Detecting COVID-19 From Chest X-Rays Using

Convolution Neural Networks and transfer learning deep learning techniques

• INTRODUCTION TO THE PROJECT

- **PROJECT OVERVIEW:**

DATA COLLECTION AND PREPARATION:

-->Collect a dataset of chest X-ray or CT scan images of COVID-19 patients and non-COVID-19 patients.

-->Label each image with the corresponding patient's diagnosis.

-->Preprocess the images to ensure that they have the same size and format.

-->Augment the data by applying random transformations to the images, such as rotation, flipping, and cropping.

TRANSFER LEARNING WITH CNN:

-->Use a pre-trained CNN model, such as VGG16, as the starting point.

-->Freeze the initial layers of the pre-trained model to prevent them from overfitting to the training data.

-->Fine-tune the top layers of the model using the COVID-19 and non-COVID-19 image dataset to adapt the model to the specific task of COVID-19 detection.

MODEL TRAINING AND EVALUATION:

-->Split the preprocessed dataset into training, validation, and testing sets.

-->Train the CNN model using the training set, monitoring its performance on thevalidation set to prevent overfitting.

-->Evaluate the trained model's performance on the unseen testing set to assess its generalization ability.

MODEL INTERPRETATION AND ANALYSIS:

-->Analyze the model's performance metrics, such as accuracy, precision, recall, and F1-score, to understand its effectiveness in classifying COVID-19 cases.

-->Visualize the model's learned features to gain insights into the patterns and characteristics that it considers important for COVID-19 detection.

MODEL DEPLOYEMENT AND ANALYSIS:

-->Integrate the trained CNN model into a diagnostic system or application for real-world use.

-->Continuously monitor the model's performance in the deployed setting and retrain itperiodically with updated data to maintain its accuracy.

- **PURPOSE OF PROJECT:**

-->The purpose of the project of detecting COVID-19 using CNN and transfer learning isto develop a computer-aided diagnosis (CAD) system that can automatically detect COVID- 19 from chest X-ray or CT scan images. This could be a valuable tool for radiologists and other medical professionals who are responsible for diagnosing COVID-19 patients.

-- >CNNs are a type of artificial intelligence that are particularly well-suited for imagerecognition tasks. They have been shown to be effective in detecting COVID- 19 fromchest X-ray and CT scan images in several studies.

--> Transfer learning is a technique that can be used to improve the performance of CNNs on new tasks. In transfer learning, a CNN that has been trained on a large dataset of images is used as the starting point for a new CNN that is trained on a smaller dataset of images. This can help to improve the performance of the new CNN, as it has already learnedsome of the features that are important for the task

--> The project of detecting COVID-19 using CNN and transfer learning has the potentialto make a significant impact on the fight against the COVID-19 pandemic. By automating the detection of COVID-19, the system could help to reduce the time and effort required to diagnose patients, which could lead to earlier treatment and better outcomes

- **LITERATURE**

Existing problem:

-->Data Collection and Quality: Gathering a large and diverse dataset of chest X-ray or CT scan images of COVID-19 patients is crucial for training a robust model. However, acquiringhigh-quality labeled data can be challenging due to privacy regulations, data silos, and the limited availability of well-annotated images.

-->Data Variability and Noise: Chest X-ray and CT scan images exhibit inherent variabilitydue to factors such as patient positioning, imaging equipment, and image acquisition protocols.Moreover, these images may contain noise or artifacts that can hinder the model's ability to extract meaningful features.

-->Class Imbalance: The prevalence of COVID-19 cases is often lower than non-COVID- 19cases, leading to a class imbalance in the dataset. This imbalance can bias the model towards the majority class, affecting its ability to accurately classify COVID-19 cases.

-->Overfitting and Generalizability: Overfitting occurs when a model learns the trainingdata too well and fails to generalize to unseen data. This can be mitigated through regularization techniques and careful model selection.

-->Interpretability and Explainability: Understanding how a CNN model makes decisionscan be challenging, especially in complex medical applications. Explainable AI techniques can provide insights into the model's decision-making process, enhancing trust and facilitating clinical interpretation.

-->Clinical Integration and Deployment: Integrating a CNN-based COVID-19 detection system into clinical workflows requires careful consideration of factors such as regulatory approvals, compatibility with existing medical imaging systems, and user training.

-->Continuous Monitoring and Adaptation: As COVID-19 variants emerge and the virus characteristics evolve, the model needs to be continuously monitored and updated to maintainits accuracy and effectiveness.

- REFERENCES:

- **PROBLEM STATEMENT DEFINITION:**

Problem Statement: Developing an Accurate COVID-19 Detection System Using CNN and Transfer Learning

-->The COVID-19 pandemic has had a profound impact on the world, causing significant health and economic disruptions. Rapid and accurate diagnosis of COVID-19 cases is crucial for effective patient management, containment strategies, and resource allocation. Traditional diagnostic methods, such as RT-PCR testing, can be time-consuming and expensive, especially inresource-limited settings.

Objective:

-->Develop a computer-aided diagnosis (CAD) system using Convolutional Neural Networks (CNNs) and transfer learning algorithms to accurately detect COVID-19 from chest X- ray or CT scan images.

Challenges:

-->Data Scarcity and Quality: Acquiring a large and diverse dataset of well-labeled chestX-ray or CT scan images of COVID-19 patients is challenging due to privacy regulations, data silos, and the limited availability of annotated images.

-->Data Variability and Noise: Chest X-ray and CT scan images exhibit inherent variabilitydue to patient positioning, imaging equipment, and image acquisition protocols. Moreover, these images may contain noise or artifacts that can hinder the model's ability to extract meaningful features.

-->Class Imbalance: The prevalence of COVID-19 cases is often lower than non-COVID- 19cases, leading to a class imbalance in the dataset. This imbalance can bias the model towards the majority class, affecting its ability to accurately classify COVID-19 cases.

-->Model Interpretability and Explainability: Understanding how a CNN model makes decisions can be challenging, especially in complex medical applications. Explainable AI techniques can provide insights into the model's decision-making process, enhancing trust andfacilitating clinical interpretation.

-->Proposed Approach: Data Collection and Preprocessing: Gather a dataset of chest X-ray or CT scan images of COVID-19 patients and non-COVID-19 patients. Preprocess the images to ensure uniformity andreduce noise.

-->Transfer Learning with CNN: Utilize a pre-trained CNN model, such as ResNet50 orVGG16, as the base architecture. Freeze the initial layers to preserve general image feature extraction capabilities and fine-tune the top layers using the COVID-19 and non- COVID-19 image dataset for COVID-19 detection.

-->Model Training and Evaluation: Split the preprocessed dataset into training, validation, and testing sets. Train the CNN model using the training set, monitoring its performance on the validation set to prevent overfitting. Evaluate the trained model's performance on the unseen testing set to assess its generalization ability.

-->Model Interpretation and Analysis: Analyze the model's performance metrics, such asaccuracy, precision, recall, and F1-score, to understand its effectiveness in classifying COVID-19 cases. Visualize the model's learned features to gain insights into the patterns and characteristics that it considers important for COVID-19 detection.

-->Model Deployment and Application: Integrate the trained CNN model into a diagnostic system or application for real-world use. Continuously monitor the model's performance in the deployed setting and retrain it periodically with updated data to maintainits accuracy.

-->Expected Outcomes:
Development of a highly accurate and generalizable CNN-based COVID-19 detectionsystem. Enhanced efficiency and effectiveness of COVID-19 diagnosis, particularly inresource-limited settings.

Improved patient management and containment strategies through rapid and accuratediagnosis. Contribution to the development of AI-powered diagnostic tools for infectious diseases.

Empathy Map Canvas:

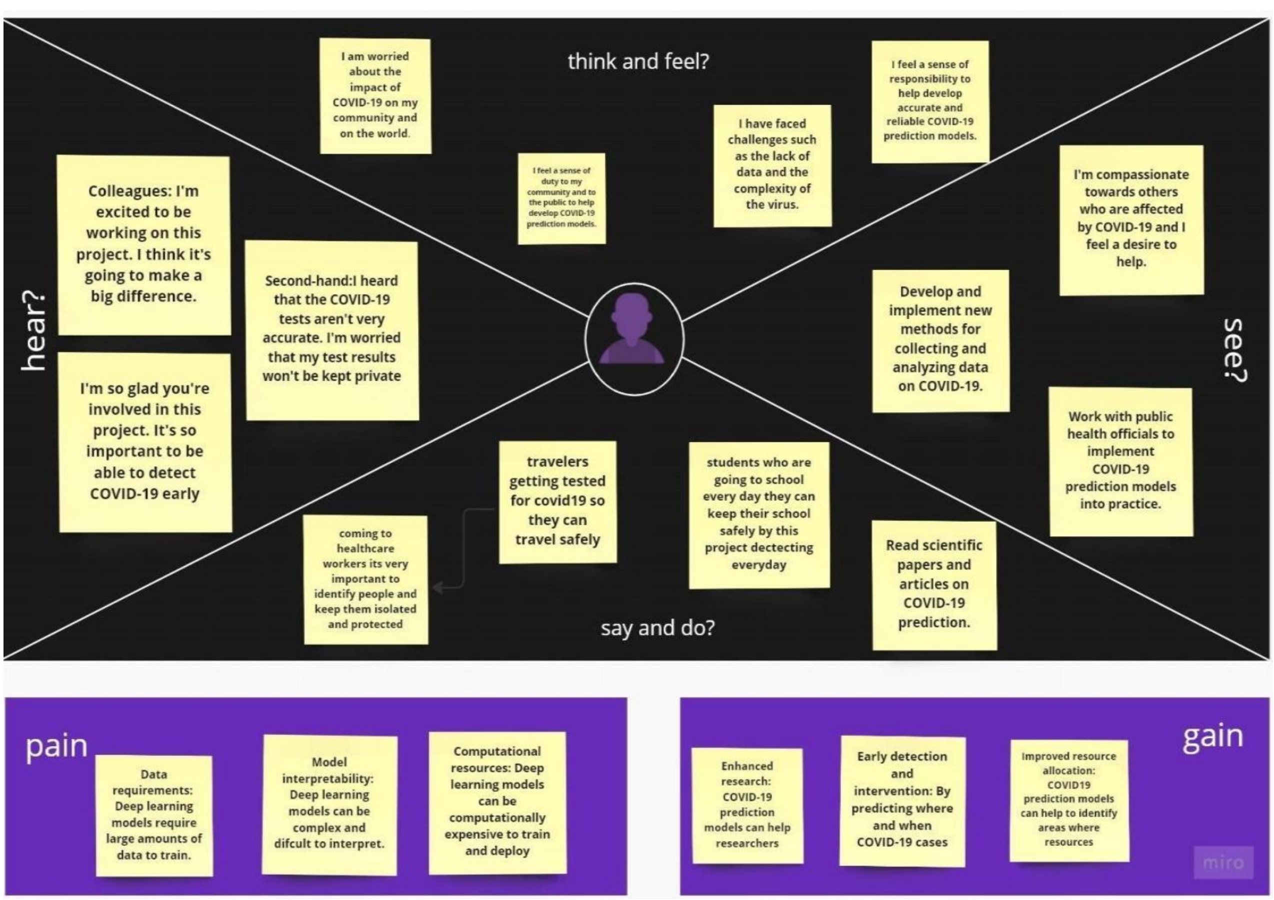
An empathy map is a simple, easy-to-digest visual that capturesknowledge about a user's behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the trueproblem and the person who is experiencing it. The exercise ofcreating the map helps participants consider things from the user'sperspective along with his or her goals and challenges.

