COVID-19 X-Ray Detector

Presented By: Group 33
Yesha Ajudia - AU1841078
Yashvi Pipaliya - AU1841092

TABLE OF CONTENTS

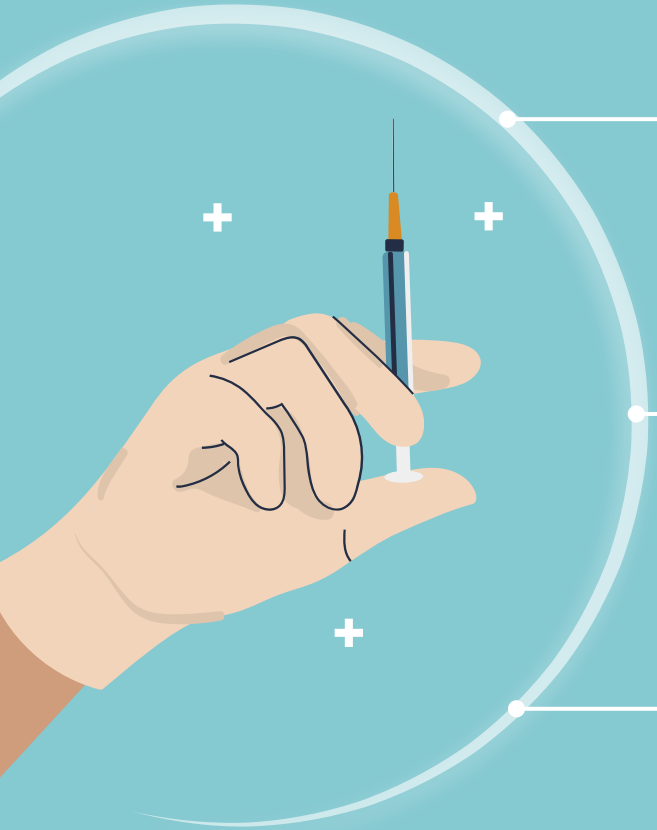
INTRODUCTION

- COVID-19 has caused a devastating effect on daily lives, public health, and the global economy.
- It is critical to detect the positive cases as early as possible to prevent the further spread of this pandemic and to quickly treat affected patients.
- The need for auxiliary diagnostic tools has increased as there are no accurate automated toolkits available.
- Recent findings obtained using radiology imaging techniques suggest that chest X-ray images contain salient information about the COVID-19 virus.
- Therefore here in this project we have created an image classifier with Tensorflow by implementing CNNs to differentiate between chest X-ray images with a COVID-19 infections versus without.

INNOVATION

- Since COVID-19 attacks the epithelial cells that line our respiratory tract, we have used X-rays to analyze the health of a patient's lungs.
- A drawback is that X-ray analysis requires a radiology expert and takes significant time which is precious when people are sick around the world.
- Therefore we have developed an automated analysis system which can detect COVID-19 using the X-ray images and will ultimately save medical professionals valuable time

TOOLS USED



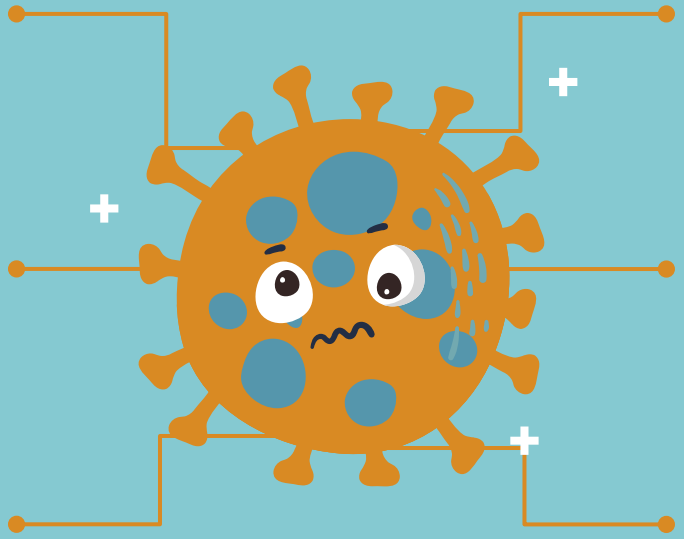
1	**Python - GOOGLE COLAB**	To write the code for the project
2	**Libraries**	• TensorFlow • Matplotlib
3	**Deep Learning**	The pre trained Convolutional Neural Networks are used to train the model

