GROUP 8

DETECTING COVID-19 FROM CHEST X-RAY IMAGES

USING CNN

TEAM MEMBERS

Alap S Suresh - AM.EN.U4AIE21110

Annapoorna A K - AM.EN.U4AIE21114

Gayathry M Wariyar - AM.EN.U4AIE21130

Navneeth Suresh - AM.EN.U4AIE21147

Shreya Pulluri - AM.EN.U4AIE21159

Varalakshmi M P - AM.EN.U4AIE21165

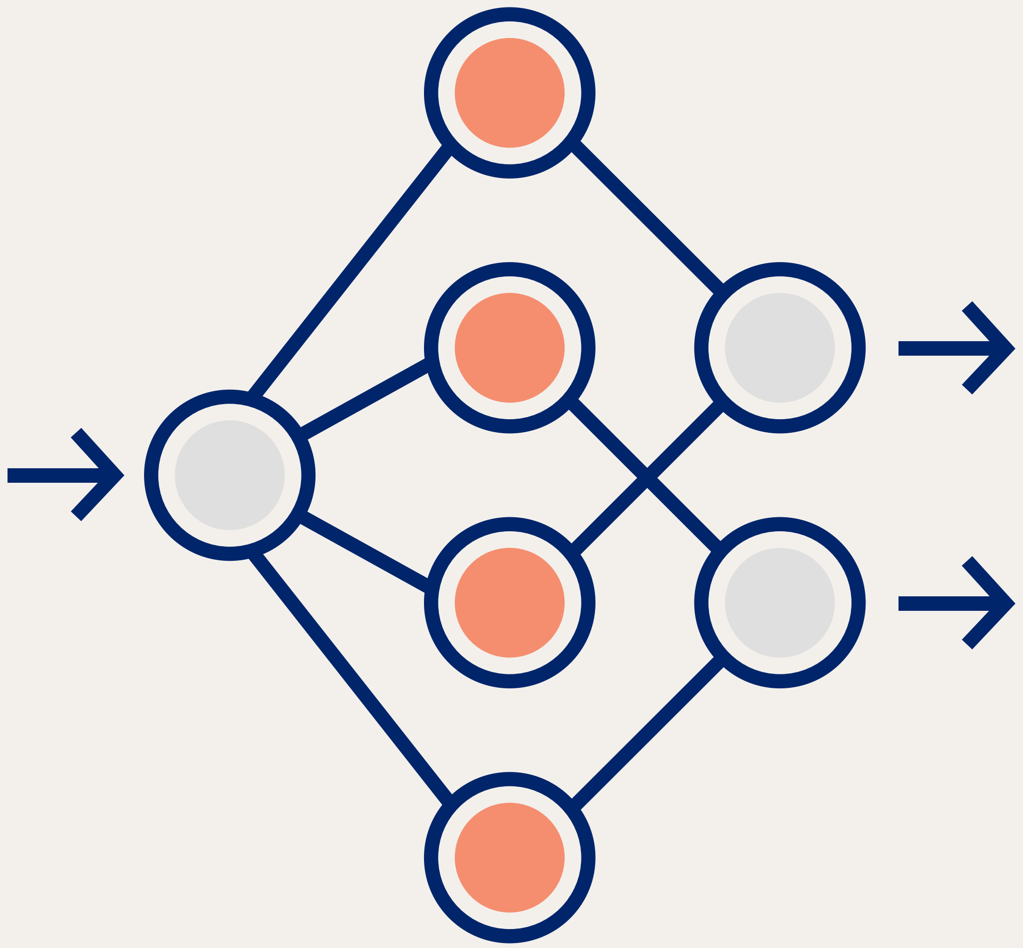
INTRODUCTION

- COVID-19 is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)-related pneumonia. The outbreak originated in Wuhan, China, and rapidly spread worldwide, overwhelming healthcare systems and straining medical resources.

- This project aims to develop and implement a Convolutional Neural Network (CNN) model for the detection of COVID-19 from chest X-ray images, to create a precise and efficient model that can significantly contribute to the early and accurate diagnosis of COVID-19, thereby facilitating effective patient management and curbing the transmission of the disease.

PROBLEM STATEMENT

- The objective of this project is to develop a precise and efficient model capable of early and accurate COVID-19 diagnosis. By achieving this, the model will play a crucial role in enabling effective patient management and containment of disease transmission, ultimately contributing to the control and mitigation of the COVID-19 pandemic.



