COVID-19 IMAGE CLASSIFIER

Presented to: Dr. Wessam El Behaidy & Eng. Salma Doma

COVER SHEET

FACULTY NAME: FACULTY OF COMPUTERS AND ARTIFICIAL INTELLIGENCE

COURSE NAME: SELECTED TOPICS IN COMPUTER SCIENCE-2

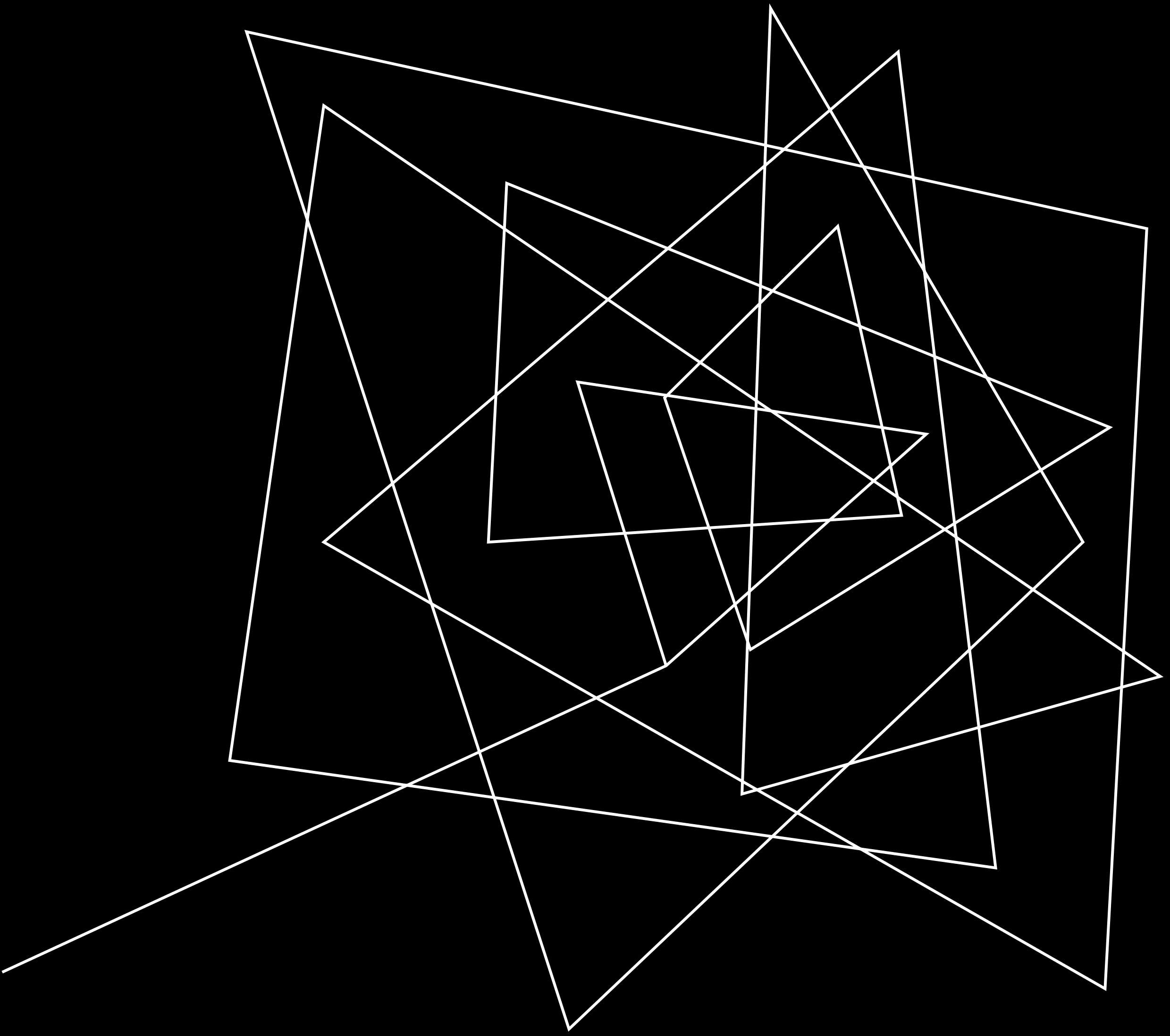
TEAM NUMBER: **14**

Name	ID	Role
1) Mohammed Aly Ebrahim Hanafy	201900710	Team Leader
2) Mohammed Emam Salama Rizq	201900640	Member
3) Omar Emad El Dosoki Ebrahim	201900521	Member
4) Abdelrehim tareq Abdelrehim Khalifa	201900445	Member
5) Ahmed Tarek Esmail Mohammed	201900047	Member
6) Hesham Ahmed Sayed Moawad	201900939	Member
7) Mahmoud Islam Mohammed Yossef	201900759	Member