Deep learning approach for predicting the covid19 virus :

Deep learning has been used to develop a number of different approaches for detecting the COVID-19 virus. These approaches typically involve training a deep learning model on a large dataset of images or other data from patients with and without COVID-19. Once the model is trained, it can be used to identify new cases of COVID-19 from new data

Example :



deep learning approach for predicting the covid19 virus Deep learning has been used to develop
a number of different approaches for detecting the COVID- 19 virus. These approaches typically involve training a deep learning model on a large dataset of images or other data from patients with and without COVID - 19 Once the model is trained it can be used to identity new cases of COVID - 19 from new data.

What do they HEAR?

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?

Travelers: Travelers may be required to gettested for COVID-19 before they can travel to or from certain countries even they required covid19 detection in their body

Students: Students may be tested for COVID-19 regularly at their schools or universities.these help them to identify or prediction of covid19 before they spred to otherstudents or staff members

GOAL
Healthcare workers: Healthcare workers are athigh risk of contracting COVID-19 because they are regularly exposed to the virus.so they need to check health everyday forthat they required detection of covid19
What do they THINK and FEEL?

PAINS GAINS

What do they need to DO?
How to allocate resources to different testing and surveillance strategiesHow to balance the need for accuracy withthe need for speed and accessibility
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?

The overall job of people involved in a COVID-19 detection project is to reduce the spread of COVID-19 and protect people from getting sick

Increase testing capacity: More people need to be tested for COVID- 19, more often.

People involved in COVID-19 detection projects may hear others talking about their
They may also read scientific journals and articles to stay up-to-date on the latest research on COVID-19 detection.

What are the you hearing others say? What are the y hearing fro m friends
?
What are the y hearing fro m colleagues? What are the y hearing second han d?

What are their fears,frustrations, and anxieties?

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

In marketplaces theysee such as Companies are developing new COVID-19 tests that are more accurate, faster, or less expensive than existing tests.
concerns about the pandemic, their experiences with COVID-19 testing, and their hopes for the future.

It is important to be aware of theseconcerns and to address them in asensitive and informative way. Enhanced research: COVID-19 prediction models can help researchers to better understand the transmission dynamicsof the virus.

Friends:
I'm so glad you're involved in this project. It's so important to be able to detect COVID-19early.
I'm a little worried about you working with COVID-19. Please be careful.

Model interpretability: Deep learning models can be complex and difficultto interpret.
Improved resource allocation: COVID- 19 prediction models can help to identify areas whereresources are most needed.
Early detection and intervention: By predicting where and when COVID-19 cases are likely to occur, publichealth officials can take early action to prevent the spread of the virus.

Data requirements: Deep learning models require large amounts ofdata to train.
What do they SEE?

Computational resources: Deep learning models canbe computationally expensive to train and deploy.
Help to identify andtrack the spread of new variants of the virus. andInform thedevelopment of vaccination and booster strategies.

What are they watching and reading?

Colleagues:
I'm excited to be workingon this project. I think it'sgoing to make a big difference.
I'm concerned about thesafety of our team. Whatsteps are we taking to protect ourselves?

What other thoughts and feelings might influence their behavior?

What do they SAY?
What have we heard them say?What can we magine them saying
People involved in COVID-19 detection projects may be motivated by a desireto help others and tomake a difference in the fight against the pandemic.

Researchers may be developing new COVID-19 tests, evaluating the accuracy of existing tests, or studying the spread of the virus.
and to
the
public.

Second-hand:I heard that the COVID-19 tests aren't very accurate.
I'm worried that my test results won't bekept private.
travelers getting tested for covid19 so they can travelsafely
coming to healthcare workersits very important toidentify people andkeep them isolatedand protected

People involved in

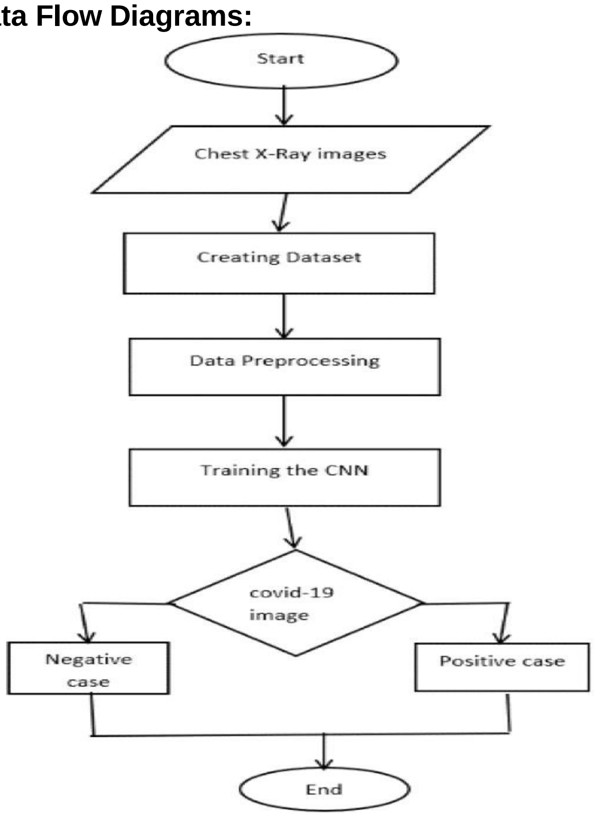
People may be anxious about thefuture and the impact of COVID-19 on their lives.

People with compromised immune systems may be tested for COVID-19 regularly or as needed to monitor their health.

Healthcare workers may collect samplesfrom patients, perform tests, and interpret test results.

Project Design Phase- I I Data **Flow Diagram &User Stories**

Data Flow Diagrams:



A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example: (Simplified)

- Data Acquisition: Input Chest X ray Output Raw X-ray images
- Preprocessing: Input Raw X-ray images Output :Preprocessed images
- Deep Learning Model Training: Input : Preprocessed images and COVID - 19 labels Output :Trained deep

learning model

- Prediction: Input New chest X-rays ,Trained model Output :Predicted COVID - 19 results
- Reporting and Storage: Input: Predicted results Output: Stored
- results and reporting to the Radiology Department or otherrelevant stakeholders.

Requirement analysis

Prerequisites:

To complete this project, you must require the following software's, concepts and packages

- **Anaconda navigator or google colab**
  - In google colab is the better option for doing deep learning projects for its pre installed packages for implementing or building the model of cnn and transfer learning
- **Python packages:**
  - Type "pip install numpy" and click enter.
  - Type "pip install pandas" and click enter..
  - Type "pip install tensorflow==2.3.2" and click enter.
  - Type "pip install keras==2.3.1" and click enter.
  - Type "pip install Flask" and click enter.

Prior Knowledge:

You must have prior knowledge of following topics to complete this project.