ABOUT THE DATASET

The given dataset is split into two folders, namely covid-19 and healthy.

The dataset contains various images of chest X-rays of infected and non-infected people.

The dataset is a balanced one as the number of infected and non-infected images are nearly equal.



The given dataset is not exactly clean and processed.

Some of the images have red arrows on them and some of them have additional text, like date or index.

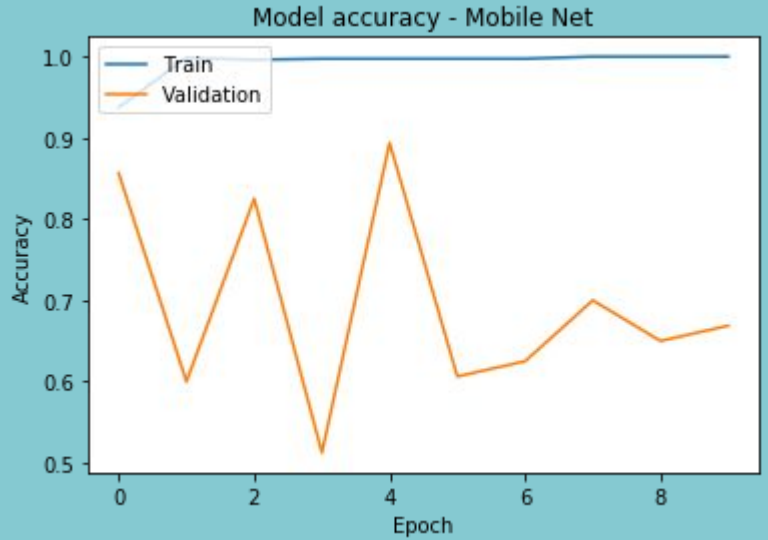
The dataset can be extended and made more accurate by adding more infected images.