8) Mohammed Shaaban Hashem Ahmed	201900689	Member

PROJECT IDEA

There is no doubt that Covid-19 is one of the most brutal diseases in 21 century, because of that 6,259,593 person die. So it’s important to help the world to git rid of this crisis, even if with a little help. So, our main goal is to implement covid-19 image classifier which can predict whether the patient has covid, viral pneumonia, or he is a normal person.

TABLE OF CONTENTS



Paper Details

PAPER CITATION

PAPER NAME

Classification Of Image Using Convolutional Neural Network(CNN)

AUTHOR NAME

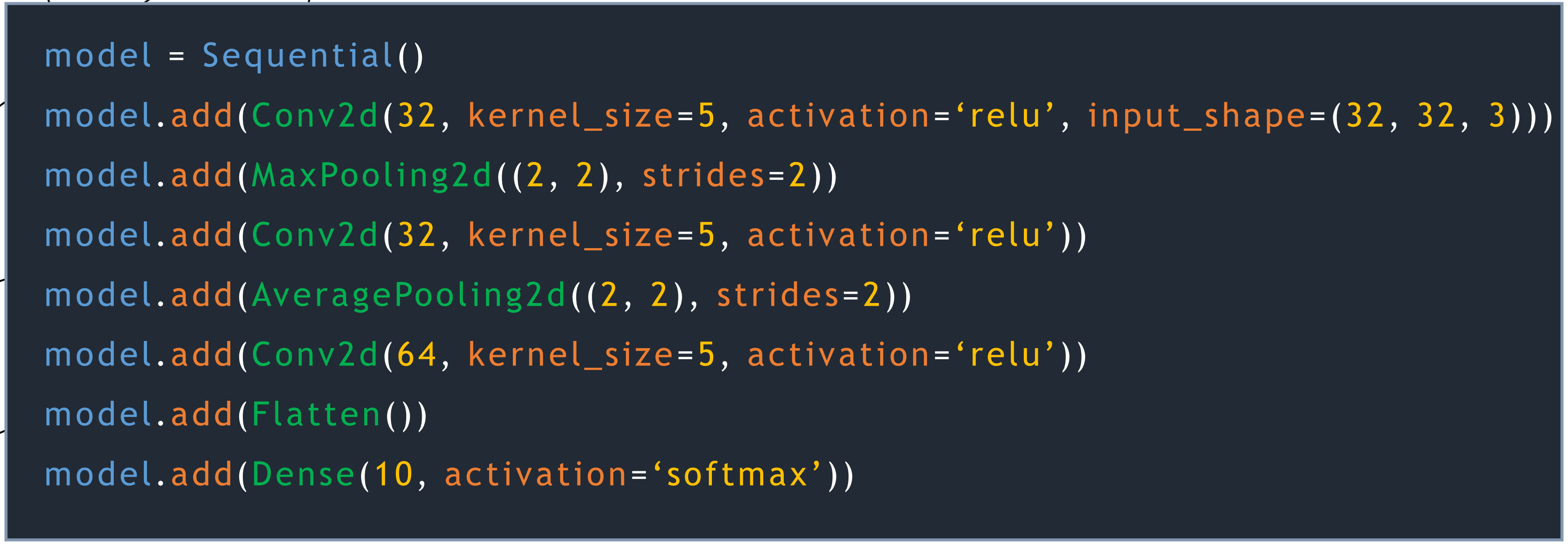
Md. Anwar Hossain & Md. Shahriar Alam Sajib