- By training a CNN model on a large dataset of chest X-ray images, consisting of COVID-19-positive and negative cases, we seek to establish a reliable and efficient method for COVID-19 detection. The trained model will be able to classify new chest X-ray images as either COVID-19-positive or negative, thereby aiding healthcare professionals in making timely and informed decisions.

WHAT DO WE KNOW ABOUT CURRENT COVID-19 TESTING ?

- While the potential for rapid point-of-care COVID-19 tests is promising, the current turnaround times for test results vary from 3 to more than 48 hours, and not all regions may have access to such tests.

- To address this issue, the automated detection of COVID-19 using X-ray images could offer a valuable alternative, particularly for countries and hospitals without access to laboratory test kits or CT scanners.

- The accurate and timely diagnosis of COVID-19 is important as there are no definitive treatments for the disease.

COVID-19 AND ITS EFFECTS ON THE LUNGS

1) Virus Entry and Attachment:

- Once inside the lungs, the virus targets specific cells known as respiratory epithelial cells, which line the airways and air sacs.
- It uses a particular protein, the spike protein to attach itself to these cells. The spike protein binds to a receptor on the respiratory epithelial cells known as ACE2, allowing the virus to enter the cells.

2) Viral Replication and Inflammation:

- Once inside the cells, the virus replicates and multiplies.
- Viral replication damages host cells and triggers an inflammatory response.

3) Cytokine Storm:

- The inflammatory response involves immune cells and signalling molecules.
- Excessive cytokine release leads to a cytokine storm which causes severe inflammation in the lungs and other organs.

WHY CHEST X-RAY ?

Comparison of chest X-ray images and CT scans of the thorax for COVID-19 diagnosis:

Advantages of Chest X-ray Images:

- Faster diagnostic time compared to CT scans.
- Typically only takes a few minutes per patient.
- Well-suited for high patient throughput in busy healthcare settings.

Advantages of CT Scans of the Thorax:

- High-resolution imaging and detailed visualization of lung structures.
- Aids in identifying early COVID-19 manifestations.

Time Consumption of CT Scans:

- Zheng et al. (2020) reported an average of 25 minutes per patient for obtaining and interpreting CT scans.
- Prolonged patient contact duration and potential exposure of healthcare workers to infected individuals.

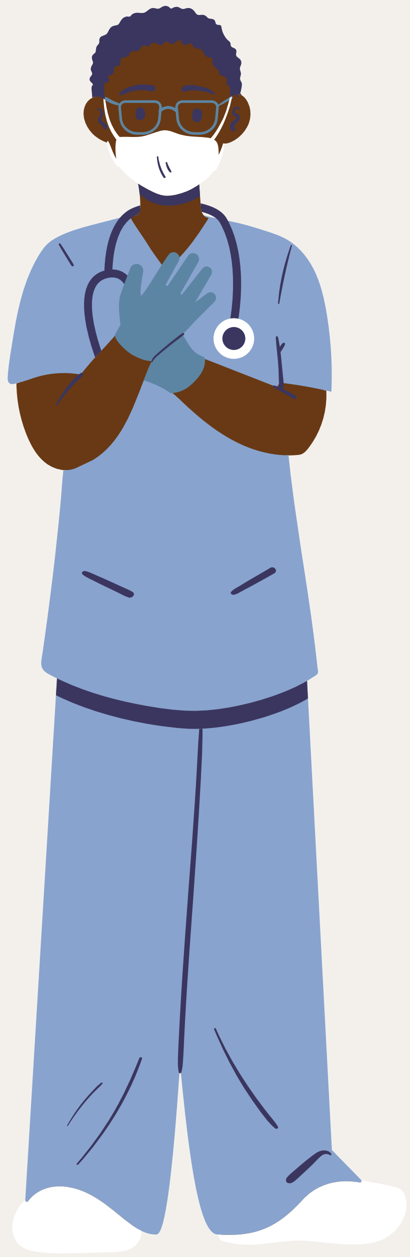
The efficiency of Chest X-ray Scans:

-A retrospective study comparing chest X-ray and CT scans of confirmed COVID-19 patients found that chest X-rays demonstrated a sensitivity of 69.2% and a specificity of 95.5%, indicating their potential as a valuable screening tool for suspected cases.