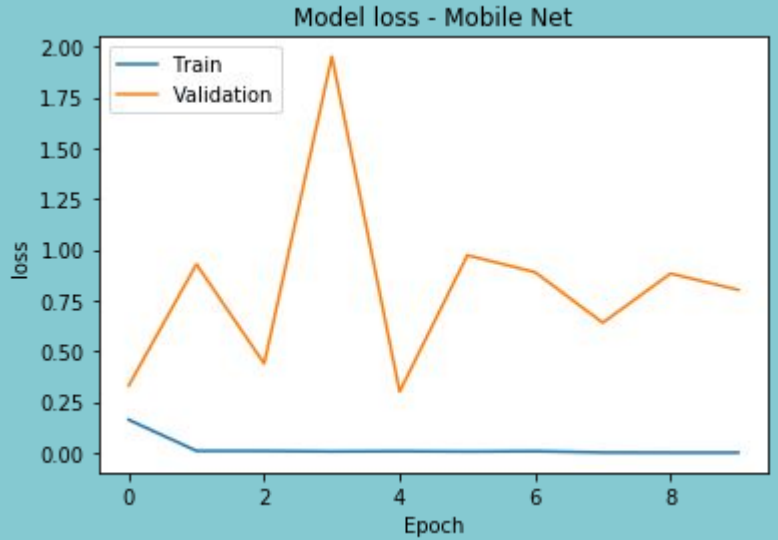
CODE SNIPPET

```
plt.plot(history_mobile.history['accuracy'])
plt.plot(history_mobile.history['val_accuracy'])
plt.title('Model accuracy - Mobile Net')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

plt.plot(history_mobile.history['loss'])
plt.plot(history_mobile.history['val_loss'])
plt.title('Model loss - Mobile Net')
plt.ylabel('loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

RESULTS


Model accuracy - Mobile Net
Train
Validation
Accuracy
Epoch
1.0
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8

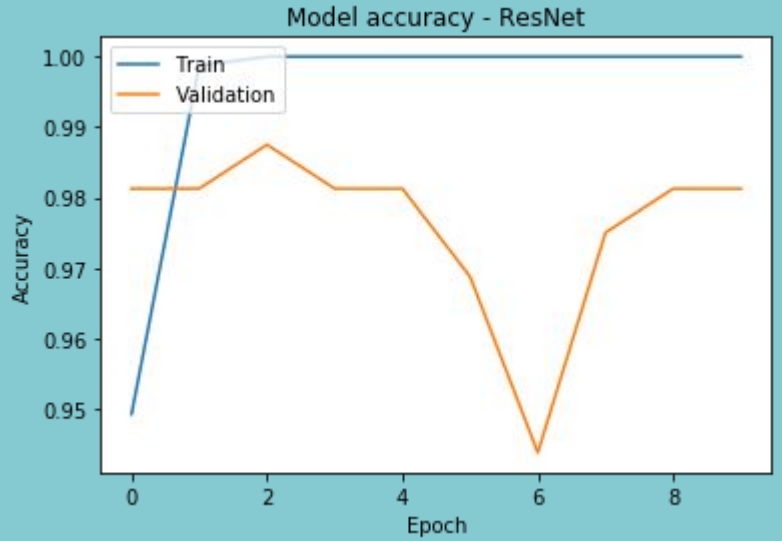

Model loss - Mobile Net
Train
Validation
loss
Epoch
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
2
4
6
8

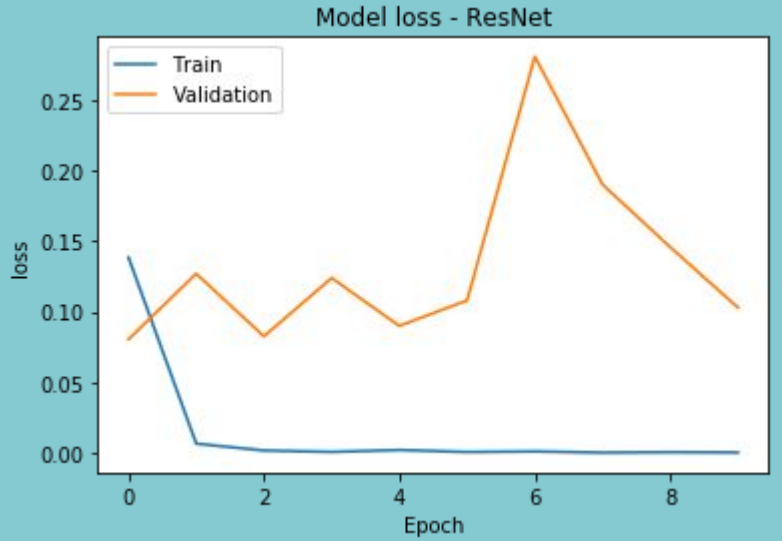
CODE SNIPPET

```
plt.plot(history_resnet.history['accuracy'])
plt.plot(history_resnet.history['val_accuracy'])
plt.title('Model accuracy - ResNet')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

plt.plot(history_resnet.history['loss'])
plt.plot(history_resnet.history['val_loss'])
plt.title('Model loss - ResNet')
plt.ylabel('loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

RESULTS


Model accuracy - ResNet
Train
Validation
Accuracy
1.00
0.99
0.98
0.97
0.96
0.95
0
2
4
6
8
Epoch