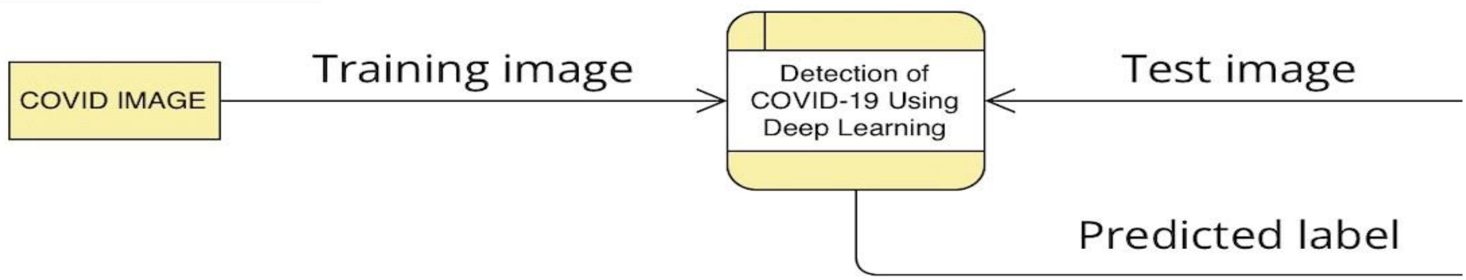
- **Deep Learning Concepts**
- **CNN:** https://towardsdatascience.com/basics-of-the-classic-cnn-a3dce1225add
- **VGG16:**

https://medium.com/@mygreatlearning/what-is-vgg16-introduction-to-vgg16-f2d63849f615

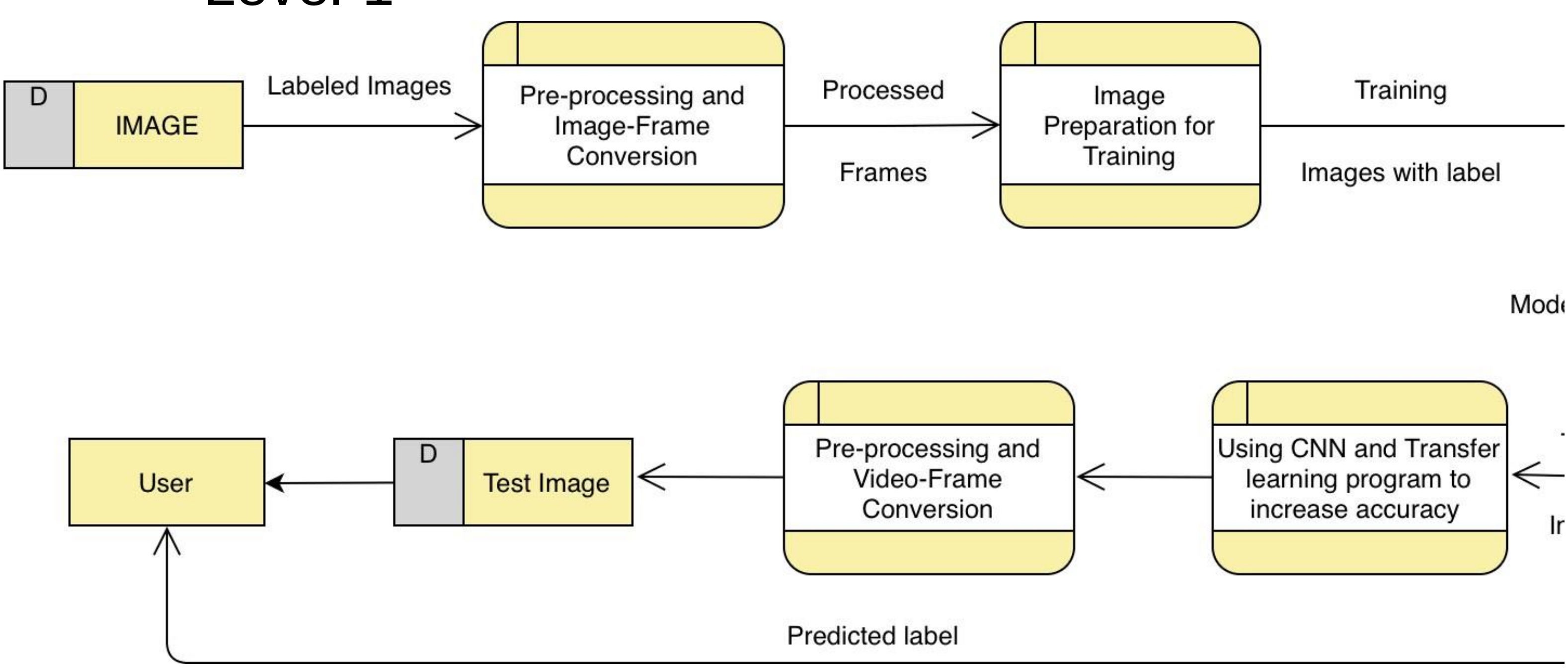
- **Flask:** Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.
  Link: **https://www.youtube.com/watch?v=lj4I_CvBnt0**

DATA FLOW DIAGRAMS :

- Level-0



- Level-1



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
Radiologist	upload a chestX-ray image	USN-1	I want to receive a COVID-19 predictionso that I can quickly and accurately identify potential COVID-19 cases and prioritize their treatment.	I can access myuploaded images	High	Sprint -1
Patient	chest X-ray processed bythe system	USN-2	I want to determine if I have COVID-19, enabling early diagnosis and intervention for my health.	I can receive the processed images	Medium	Sprint -1
Hospital Administrator	system to maintain a secure and easily accessible	USN-3	I want the repository of processed X-raysand their corresponding results for	I can view the repository imagesfor x-rays	Low	Sprint -2

			auditing, reporting, and quality control purposes.			
Deep Learning Researcher	labeled chest X-ray datasets	USN-4	I want access to the ability to retrain the deep learning model with new data to continuously improve its accuracy and performance.	I can access thedeep learning model	Medium	Sprint-1
Radiology Technician	Pre-process X-ray images, such as resizing and normalization	USN-5	I want the system to ensure that they arein the optimal format for deep learning analysis.	I can access thedeep learning analysis	Low	Sprint-1

Healthcare Regulator	comply with data privacy regulations andsecurity	USN-6	I want the system to protect patients' sensitive medical information.	I can secure the patients medical information	High	Sprint-2
Developer	well-documented	USN	I want access system that	I can view detection	Mediu	Sprin

	API	-7	allows me to integrate the COVID-19 detection systemwith our hospital's electronic health record (EHR) system for seamless patient management.	systemand EHR	m	t-3
Quality Assurance Specialist	run performancetests and validation	USN-8	I want the ability of the deep learning model to ensure that it meets accuracyand reliability standards.	I can perform testion for the data	High	Sprint-2
Research Institution	collaborate withthe system developers	USN-9	I want to investigate and validate the performance and reliability of the deep learning model on a diverse range of chest X-ray datasets.	I can go through the datasets of theimage	Low	Sprint-3
Data Scientist	comprehensiveset of performance metrics	USN-10	I want access to a metrics and visualizatio	I can access themetrics data of visualizati	High	Sprint-1

			ns for evaluating the deep learning model's performance and assisting in the model's continuous improvement.	on of deep learning		

Project Design Phase- ISolution Architecture

Solution Architecture:

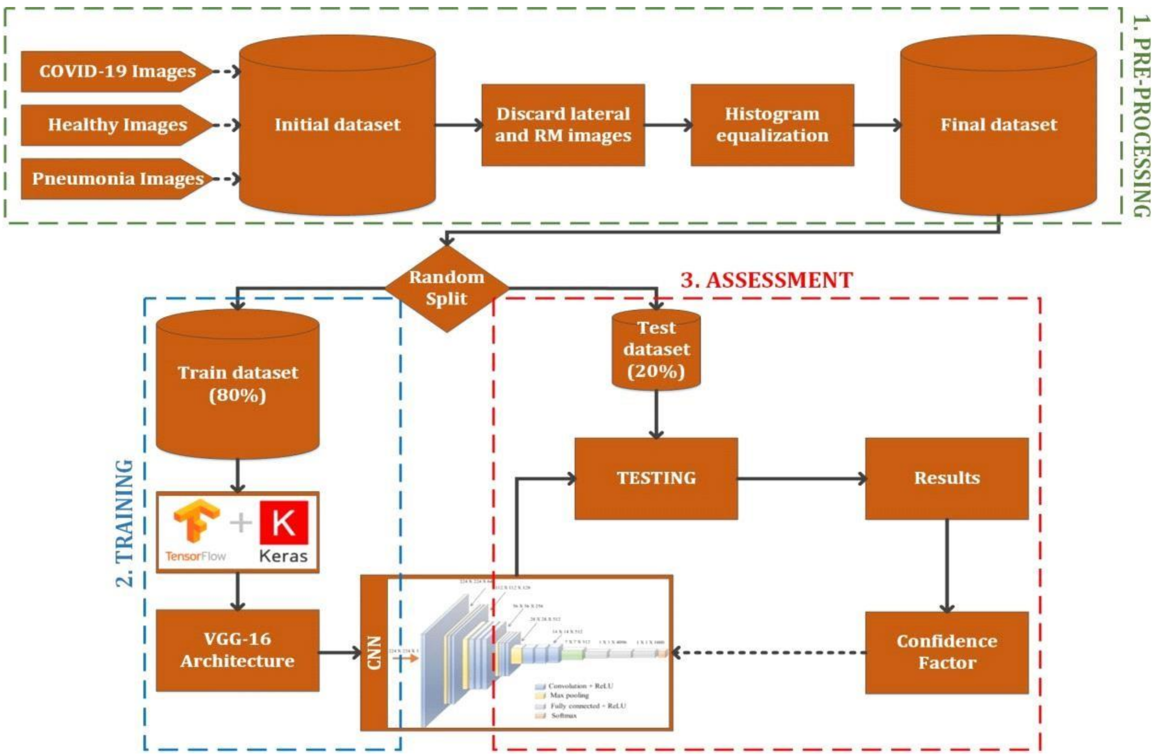
Model architecture: This component defines the specific architecture of the CNN deep learning model. There are many different CNN architectures that can be used for COVID-19detection, such as ResNet, VGGNet, and DenseNet.

Model training: This component trains the CNN deep learning model on the preprocesseddata. The training process involves feeding the model batches of images and labels, and adjusting the model's parameters to minimize the loss function.

Transfer learning: This component extracts the features learned from a pre-trained CNN deeplearning model and uses them to initialize the CNN deep learning model for the COVID-19 detection project.

Model evaluation: This component evaluates the performance of the trained CNN deep learning model on a held-out test set. The evaluation metrics used may include accuracy,precision, recall, and F1 score.