PATHOLOGICAL FEATURES ON IMAGING

1. **Ground-Glass Opacities:**
   - Hazy areas of increased fluid, inflammatory cells, and damaged tissue in the air sacs.
2. **Consolidation:**
   - Air-filled spaces in the lungs are filled with fluid, inflammatory cells, and debris, causing opacity in imaging.
3. **Bilateral Involvement:**
   - Both lungs are affected by COVID-19, observable on imaging.
4. **Patchy Infiltrates:**
   - Irregular patterns of abnormal lung tissue due to infection and inflammation.
5. **Severe Cases - Viral Pneumonia:**
   - The infection leads to viral pneumonia.
   - Inflammation in the air sacs impairs normal lung function.
   - It can result in Impaired oxygenation.

DATASET INFORMATION



- For this project on detecting COVID-19, we utilized a publicly available dataset sourced from Kaggle. The dataset, titled **"Chest X-Ray for COVID-19 Detection"** provides a valuable collection of chest X-ray images that include both **COVID-19-positive and COVID-19-negative cases.** It contains images sourced from various hospitals and medical institutions. All files are in jpeg/jpg/png format.
  Link: https://www.kaggle.com/datasets/fusicfenta/chest-xray-for-covid19-detection

- The COVID-19-positive category includes chest X-ray images of patients diagnosed with COVID-19, exhibiting characteristic signs and abnormalities associated with the disease. On the other hand, the COVID-19-negative category consists of chest X-ray images of clear lungs.

- The dataset is maintained in a directory named “Dataset” which is divided into 3 sub-directories named: “Prediction”, “Train” and “Val” (Validation). In Total the dataset consists of hundreds of Chest X-Ray images.

METHODOLOGY

1. ***Creating convolutional, max-pooling, and fully connected layers.***

- created multiple convolutional layers for feature extraction
- maxpooling layer is to downsample our feature map.
- dropout layer was to prevent overfitting by introducing some level of regularization.
- flatten layer converts the output into vectors to send it through the fully connected layers
- Dense, fully connected layers, deeply connected with its preceding layer and gives out the output of classified label.

CNN MODEL SUMMARY

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      896

 conv2d_1 (Conv2D)           (None, 220, 220, 64)      18496

 max_pooling2d (MaxPooling2  (None, 110, 110, 64)      0
 D)

 dropout (Dropout)           (None, 110, 110, 64)      0

 conv2d_2 (Conv2D)           (None, 108, 108, 64)      36928

 max_pooling2d_1 (MaxPoolin  (None, 54, 54, 64)        0
 g2D)

 dropout_1 (Dropout)         (None, 54, 54, 64)        0

 conv2d_3 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPoolin  (None, 26, 26, 128)       0
 g2D)

 dropout_2 (Dropout)         (None, 26, 26, 128)       0

 conv2d_4 (Conv2D)           (None, 24, 24, 128)       147584

 max_pooling2d_3 (MaxPoolin  (None, 12, 12, 128)       0
 g2D)

 dropout_3 (Dropout)         (None, 12, 12, 128)       0

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 64)                1179712

 dropout_4 (Dropout)         (None, 64)                0

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 1457537 (5.56 MB)
Trainable params: 1457537 (5.56 MB)
Non-trainable params: 0 (0.00 Byte)
```

2. DATA AUGMENTATION

```
train_datagen = image.ImageDataGenerator(
    rescale = 1./255,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
)
test_dataset = image.ImageDataGenerator(rescale = 1./255)
```

```
train_generator = train_datagen.flow_from_directory(
    'Dataset/Train',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary'
)
```