PUBLISHER NAME & YEAR OF PUBLICATION

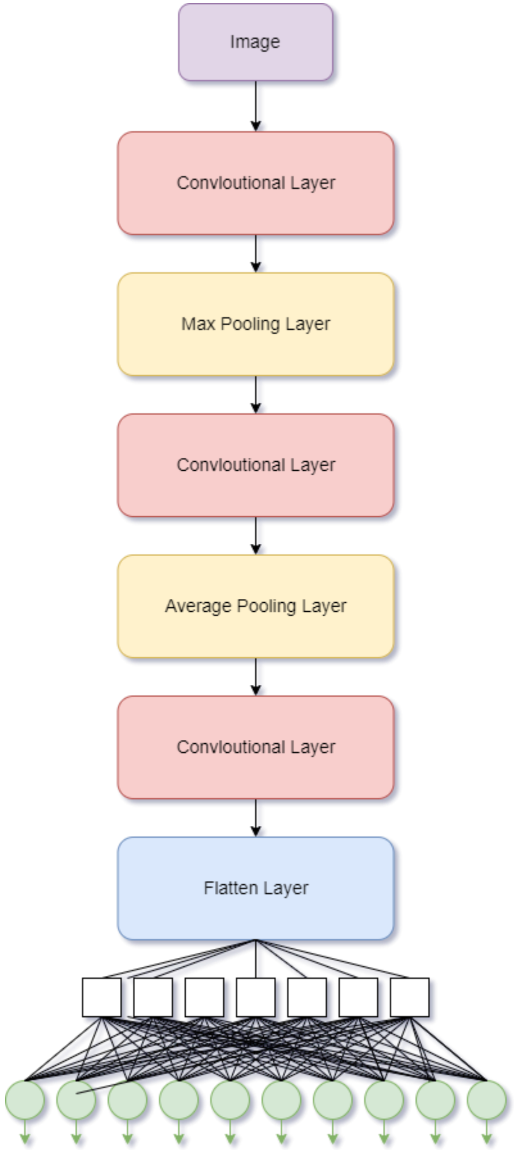
Global Journals Volume **19** Issue **2** Version **1.0** Year **2019**

PAPER DATASET

CIFAR-10 dataset



```
model = Sequential()
model.add(Conv2d(32, kernel_size=5, activation=‘relu’, input_shape=(32, 32, 3)))
model.add(MaxPooling2d((2, 2), strides=2))
model.add(Conv2d(32, kernel_size=5, activation=‘relu’))
model.add(AveragePooling2d((2, 2), strides=2))
model.add(Conv2d(64, kernel_size=5, activation=‘relu’))
model.add(Flatten())
model.add(Dense(10, activation=‘softmax’))
```