Solution Architecture Diagram



Guidelines:

1. Include all the processes (As an application logic / TechnologyBlock)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Technical Architecture:

Project Design Phase-II Technology Stack (Architecture & Stack)

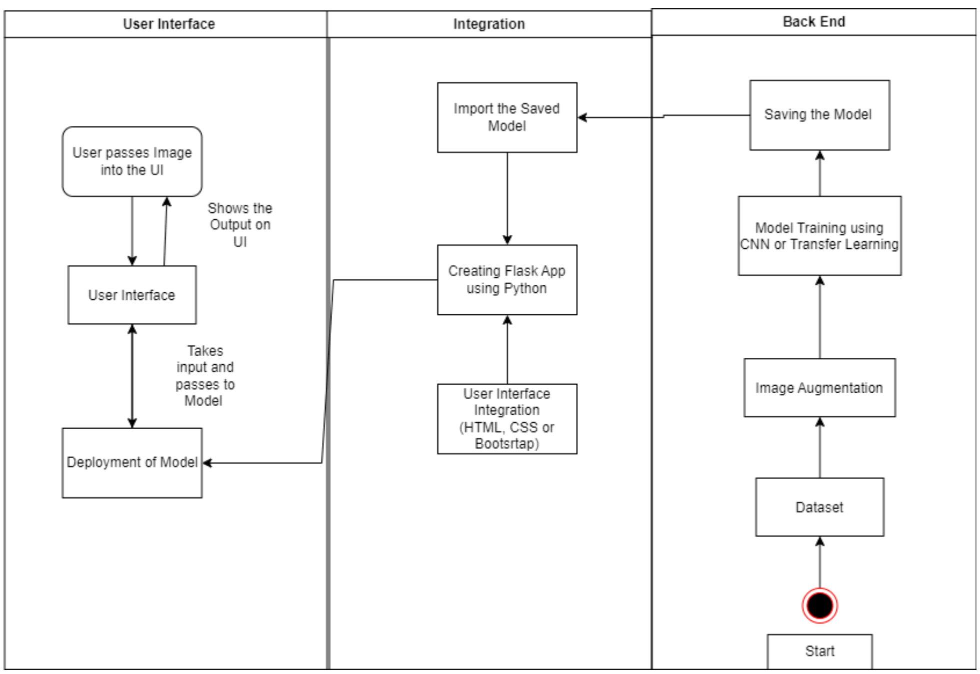


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.Web UI	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Database	Collect the Dataset Based on the Problem Statement	File Manager,MySQL

4.	File Storage/ Data	File storage requirements for Storing the dataset	Local System, Google Drive
5.	Frame Work	Used to Create a web Application, Integrating Frontend and Back End	PythonFlask, Django
6.	Deep Learning Model	Purpose of Model	CNN, Transfer Learning, etc.
7.	Infrastructure(Server / Cloud)	Application Deployment on Local System / CloudLocal Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Image Classification and Interpretability:	Identifying COVID-19 in medical images with explainable AI.	CNNs, pre-trained models,interpretability tools
2.	Scalability and Interoperability:	Scaling the system and seamless integration with healthcare platforms.	Cloud, containers, APIs, and interoperability standards.
3.	Data Security and Compliance:	Protecting patient data and adhering to healthcareregulations.	Encryption, access controls, compliance tools
4.	Continuous Monitoring and Collaboration:	Ongoing model evaluation and healthcareexpert involvement.	Monitoring tools, automation, and secure collaboration platforms.

S.No	Characteristics	Description	Technology
5.	Ethical & Transparent AI	Addressing ethics and maintaining transparency in healthcare AI	Ethical guidelines, expert input,and thorough documentation.

HYPER

LINK

"https://w

ww.l

eanix.n
et/en/wi
ki/ea/te
chnical
-
-
archite
cture"R
eferenc
es: https://aws.amazon.com/architecture

https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture- diagrams-2d20c9fda90d

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

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story points	Priority	Team members
Sprint-1	upload a chest X-ray image	USN-1	I want to receive a COVID-19 prediction so that Ican quickly and accurately	1	High	Balu

			identify potential COVID-19 cases and prioritize their treatment.			
Sprint-1	chest X-ray processed by the system	USN-2	I want to determine if I have COVID-19, enabling early diagnosis and intervention for my health.	2	Medium	Adarsh
Sprint-2	labeled chest X-ray datasets	USN-4	I want access to the ability to retrain the deeplearning model with new data to continuously improve its accuracy and performance.	2	Medium	Balu

Sprint-2	Pre-process X-rayimages, such as resizing and normalization	USN-5	I want the system to ensure that they are in the optimal format for deep learning analysis.	3	Low	David
Sprint-3	comprehensive setof performance metrics	USN-10	I want access to a metrics and visualizations for evaluating the deep learning model's performance and assisting in the model's continuous improvement.	4	High	Adarsh
Sprint-3	system to maintain a secure and easily accessible	USN-3	I want the repository of processed X-rays and their	3	Low	Varshath

			corresponding results for auditing, reporting, and quality control purposes.			
Sprint-4	comply with data privacy regulationsand security	USN-6	I want the system to protect patient s sensitive medic a	2	High	David

			I information.			
Sprint-4	run performance tests and validation	USN-8	I want the ability of the deep learning model to ensure that it meets accuracy and reliability standards.	3	High	Varshath
Sprint-5	well-documented API	USN-7	I want access system that allows me to integrate the COVID-19 detection system with our hospital's electronic health record (EHR) system for seamless patient management.	2	Medium	Balu

Sprint-5	collaborate with the system developers	USN-9	I want to investigate and validate the performanceand reliability of the deep learning model on a diverse range of chest X-ray datasets.	1	Low	Varshath

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	3	2 Days	18 oct 2023	20 oct 2023	20	18 oct 2023
Sprint-2	5	3 Days	21 oct 2023	23 oct 2023		
Sprint-3	7	5 Days	24 oct 2023	28 oct 2023		
Sprint-4	5	5 Days	29 oct 2023	2 nov 2023		
Sprint-5	3	2 Days	3 nov 2023	5 nov 2023		

Velocity:

Imagine we have a 29-days sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

AV= 29/20 = 1.45

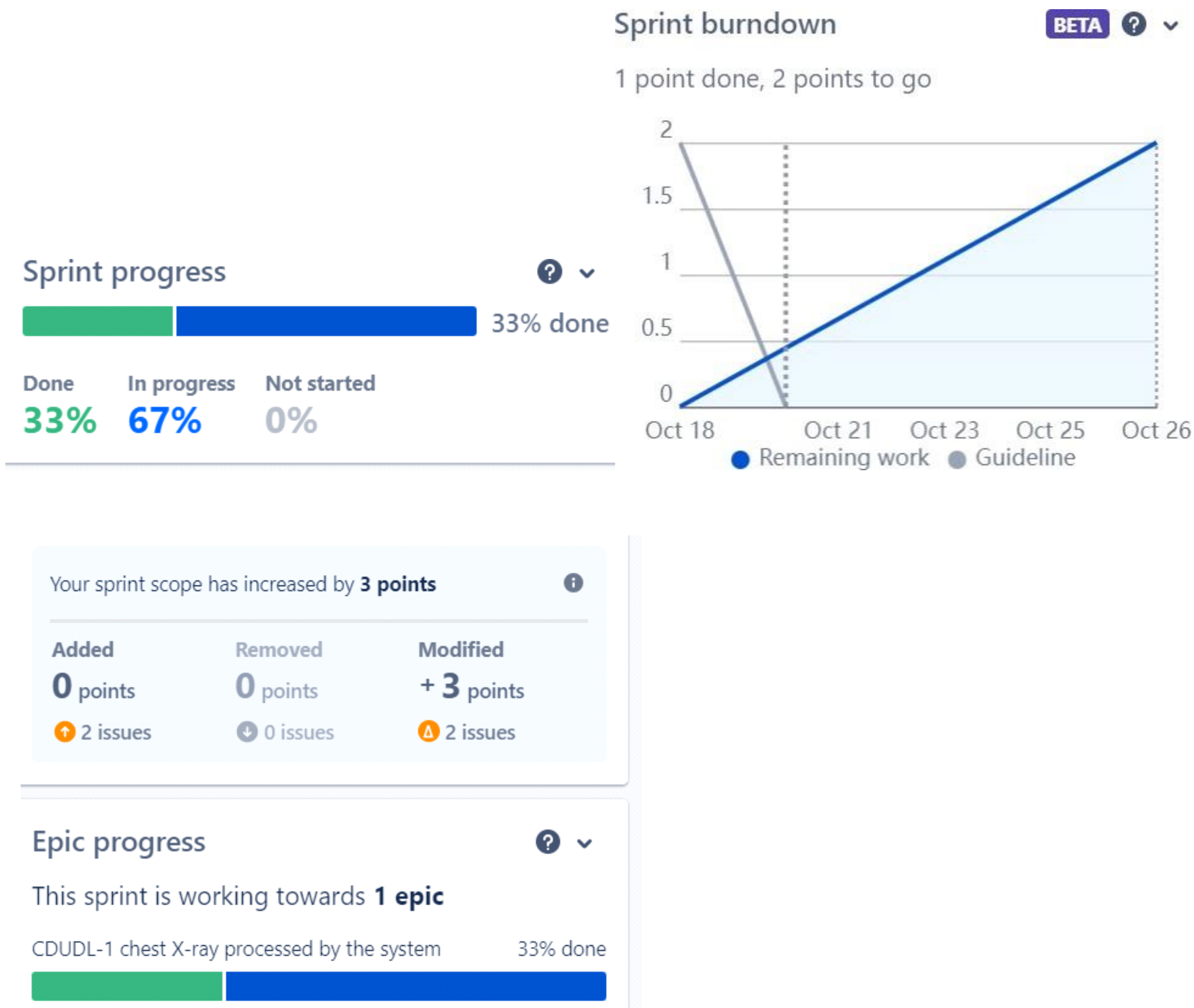
Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software HYPERLINK "https://www.visual-paradigm.com/scrum/what-is-agile-software-development/" HYPERLINK "https://www.visual-paradigm.com/scrum/what-is-agile-software-development/"development HYPERLINK "https://www.visual-paradigm.com/scrum/what-is-agile-software-development/" methodologies suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

https://www.visual-paradigm.com/scrum/scrum-burndown-chart/
https://www.atlassian.com/agile/tutorials/burndown-charts

