes it to match the model's expected input size (224x224 pixels), and preprocesses it for model inference by converting it into a numpy array.

predict(image): Accepts a preprocessed image as input and utilizes the pre-trained ResNet50 model to make a prediction regarding the probability of COVID-19 infection. It returns either "COVID Positive" if the model predicts a probability greater than 0.5 or "COVID Negative" otherwise.

upload_file() (route handler): This function handles both GET and POST requests to the root URL ("/"). Upon a POST request triggered by an uploaded file, it checks if a file has been selected. If a file is chosen, it attempts to read the uploaded image, verifies its validity, and calls the predict() function to obtain the COVID-19 prediction result based on the uploaded image. Any errors during this process are caught and an appropriate error message is displayed. The result is then rendered on the HTML template.

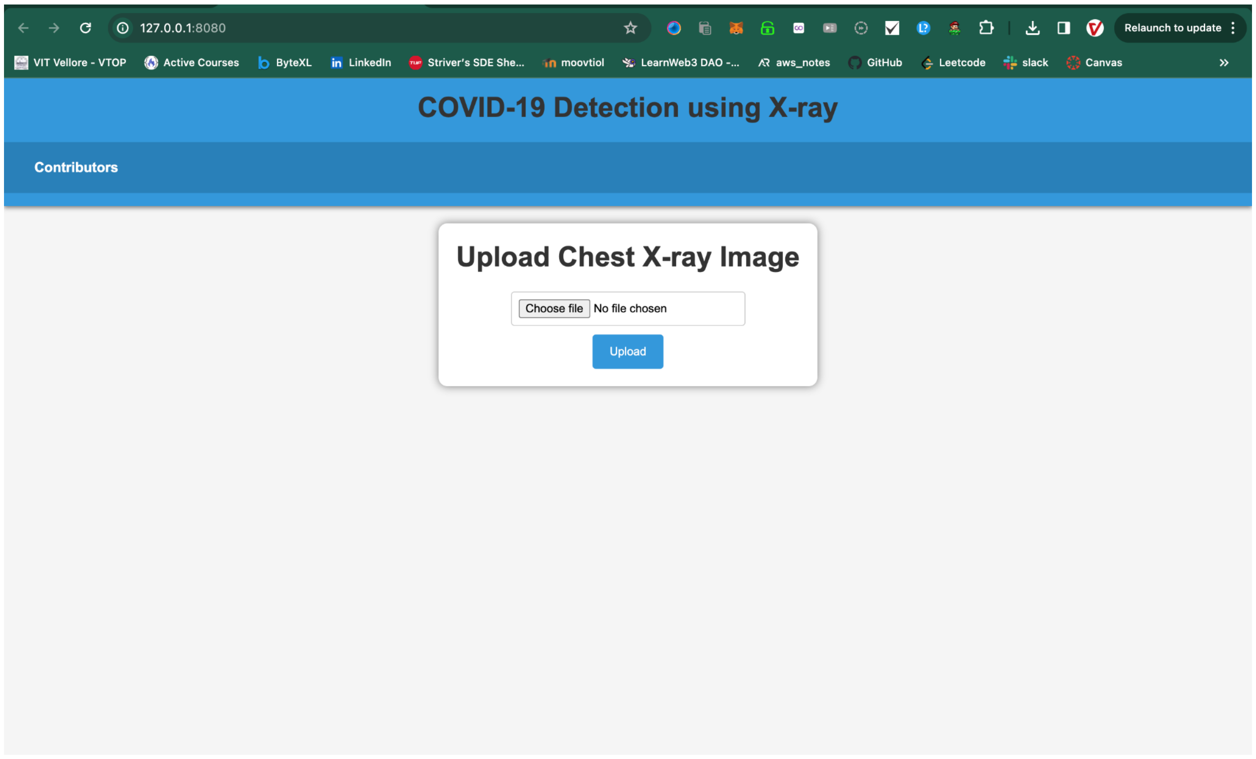
Index.html:

```
<!DOCTYPE html>
<html lang="en">
 <head>
       <meta charset="UTF-8" />
       <title>PneumoScan19</title>
       <style>
       body {
       font-family: Arial, sans-serif;
       margin: 0;
       padding: 0;
       background-color: #f5f5f5; /* Light gray background */
       display: flex;
       flex-direction: column;
       align-items: center;
       height: 100vh;
       }
       header {
       width: 100%;
       background-color: #3498db; /* Light blue */
       color: #ffffff; /* White text */
       padding: 15px 0;
       text-align: center;
       box-shadow: 0px 2px 5px 0px rgba(0, 0, 0, 0.5); /* Subtle shadow */