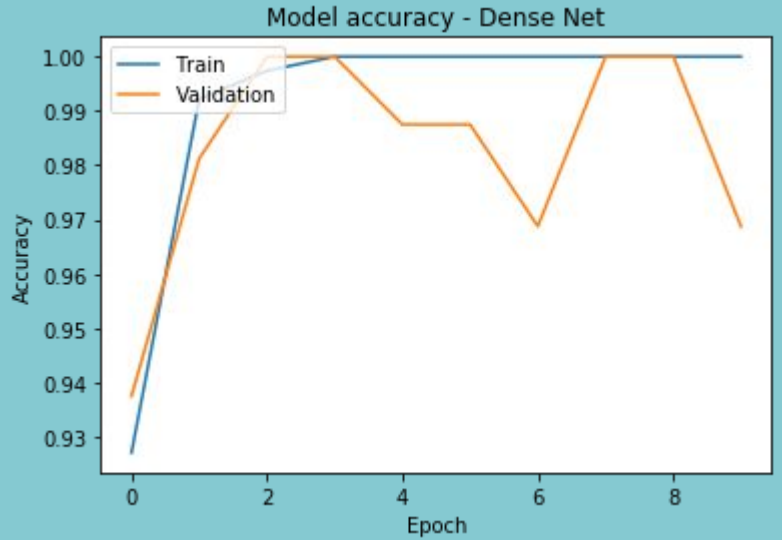

Model loss - ResNet
Train
Validation
loss
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
Epoch

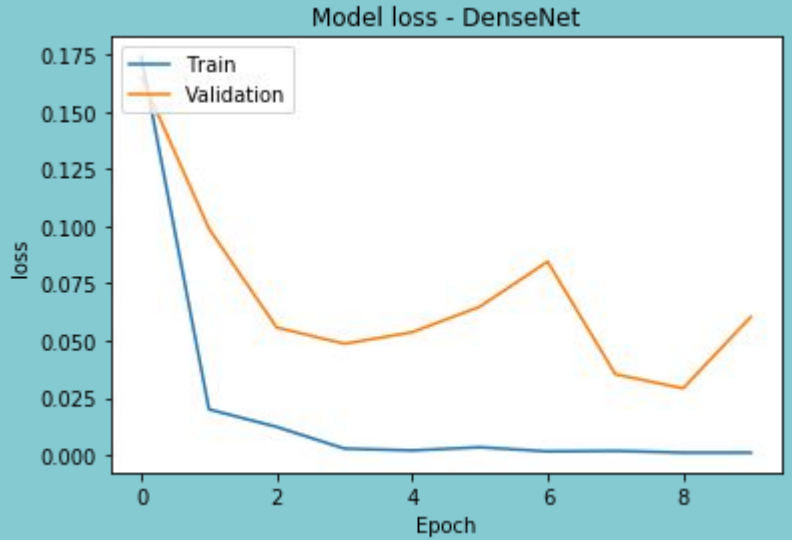
CODE SNIPPET

```
plt.plot(history_densenet.history['accuracy'])
plt.plot(history_densenet.history['val_accuracy'])
plt.title('Model accuracy - Dense Net')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

plt.plot(history_densenet.history['loss'])
plt.plot(history_densenet.history['val_loss'])
plt.title('Model loss - DenseNet')
plt.ylabel('loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

RESULTS


Model accuracy - Dense Net
Train
Validation
Accuracy
Epoch
1.00
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
2
4
6
8


Model loss - DenseNet
Train
Validation
loss
Epoch
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
2
4
6
8

LEARNING

- Learned to apply different CNNs to a large image dataset
- Understood the use of applying different layers in a network
- Got some more biological information on COVID-19 and its effect on lungs

REFERENCES

- https://rubikscode.net/2020/03/23/detection-of-covid-19-in-chest-x-rays-with-deep-learning/
- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7187882/
- https://github.com/NMZivkovic/covid-19-x-ray-detector
- https://link.springer.com/article/10.1007/s10489-020-01867-1

THANK YOU!