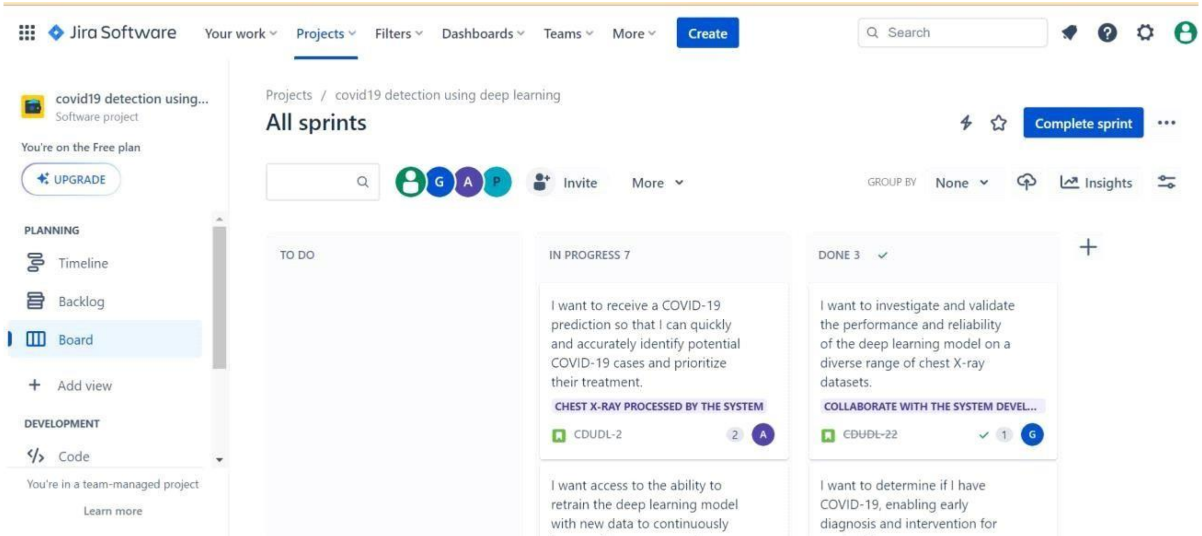
Reference:
https://www.atlassian.com/agile/project-management
https://www.atlassian.com/agile/tutorials/how-to-do-scrum-with-jira-software
https://www.atlassian.com/agile/tutorials/epics
https://www.atlassian.com/agile/tutorials/sprints
https://www.atlassian.com/agile/project-management/estimationhttps://www.atlassian.com/agile/tutorials/burndown-chart

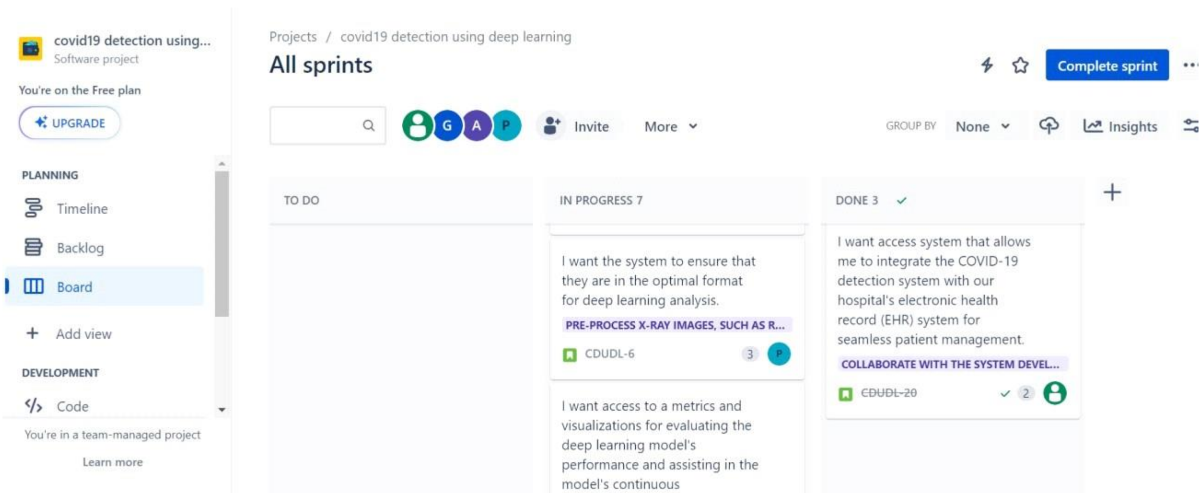
Burndown Chart:



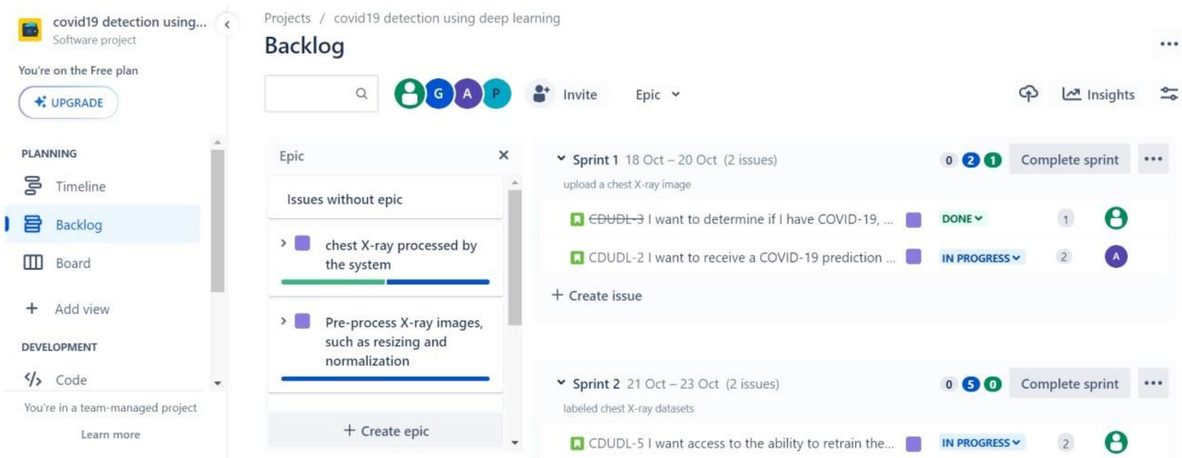
Board section.

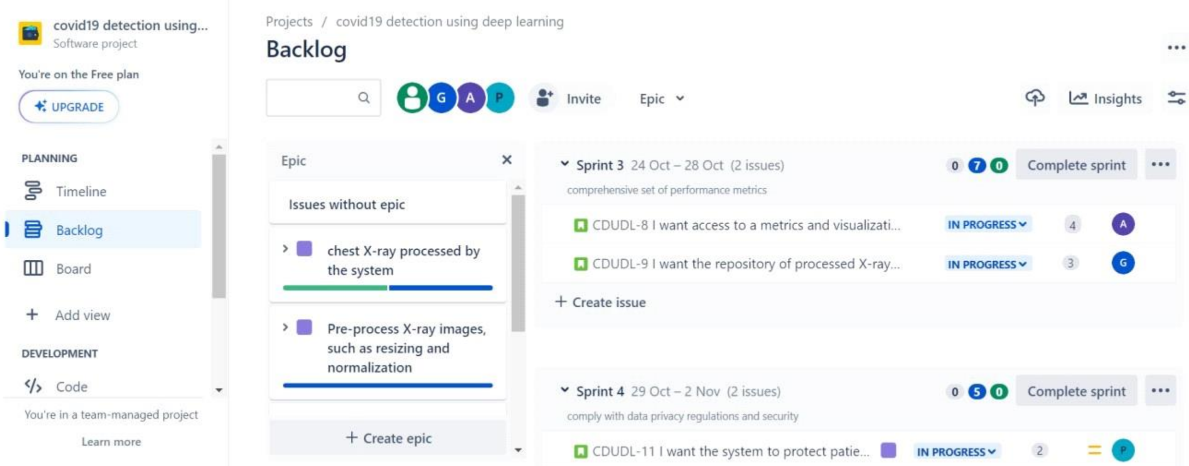
We have completed sprint 1 and 2. So we can see the remaining tasks on board.