```
Found 288 images belonging to 2 classes.
```

- Two instances of ImageDataGenerator are used to perform data augmentation and rescaling pixel values for the training and testing datasets.
- Batches of augmented image data along with their corresponding labels are yielded by using the flow_from_directory method, during the training process. The images will be automatically resized, augmented, and normalized.
- A similar augmentation is done to generate batches of validation data from the val directory.

3. MODEL FITTING

The fit_generator method starts the training process, where the model's weights are updated iteratively based on the provided training data. After each epoch, the model's performance is evaluated using the validation data.

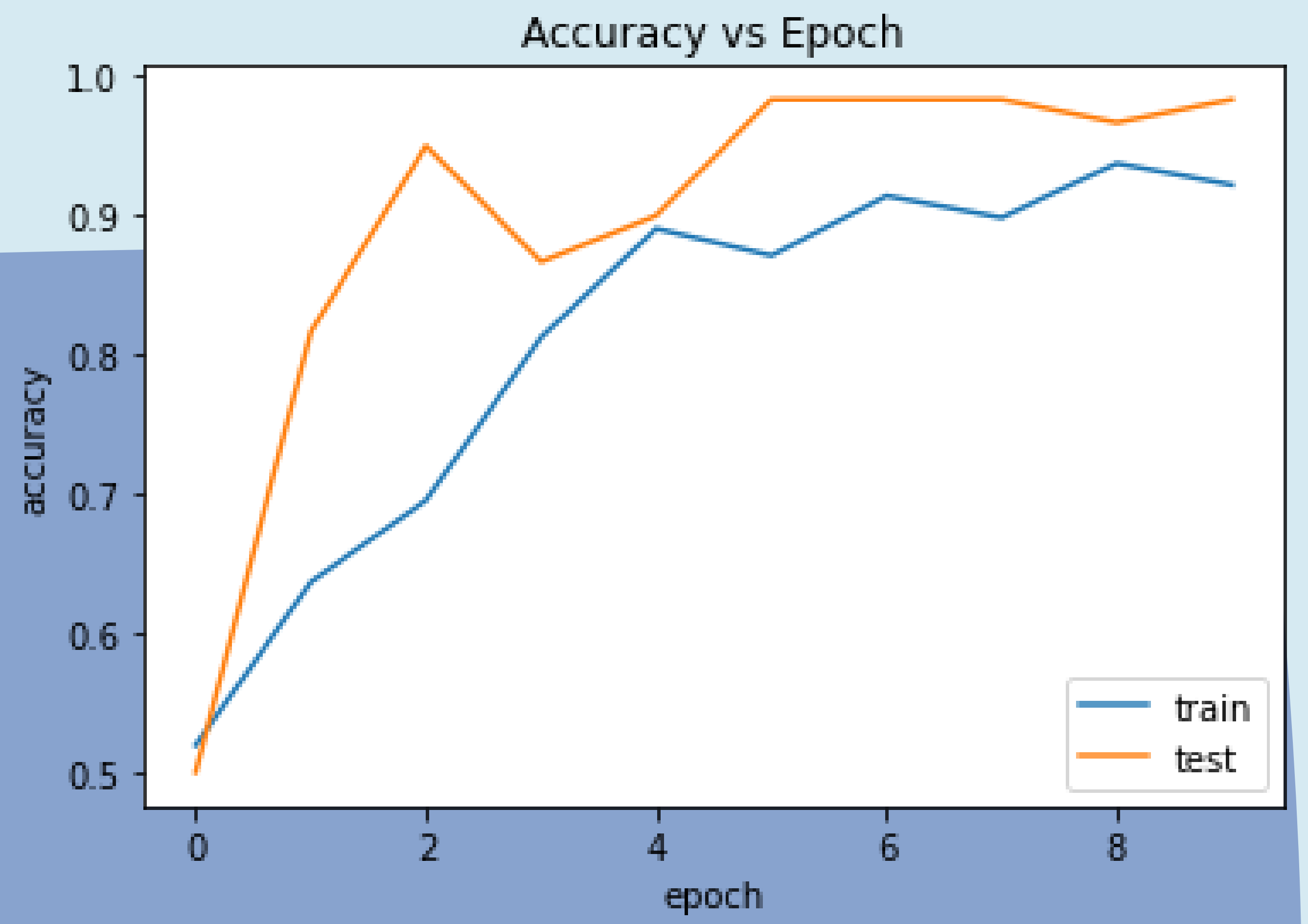