       }
       nav {
       width: 100%;
       display: flex;
       justify-content: flex-start; /* Align to the left */
       background-color: #2980b9; /* Darker blue */
       padding: 10px 20px; /* Padding adjusted */
       }
       nav a {
       color: #ffffff; /* White text */
       text-decoration: none;
       font-weight: bold;
       transition: background-color 0.3s ease;
       border-radius: 4px;
       padding: 10px 15px; /* Adjusted padding */
       display: inline-block; /* Display links inline */
       }
       nav a:hover {
       background-color: #1abc9c; /* Light green on hover */
       }
       .dropdown {
       position: relative; /* Set dropdown as a relative position */
       display: inline-block; /* Display dropdown inline */
       }
       .contributors {
       display: none;
       position: absolute; /* Position contributors dropdown absolutely */
       left: 0; /* Shift dropdown to the left */
       background-color: #ffffff;
       box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.5);
       border-radius: 4px;
       z-index: 1; /* Ensure visibility over other elements */
       }
       .dropdown:hover .contributors {
       display: block; /* Show contributors on hover */
       }
       .contributors a {
       display: block;
       margin-top: 5px;
       text-decoration: none;
       color: #333333; /* Dark gray text */
       transition: color 0.3s ease;
       font-weight: bold;
       padding: 10px; /* Adjust padding */
       }
       .contributors a:hover {
       color: #555555; /* Slightly lighter text on hover */
       }
```