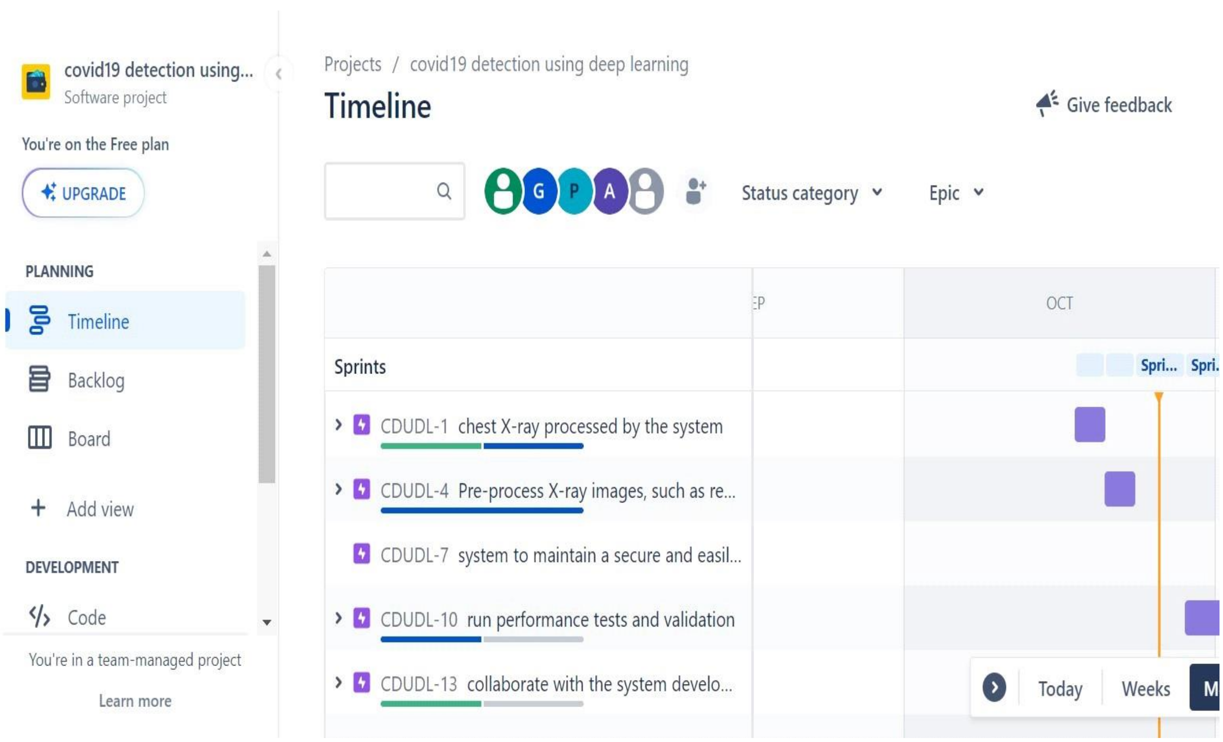


Backlog section


covid19 detection using...
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
You're in a team-managed project
Learn more
Projects / covid19 detection using deep learning
Backlog
Invite
Epic
Insights
Epic
Issues without epic
chest X-ray processed by the system
Pre-process X-ray images, such as resizing and normalization
Create epic
Sprint 1 18 Oct – 20 Oct (2 issues)
upload a chest X-ray image
Complete sprint
CDUDL-3 I want to determine if I have COVID-19, ...
DONE
CDUDL-2 I want to receive a COVID-19 prediction ...
IN PROGRESS
Create issue
Sprint 2 21 Oct – 23 Oct (2 issues)
labeled chest X-ray datasets
Complete sprint
CDUDL-5 I want access to the ability to retrain the...
IN PROGRESS


covid19 detection using...
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
You're in a team-managed project
Learn more
Projects / covid19 detection using deep learning
Backlog
Invite
Epic
Insights
Epic
Issues without epic
chest X-ray processed by the system
Pre-process X-ray images, such as resizing and normalization
Create epic
Sprint 3 24 Oct – 28 Oct (2 issues)
comprehensive set of performance metrics
Complete sprint
CDUDL-8 I want access to a metrics and visualizati...
IN PROGRESS
CDUDL-9 I want the repository of processed X-ray...
IN PROGRESS
Create issue
Sprint 4 29 Oct – 2 Nov (2 issues)
comply with data privacy regulations and security
Complete sprint
CDUDL-11 I want the system to protect patie...
IN PROGRESS

Timeline


covid19 detection using...
Software project
You're on the Free plan
UPGRADE
PLANNING
Timeline
Backlog
Board
Add view
DEVELOPMENT
Code
You're in a team-managed project
Learn more
Projects / covid19 detection using deep learning
Timeline
Give feedback
Status category
Epic
OCT
Sprints
CDUDL-1 chest X-ray processed by the system
CDUDL-4 Pre-process X-ray images, such as re...
CDUDL-7 system to maintain a secure and easil...
CDUDL-10 run performance tests and validation
CDUDL-13 collaborate with the system develo...
Today
Weeks

6. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

- **CNN Model**

Implement the necessary libraries for cnn deep learning algorithm These statements are importing modules and classes from the TensorFlow Keras library for building convolutional neural networks (CNNs).
from tensorflow.keras.models

import Sequential: This statement imports the Sequential class from the tensorflow.keras.models module.

The Sequential class is used to define and build CNN models by stacking layers sequentially.from tensorflow.keras.layers import Convolution2D, MaxPooling2D,Flatten,Dense: This statement imports several classes from the tensorflow.keras.layers module:

Convolution2D: This class represents a convolutional layer, which is a core building block of CNNs. It
performs convolutions to extract features from input images.

MaxPooling2D: This class represents a max-pooling layer, which performs downsampling by taking
the maximum value from a rectangular region of the input feature maps.

Flatten: This class represents a flattening layer, which converts the output of a convolutional layer into
a one-dimensional vector suitable for fully connected layers.

Dense: This class represents a fully connected layer, which performs linear transformations on the input
data. It is commonly used in the final stages of CNNs for classification or regression tasks.

CNN Model building

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D,Flatten,Dense
```

```
# adding layers

model = Sequential()

model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))  #convolution layer
model.add(MaxPooling2D(pool_size =(2,2)))  # maxpooling layer
model.add(Flatten())  # flatten layer

model.add(Dense(300,activation ='relu')) # hidden layer 1
model.add(Dense(150,activation ='relu')) # hidden layer 2

model.add(Dense(3,activation ='softmax')) # output layer
```

```
# compile the model
model.compile(optimizer = 'adam',loss= 'categorical_crossentropy',metrics =['accuracy'])
```

- **VGG 16 Model**

To build a DL model we have to split training and testing data into two separate folders. But In the project dataset folder training and testing folders are presented. So, in this case we just have to assign a variable and pass the folder path to it. Four different transfer learning models are used in our project and the best model (vgg16) is selected. The image input size of vgg16 model is 224,224.

implementing transfer learning using vgg16

```
trainpath = '/content/drive/MyDrive/covid19_prediction /Data/train'
testpath = '/content/drive/MyDrive/covid19_prediction /Data/test'
```

```
train_datagen = ImageDataGenerator(rescale = 1./255,zoom_range= 0.2,shear_range= 0.2)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

```
train = train_datagen.flow_from_directory(trainpath,target_size =(224,224),batch_size = 16)
test = test_datagen.flow_from_directory(testpath,target_size =(224,224),batch_size = 16)
```

```
Found 5144 images belonging to 3 classes.
Found 1288 images belonging to 3 classes.
```

```
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
```

```python
vgg = VGG16(include_top = False,input_shape = (224,224,3))
```

```python
[ ] for layer in vgg.layers:
      print(layer)
```

```
<keras.src.engine.input_layer.InputLayer object at 0x7805b2e716f0>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x780494329960>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2e735e0>
<keras.src.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7805301dbd30>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x78054cbec8e0>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x78049432a710>
<keras.src.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7805b2fbd870>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2e73a30>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2fbe770>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x78049432a950>
<keras.src.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7805b2fbc490>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2fbe6e0>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2fbecb0>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x78049432a4a0>
<keras.src.layers.pooling.max_pooling2d.MaxPooling2D object at 0x7804943396c0>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2fbff70>
<keras.src.layers.convolutional.conv2d.Conv2D object at 0x7805b2fbee60>
```

7.2 Feature 2

- HTML

```html
Users > adarshvenkatagattamaneni > Downloads > flask2 > templates > index.html > html > head
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>covid19_prediction</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
     <element class="bg-dark"> :ic', filename='css/main.css') }}" rel="stylesheet">

     Selector Specificity: (0, 1, 0)
    .bg-dark {
        background-color:   #1a1552!important;
    }
    #result {
        color: #1a1552;
    }
    body
{
    background-image: url("https://wallpaperaccess.com/full/327148.jpg");
    background-size: cover;
}

    </style>
</head>

<body>

    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
            <a class="navbar-brand" href="#">prediction of covid 19</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">
        <div class="container">
          <div class="row">
            <div class="col-sm-6 bd">
              <h3>covid19 </h3>
              <br>
              <p>Detecting COVID-19 using Convolutional Neural Networks (CNNs) and the VGG16 architecture has emerged as a promising approach in the fight against the pandemic. CNNs are a type of deep learnin
            </p>
                <img src="/covid19.jpeg" height="50%",width="20%">
            </div>
```

```html
        <div class="row">
          <div class="col-sm-6 bd">
            <h3>covid19 </h3>
            <br>
            <p>Detecting COVID-19 using Convolutional Neural Networks (CNNs) and the VGG16 architecture has emerged as a promising approach in the fight against the pandemic. CNNs are a type of deep learnin
          </p>
              <img src="/covid19.jpeg" height="50%",width="20%">
          </div>
          <div class="col-sm-6">
              <div>
                  <h4>Upload Image of lung scan for covid19 prediction</h4>
          <form action = "http://localhost:5000/" id="upload-file" method="post" enctype="multipart/form-data">
              <label for="imageUpload" class="upload-label">
                  Choose...
              </label>
              <input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">
          </form>


          <div class="image-section" style="display:none;">
              <div class="img-preview">
                  <div id="imagePreview">
                  </div>
              </div>
              <div>
                  <button type="button" class="btn btn-success" id="btn-predict">Predict!</button>
              </div>
          </div>

          <div class="loader" style="display:none;"></div>

          <h3>
              <span id="result"> </span>
          </h3>

      </div>
          </div>

        </div>
      </div>
      </div>
    </div>
</body>

<footer>
    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>

</html>
```

- PYTHON

```python
# app.py > ...
model = load_model(r"covid.h5",compile = False)
@app.route('/')
def index():
    return render_template("index.html")

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method=='POST':
        f = request.files['image']
        basepath=os.path.dirname(__file__)
        filepath = os.path.join(basepath,'uploads',f.filename)
        f.save(filepath)

        img = image.load_img(filepath, target_size=(64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(model.predict(x), axis=1)
        index = ['covid19','normal','pneumonia']
        text = f"The classified prediction is : {str(index[pred[0]])}"
        return text


if __name__=='__main__':
    app.run(debug=True)
```