The variable hist will store the history of the training process, including the values of the defined metrics (such as accuracy and loss) at each epoch.

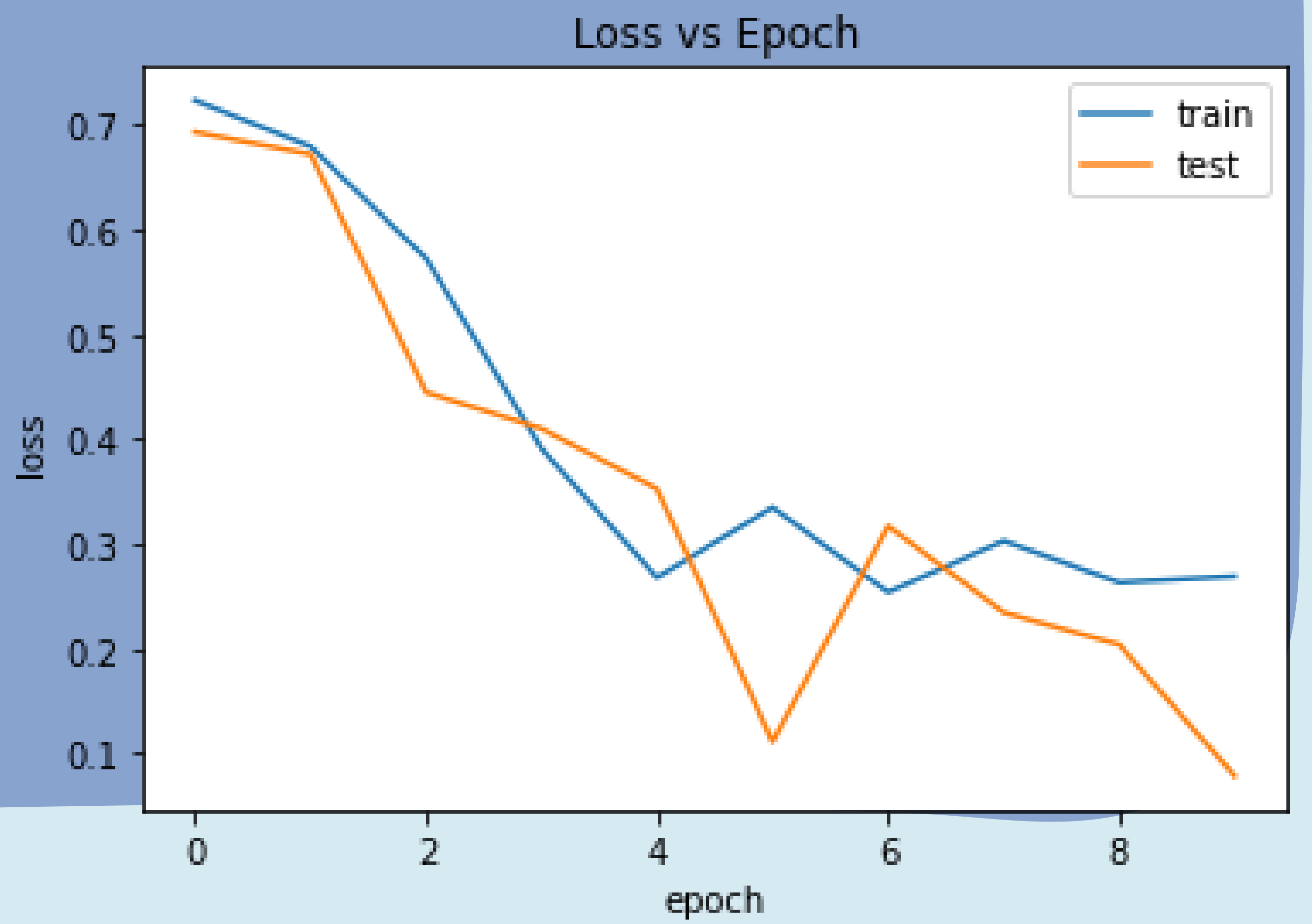
```
hist = model.fit_generator(
    train_generator,
    steps_per_epoch = 8,
    epochs = 10,
    validation_data = validation_generator,
    validation_steps = 2
)
```

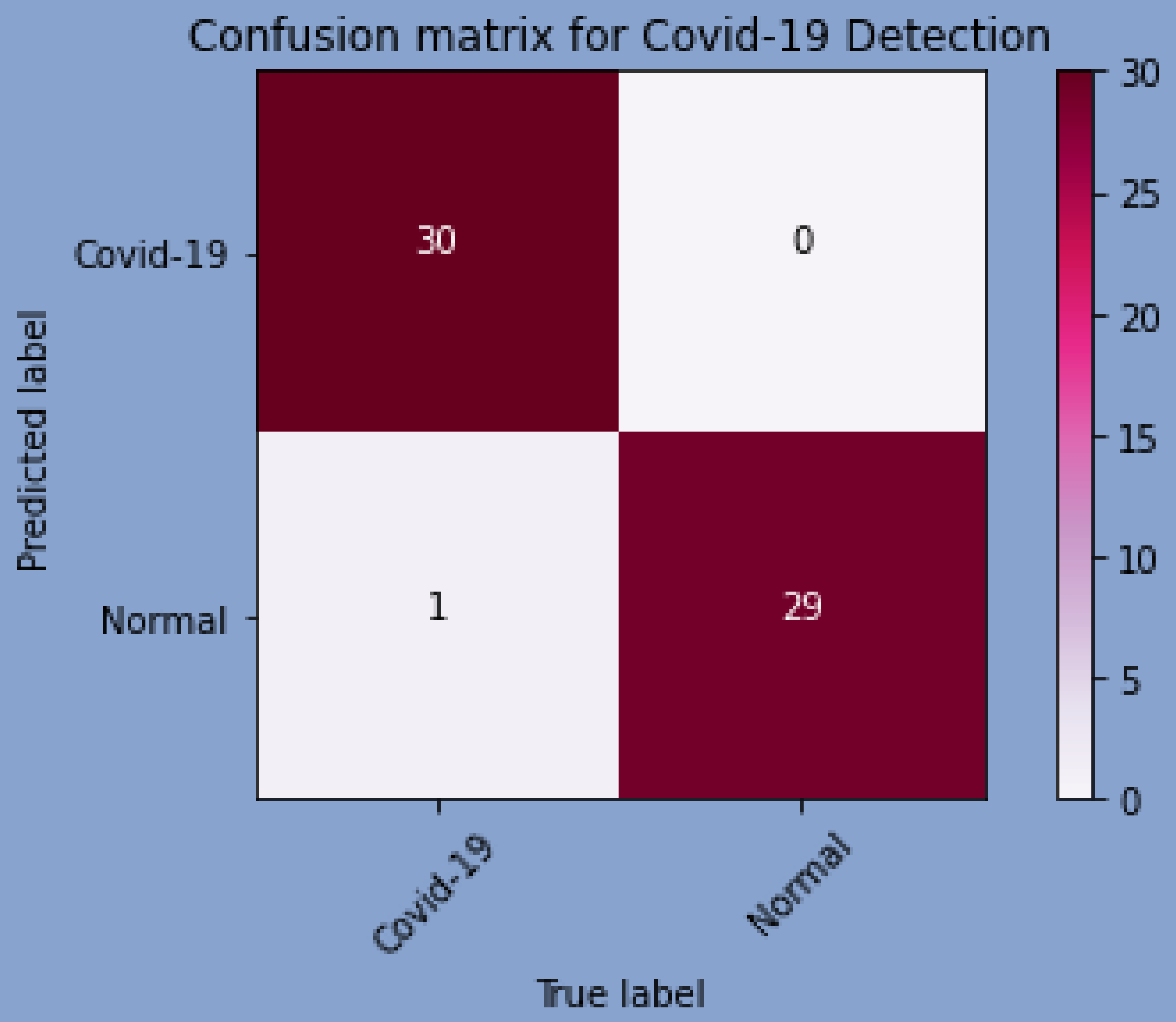
RESULTS

Accuracy increasing with each Epoch, while Model loss decreased steadily