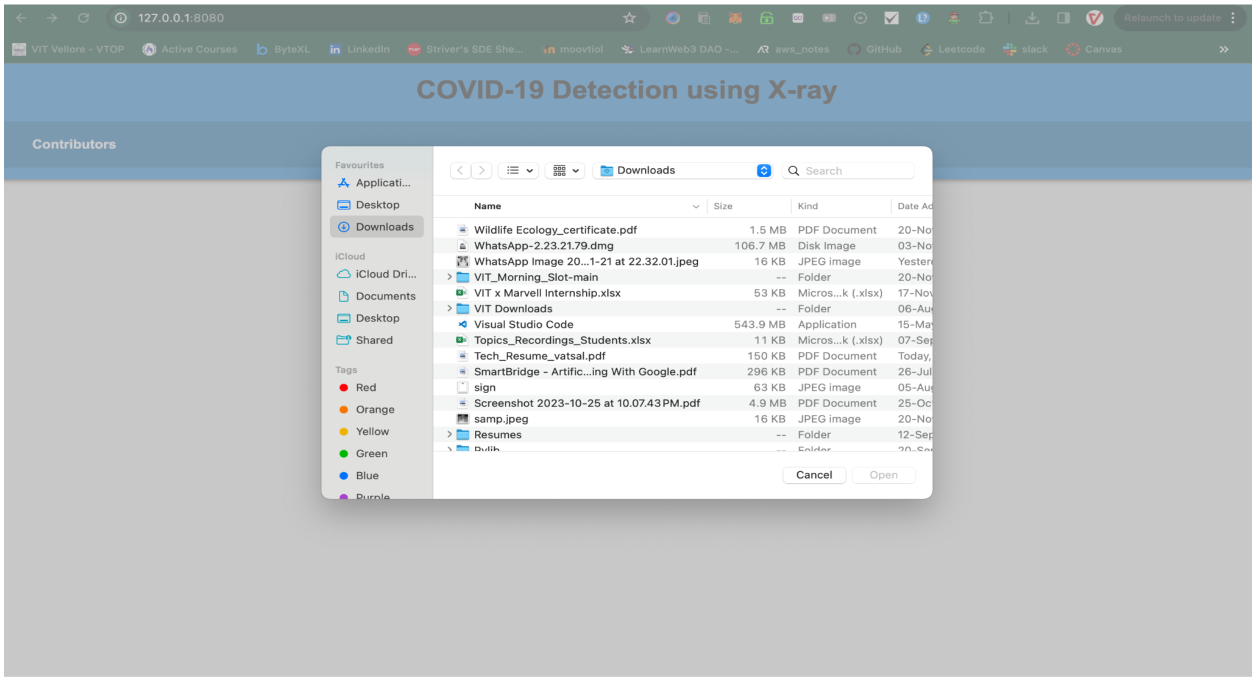
```
        #upload-form {
        background-color: #ffffff; /* White form background */
        padding: 20px;
        border-radius: 10px;
        text-align: center;
        box-shadow: 0px 0px 10px 0px rgba(0, 0, 0, 0.5);
        margin-top: 20px;
        }
        h1 {
        margin-top: 0;
        color: #333333; /* Dark gray text */
        }
        input[type="file"] {
        margin-bottom: 10px;
        padding: 8px;
        border: 1px solid #ccc;
        border-radius: 4px;
        }
        input[type="submit"] {
        background-color: #3498db; /* Blue button */
        color: #ffffff; /* White text */
        padding: 12px 20px;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        transition: background-color 0.3s ease;
        }
        input[type="submit"]:hover {
        background-color: #2980b9; /* Darker blue on hover */
        }
        .result {
        margin-top: 20px;
        font-size: 18px;
        }
        .result.positive {
        color: #e74c3c; /* Red for positive result */
        }
        .result.negative {
        color: #2ecc71; /* Green for negative result */
        }
        </style>
</head>
<body>
        <header>
        <h1>COVID-19 Detection using X-ray</h1>
        <nav>
        <div class="dropdown">
        <a href="#" class="dropbtn">Contributors</a>
        <div class="contributors" id="contributors">
        <a
        href="https://www.linkedin.com/in/vatsal-shah-528412218/"
        target="_blank"
        >Vatsal</a>
        <a href="https://www.linkedin.com/" target="_blank">Rohit</a>
        <a href="https://www.linkedin.com/" target="_blank">Harsh</a>
        <a href="https://www.linkedin.com/" target="_blank">Tej</a>
        <!-- Add more contributors with LinkedIn profile links -->
        </div>
        </div>
        </nav>
        </header>
        <div id="upload-form">
        <h1>Upload Chest X-ray Image</h1>
        <form action="/" method="post" enctype="multipart/form-data">
        <input type="file" name="file" />
        <br />
        <input type="submit" value="Upload" />
        </form>
        {% if result %}
        <div
        class="result {% if result == 'COVID Positive' %}positive{% elif result == 'COVID Negative'
%}negative{% endif %}"
        >
        <h2>Result: {{ result }}</h2>
        </div>
        {% endif %}
        </div>
        <script>
        document
        .querySelectorAll(".dropbtn")[0]  .addEventListener("click", function () {
        var dropdownContent = document.getElementById("contributors");
        if (dropdownContent.style.display === "block") {
        dropdownContent.style.display = "none";
        } else {
        dropdownContent.style.display = "block";
        }
        });
        </script>
 </body>
</html>
```