PERFORMANCE TESTING

8.1 Performace Metrics

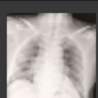
Testing the CNN model :

Testing is the process of evaluating the performance of a deep learning model on a dataset that it has not seen before. It is a crucial step in the development of any machine learning model, as it helps to determine how well the model can generalize to new data and also import necessary libraries We are giving covid19 image but it is predicting the pneumonia so we are getting wrong prediction which it effects the humans health so we are going with different algorithm transfer learning are greater accuracy for predicting covid19 .

```
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
[ ] # testing 1
    img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/test/COVID19/COVID19(573).jpg',target_size =(64,64))
```

```
[ ] img
```



```
[ ] x = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred =np.argmax(model.predict(x))
    op =['COVID19','NORMAL','PNEUMONIA']
    op[pred]
```

```
1/1 [==============================] - 0s 17ms/step
'PNEUMONIA'
```

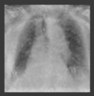
```
#traing the model
model.fit_generator(x_train, steps_per_epoch=len(x_train), epochs=20)
```

```
<ipython-input-159-b0d6195a6999>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `M
  model.fit_generator(x_train, steps_per_epoch=len(x_train), epochs=20)
Epoch 1/20
52/52 [==============================] - 69s 1s/step - loss: 0.1709 - accuracy: 0.9345
Epoch 2/20
52/52 [==============================] - 70s 1s/step - loss: 0.1783 - accuracy: 0.9364
Epoch 3/20
52/52 [==============================] - 68s 1s/step - loss: 0.1671 - accuracy: 0.9380
Epoch 4/20
52/52 [==============================] - 68s 1s/step - loss: 0.1686 - accuracy: 0.9382
Epoch 5/20
40/52 [=====================>.......] - ETA: 16s - loss: 0.1518 - accuracy: 0.9417
```

Testing the VGG16 model:

```
[ ] vgg16.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =['accuracy'])

[ ] vgg16.fit(train,validation_data=test,epochs=2,steps_per_epoch=len(train),validation_steps =len(test))

    Epoch 1/2
    322/322 [==============================] - 180s 549ms/step - loss: 0.2772 - accuracy: 0.9018 - val_loss: 0.2377 - val_accuracy: 0.9208
    Epoch 2/2
    322/322 [==============================] - 158s 492ms/step - loss: 0.1446 - accuracy: 0.9493 - val_loss: 0.1509 - val_accuracy: 0.9433
    <keras.src.callbacks.History at 0x780494340a90>

[ ] # saving  the model

    model.save('covid.h5')

    /usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file via `m
      saving_api.save_model(
```

```
[ ] from tensorflow.keras.preprocessing import image
    import numpy as np

[ ] # testing 1
    img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/test/COVID19/COVID19(464).jpg',target_size =(64,64))

[ ] img
```

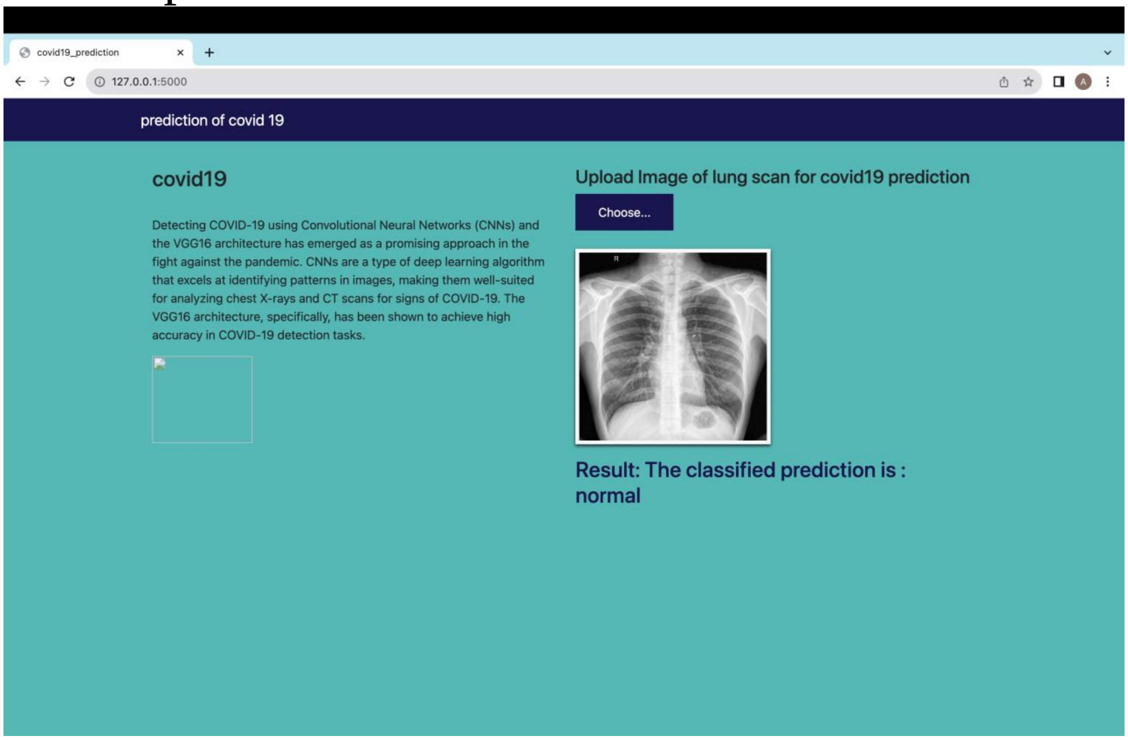


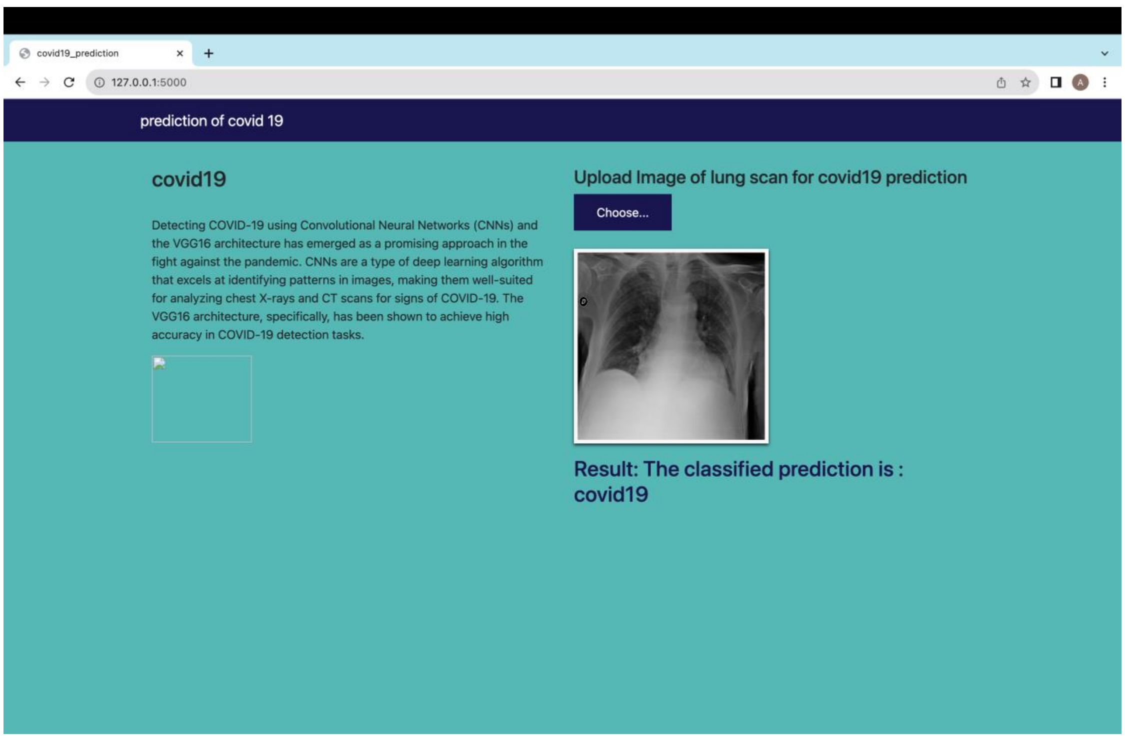
```
[ ] x = image.img_to_array(img)
    x = np.expand_dims(x,axis = 0)
    pred =np.argmax(model.predict(x))
    op =['COVID19','NORMAL','PNEUMONIA']
    op[pred]

    1/1 [==============================] - 0s 158ms/step
    'COVID19'
```