```
Epoch 1/10
8/8 [==============================] - 24s 3s/step - loss: 0.7229 - accuracy: 0.5195 - val_loss: 0.6926 - val_accuracy: 0.5000
Epoch 2/10
8/8 [==============================] - 23s 3s/step - loss: 0.6789 - accuracy: 0.6367 - val_loss: 0.6715 - val_accuracy: 0.8167
Epoch 3/10
8/8 [==============================] - 28s 3s/step - loss: 0.5722 - accuracy: 0.6953 - val_loss: 0.4446 - val_accuracy: 0.9500
Epoch 4/10
8/8 [==============================] - 28s 4s/step - loss: 0.3909 - accuracy: 0.8125 - val_loss: 0.4099 - val_accuracy: 0.8667
Epoch 5/10
8/8 [==============================] - 25s 3s/step - loss: 0.2681 - accuracy: 0.8906 - val_loss: 0.3526 - val_accuracy: 0.9000
Epoch 6/10
8/8 [==============================] - 26s 3s/step - loss: 0.3350 - accuracy: 0.8711 - val_loss: 0.1114 - val_accuracy: 0.9833
Epoch 7/10
8/8 [==============================] - 25s 3s/step - loss: 0.2541 - accuracy: 0.9141 - val_loss: 0.3170 - val_accuracy: 0.9833
Epoch 8/10
8/8 [==============================] - 25s 3s/step - loss: 0.3030 - accuracy: 0.8984 - val_loss: 0.2348 - val_accuracy: 0.9833
Epoch 9/10
8/8 [==============================] - 25s 3s/step - loss: 0.2634 - accuracy: 0.9375 - val_loss: 0.2041 - val_accuracy: 0.9667
Epoch 10/10
8/8 [==============================] - 26s 3s/step - loss: 0.2693 - accuracy: 0.9219 - val_loss: 0.0789 - val_accuracy: 0.9833
```