OUTPUT:

Website UI:



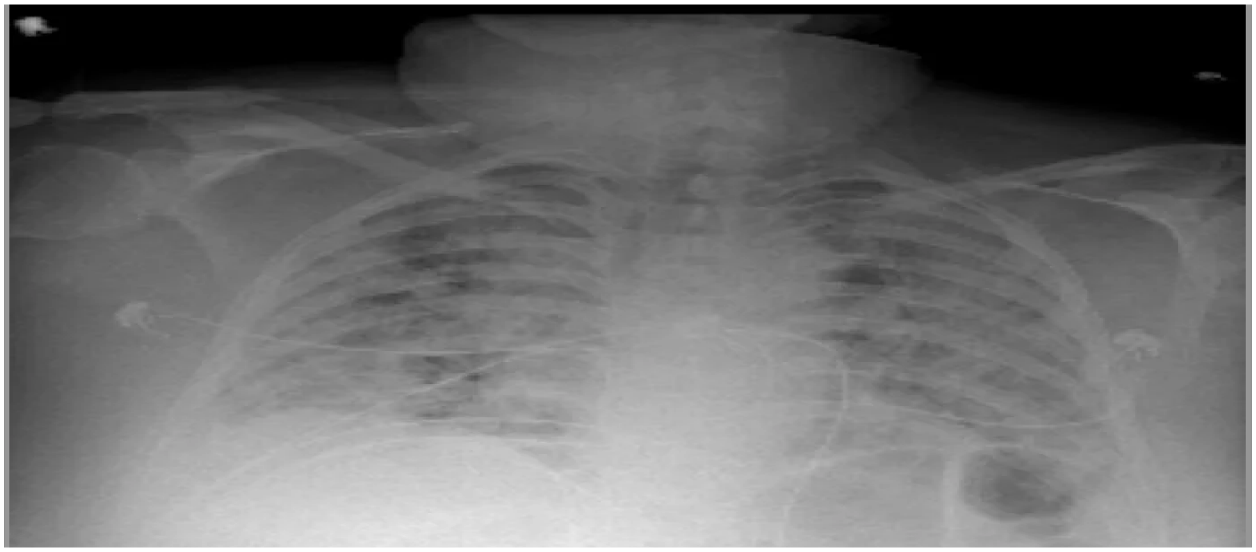
Taking Chest-Xray as input:

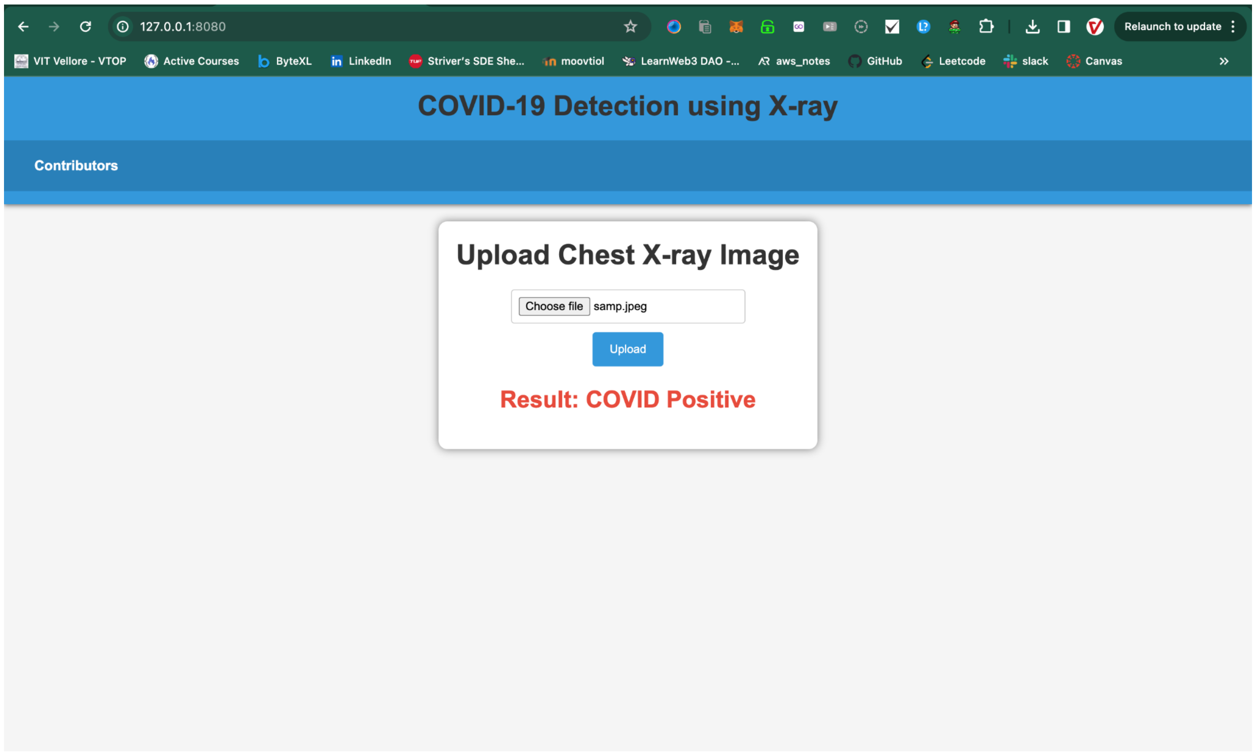


Name	Size	Kind	Date Ad
Wildlife Ecology_certificate.pdf	1.5 MB	PDF Document	20-No
WhatsApp-2.23.21.79.dmg	106.7 MB	Disk Image	03-No
WhatsApp Image 20...1-21 at 22.32.01.jpeg	16 KB	JPEG image	Yester
VIT_Morning_Slot-main	--	Folder	20-No
VIT x Marvell Internship.xlsx	53 KB	Micros...k (.xlsx)	17-Nov
VIT Downloads	--	Folder	06-Au
Visual Studio Code	543.9 MB	Application	15-Ma
Topics_Recordings_Students.xlsx	11 KB	Micros...k (.xlsx)	07-Se
Tech_Resume_vatsal.pdf	150 KB	PDF Document	Today,
SmartBridge - Artific...ing With Google.pdf	296 KB	PDF Document	26-Jul
sign	63 KB	JPEG image	05-Au
Screenshot 2023-10-25 at 10.07.43 PM.pdf	4.9 MB	PDF Document	25-Oc
samp.jpeg	16 KB	JPEG image	20-No
Resumes	--	Folder	12-Sep
Pylib	--	Folder	20-Se

Displaying the Output:

1)Samp.jpeg





COVID-19 Detection using X-ray

Contributors

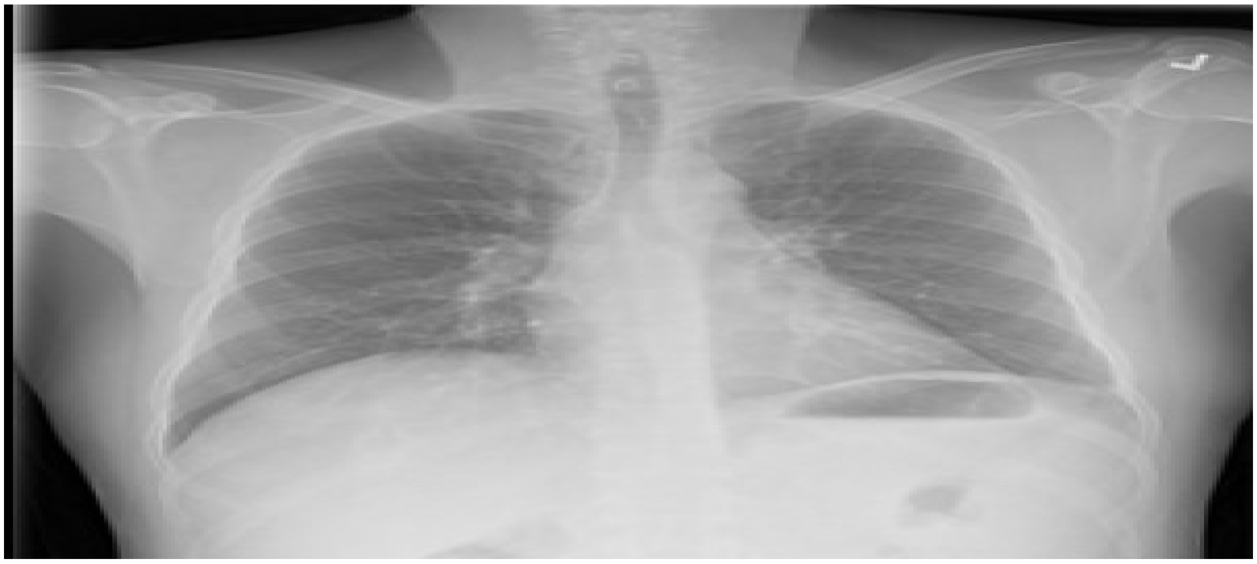
Upload Chest X-ray Image

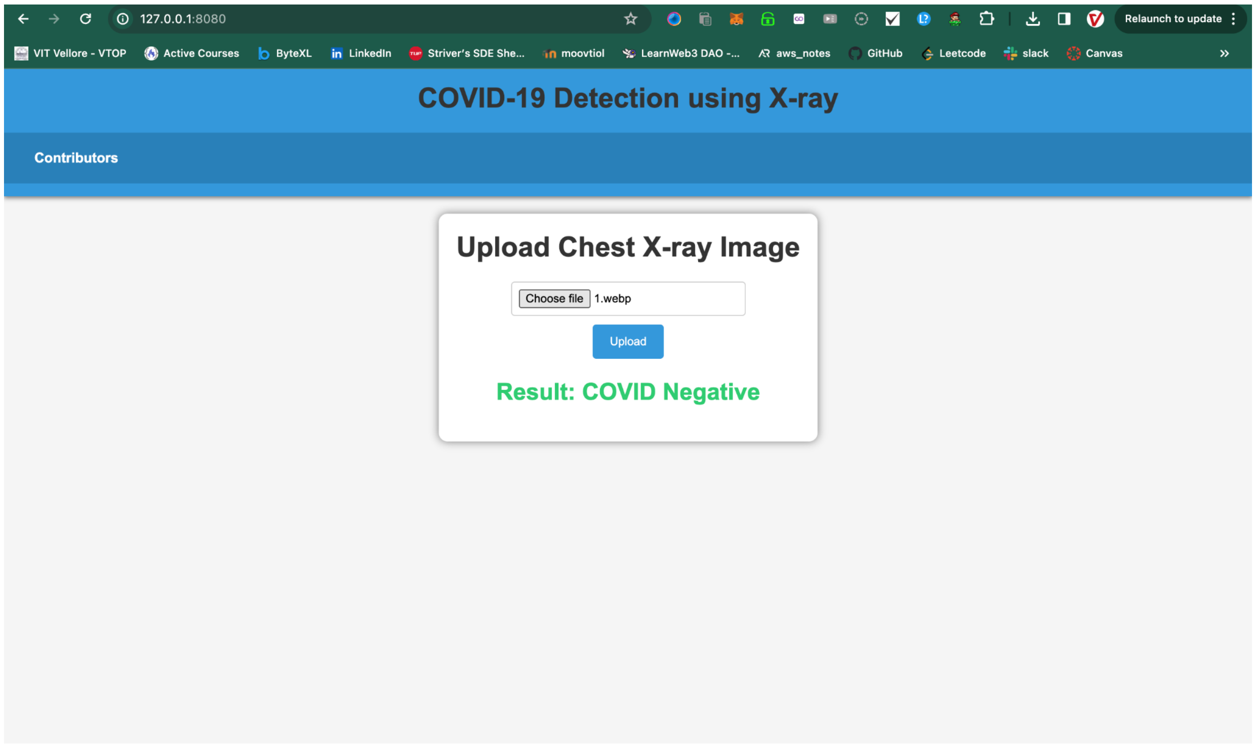
Choose file samp.jpeg

Upload

Result: COVID Positive

2) 1.webp:





COVID-19 Detection using X-ray

Contributors

Upload Chest X-ray Image

Result: COVID Negative

DEMO LINK:

https://youtu.be/L_PjlYccL2E