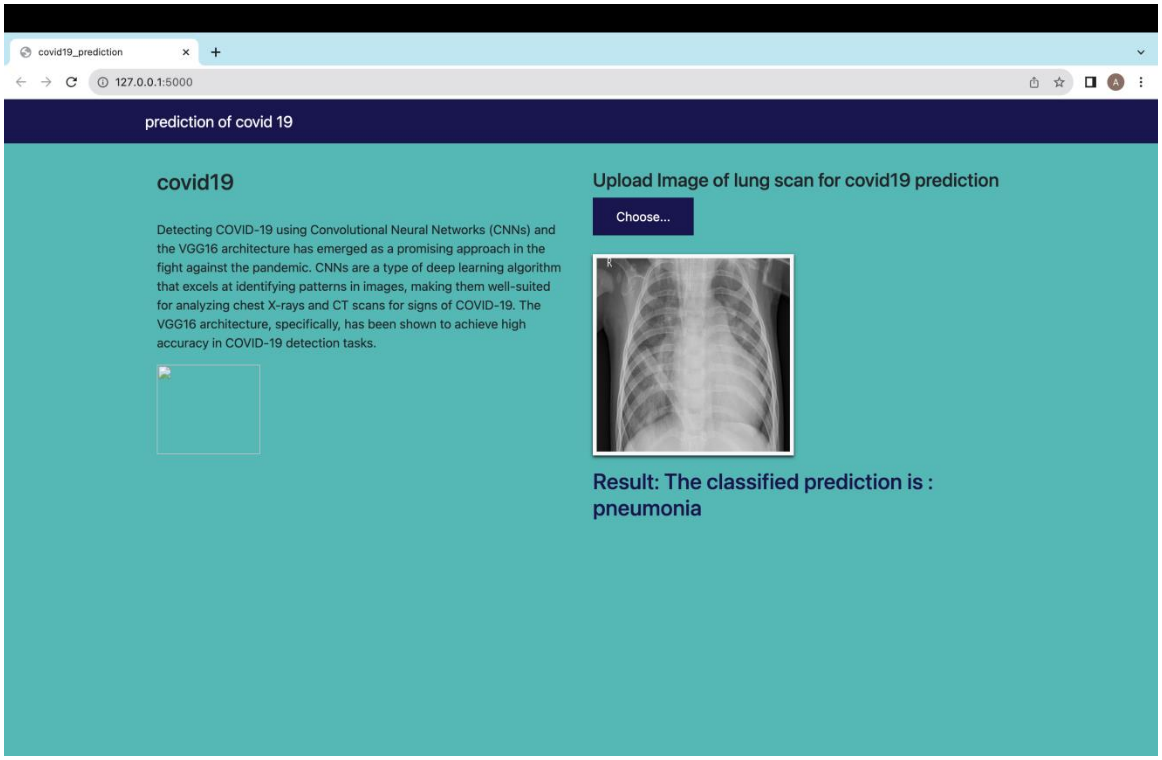
Therefore we are getting correct and greater accurate predictions by vgg16.

RESULTS

9.1 Output Screenshots







10. ADVANTAGES & DISADVANTAGES

Advantages of using deep learning techniques for COVID-19 detection:

- Deep learning models have been shown to achieve high accuracy in detecting COVID-19 from chest X-ray and CT scan images, even in the early stages of infection.
- Deep learning models can analyze images much faster than humans, which could help to reduce the time it takes to diagnose COVID-19.
- Deep learning models can be trained on large datasets of images, which can help to improve their accuracy and robustness.
- Deep learning models can be deployed on mobile devices and other low-cost hardware, making them more accessible to people in resource-limited settings.

Disadvantages of using deep learning techniques for COVID-19 detection:

- Deep learning models require large datasets of labeled images to train on. This can be challenging to obtain, especially for rare diseases such as COVID-19.
- Deep learning models can be biased, which can lead to inaccurate diagnoses for certain populations. It is important to carefully train and evaluate deep learning models to mitigate this problem.
- It can be difficult to understand how deep learning models make decisions, which can make it difficult to troubleshoot problems or identify potential biases.

11. CONCLUSION

Deep learning techniques have the potential to revolutionize COVID-19 detection by providing fast, accurate, and scalable solutions. However, it is important to be aware of the limitations of these techniques, such as the need for large datasets, the potential for bias, and the challenges of interpretability.

Despite these challenges, deep learning-based COVID-19 detection systems have the potential to play a major role in the fight against the pandemic. By automating the analysis of medical images, these systems can help to reduce the workload on healthcare workers and improve the speed and accuracy of diagnosis.

Overall, the research on COVID-19 detection using deep learning techniques is very promising. However, more work is needed to address the challenges of data requirements, model bias, and interpretability before these systems can be widely deployed in clinical settings.

12. FUTURE SCOPE

Overall, the future of deep learning for COVID-19 detection is very bright. As research in this area continues to advance, we can expect to see even more innovative and effective solutions emerge.

Here are some specific examples of future research directions in the area of COVID-19 detection using deep learning techniques:

- Development of deep learning models that can detect COVID-19 from other respiratory infections, such as influenza and pneumonia.
- Development of deep learning models that can predict the severity of COVID-19 infection and the risk of complications.

- Development of deep learning models that can be used to develop new diagnostic tests and treatments for COVID-19.

Integration of deep learning-based COVID-19 detection systems with other healthcare systems, such as electronic health records and clinical decision support systems.
By pursuing these research directions, we can accelerate the development of deep learning-based COVID-19 detection systems that can help to save lives and improve the quality of care for patients around the world.

13. APPENDIX

Source Code:

```
# import zipfile
# zip_path = "/content/drive/MyDrive/covid19_prediction /covid19_prediction.zip"  # Replace with the actual path to your zip file
# extract_path = "/content/drive/MyDrive/covid19_prediction "  # Replace with the desired path for the extracted files
# with zipfile.ZipFile(zip_path, 'r') as zip_ref:
#     zip_ref.extractall(extract_path)
from google.colab import drive
drive.mount('/content/drive')
##**Data Augmentation**
# import the nececessary lib
from tensorflow.keras.preprocessing.image import ImageDataGenerator
# data augmentation for the training variable

train_datagen = ImageDataGenerator(rescale =1./255,zoom_range=0.2,horizontal_flip = True)
# data augmentation for the testing variable

test_datagen = ImageDataGenerator(rescale =1./255)
# data augmentation on the training data


x_train = train_datagen.flow_from_directory('/content/drive/MyDrive/covid19_prediction /Data/train',
                                target_size=(64,64),
                                class_mode = 'categorical',
                                batch_size = 100)

# data augmentation on the testing data


x_test = test_datagen.flow_from_directory('/content/drive/MyDrive/covid19_prediction /Data/train',
                                target_size=(64,64),
                                class_mode = 'categorical',
                                batch_size = 100)
##**CNN Model building**
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D,Flatten,Dense
# adding layers

model = Sequential()
```

```
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))  #convolution layer
model.add(MaxPooling2D(pool_size =(2,2)))  # maxpooling layer
model.add(Flatten())  # flatten layer

model.add(Dense(300,activation ='relu')) # hidden layer 1
model.add(Dense(150,activation ='relu')) # hidden layer 2

model.add(Dense(3,activation ='softmax')) # output layer

# compile the model
model.compile(optimizer = 'adam',loss= 'categorical_crossentropy',metrics =['accuracy'])

#traing the model
model.fit_generator(x_train, steps_per_epoch=len(x_train), epochs=10)
# saving  the model

model.save('covid19.h5')
##**test the model**
from tensorflow.keras.preprocessing import image
import numpy as np
# testing 1
img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/test/COVID19/COVID19(573).jpg',target_size
=(64,64))
# test_image = tf.keras.preprocessing.image.load_img('path/to/test/image.jpg', target_size=(224, 224))
# test_image = tf.keras.preprocessing.image.img_to_array(test_image)
# test_image = np.expand_dims(test_image, axis=0)

# predictions = base_model.predict(test_image)
# print('Predicted class:', predictions)

img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
# testing 2
img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/train/COVID19/COVID19(104).jpg',target_size
=(64,64))
img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
# testing 3
img = image.load_img('/content/drive/MyDrive/covid19_prediction
/Data/train/NORMAL/NORMAL(1006).jpg',target_size =(64,64))
img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
```

```
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
##**evaluating a cnn model**
model.evaluate(x_test)
##**implementing transfer learning using vgg16**
trainpath = '/content/drive/MyDrive/covid19_prediction /Data/train'
testpath = '/content/drive/MyDrive/covid19_prediction /Data/test'

train_datagen = ImageDataGenerator(rescale = 1./255,zoom_range= 0.2,shear_range= 0.2)
test_datagen = ImageDataGenerator(rescale = 1./255)
train = train_datagen.flow_from_directory(trainpath,target_size =(224,224),batch_size = 16)
test = test_datagen.flow_from_directory(testpath,target_size =(224,224),batch_size = 16)

from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.layers import Dense, Flatten
from tensorflow.keras.models import Model
vgg = VGG16(include_top = False,input_shape = (224,224,3))
for layer in vgg.layers:
  print(layer)

len(vgg.layers)
for layer in vgg.layers:
  layer.trainable = False
x= Flatten()(vgg.output)
output = Dense(3, activation ='softmax')(x)
vgg16 = Model(vgg.input,output)

vgg16.summary()

vgg16.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics =['accuracy'])

vgg16.fit(train,validation_data=test,epochs=3,steps_per_epoch=len(train),validation_steps =len(test))

# saving  the model
model.save('covid.h5')
##**testing vgg16 mode**

!pip install tensorflow
import tensorflow as tf
from tensorflow.keras.preprocessing import image
import numpy as np
# testing 1
img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/train/COVID19/COVID19(101).jpg',target_size
=(64,64))
# test_image = tf.keras.preprocessing.image.load_img('/content/drive/MyDrive/covid19_prediction
/Data/train/COVID19/COVID19(100).jpg', target_size=(224, 224))
# test_image = tf.keras.preprocessing.image.img_to_array(test_image)
# test_image = np.expand_dims(test_image, axis=0)

# predictions = vgg.predict(test_image)
# print('Predicted class:', predictions)
```

```
img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
#testing 2
img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/test/COVID19/COVID19(473).jpg',target_size =(64,64))
img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
#testing3
img = image.load_img('/content/drive/MyDrive/covid19_prediction /Data/test/PNEUMONIA/PNEUMONIA(3434).jpg',target_size =(64,64,))
img
x = image.img_to_array(img)
x = np.expand_dims(x,axis = 0)
pred =np.argmax(model.predict(x))
op =['COVID19','NORMAL','PNEUMONIA']
op[pred]
```

GitHub Link :

https://github.com/smartinternz02/SI-GuidedProject-603219-1697624740

Project Demo Link:

https://drive.google.com/file/d/148M2jNv7e89uLnSZnKnQ2EuDBbmpPhA-/view?usp=drivesdk