The confusion matrix revealed that the model exhibited high sensitivity and specificity in detecting COVID-19 cases.

The model gave high TP(30) and TN(29) rates, with low FP(0) and FN(1) rates.

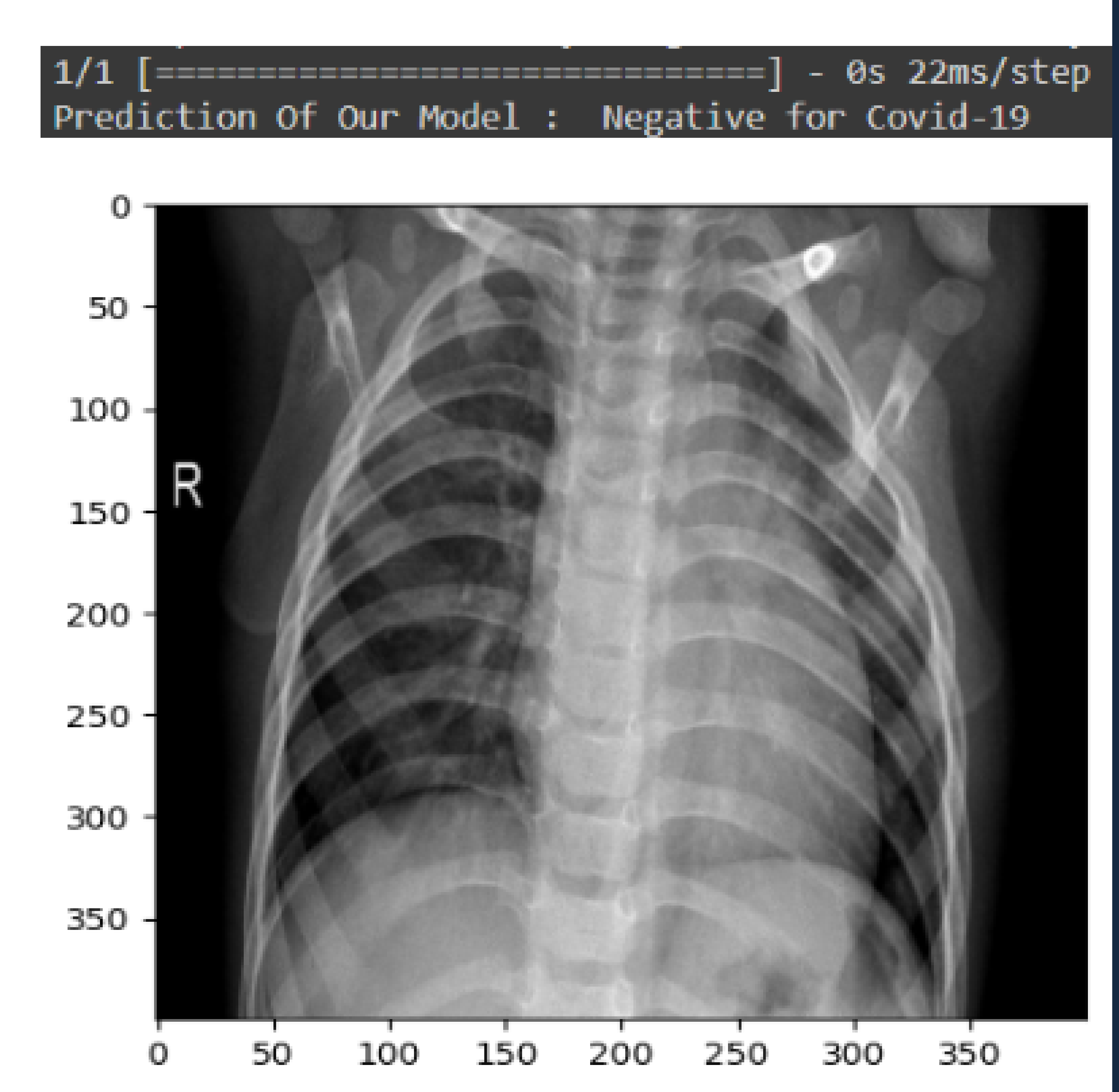
DISCUSSION

Model accuracy improved over iterations, starting from 51.95% and reaching 92.19% for testing dataset.

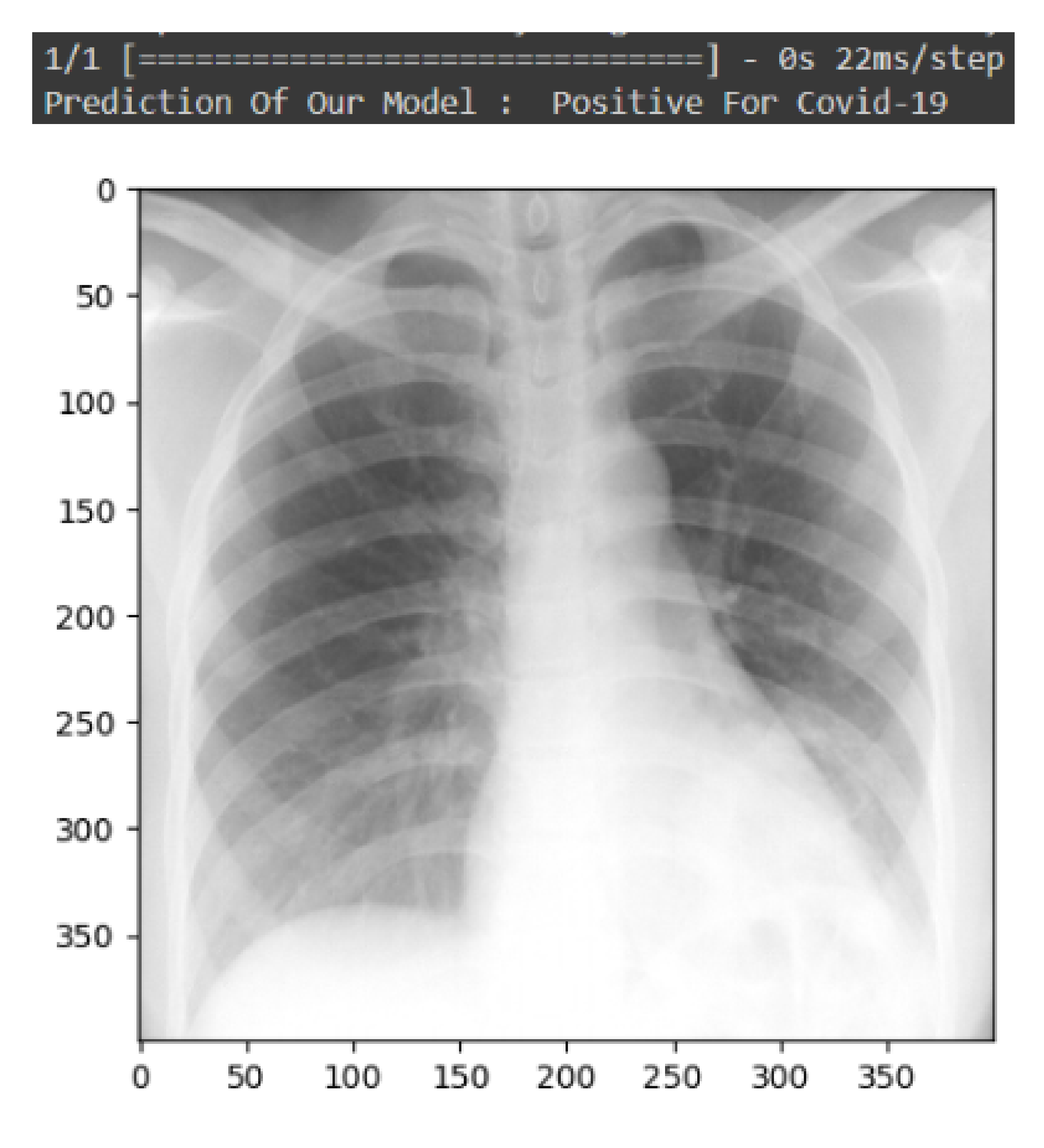
The evaluation of the CNN model on the validation dataset showed promising results, with a high accuracy rate of 0.96 and minimal overfitting since the model performed well on both training and validation data.This indicates that the model generalized well to unseen data.

Expanding the dataset with more diverse and annotated chest X-ray images can help improve the model's ability to generalize to a broader range of cases. This can include obtaining more COVID-19 positive cases as well as samples from other lung diseases.

PREDICTION FROM X-RAY IMAGES



Tested negative



Tested positive

GUI DEMO


Recycle Bin
Quick Heal Safe Banking
Google Chrome
Quick Heal Secure Br...
Adobe Acrobat
Quick Heal IS Essentials
Microsoft Edge
uTorrent Web
VLC media player
Proton VPN
Documents
This PC
21130
Microsoft Teams
Visual Studio Code
S4 B.Tech 2023 Even s...
Discord
Oracle VM VirtualBox
S4
MATLAB R2022b
Visual Studio 2022
LTspice IV
Android Studio
Cisco Packet Tracer
SCREEN RECORDER
ENG
01:58
21-07-2023

THANK YOU!