Image

Convloutional Layer

Max Pooling Layer

Convloutional Layer

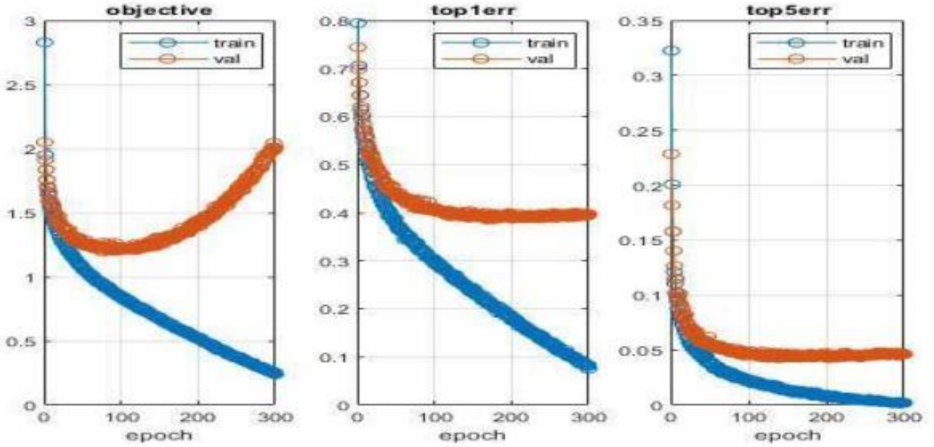
Average Pooling Layer

Convloutional Layer

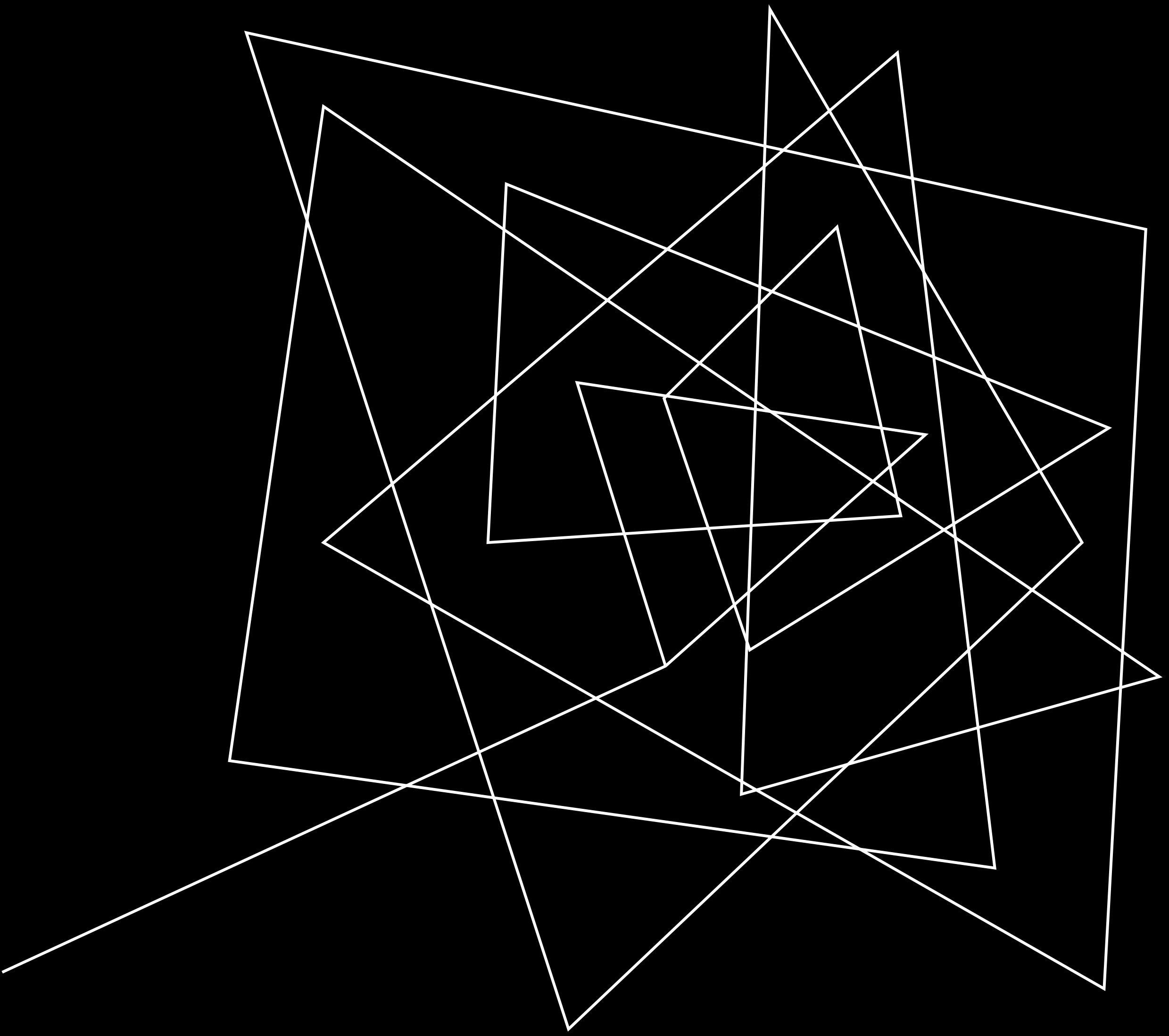
Flatten Layer

IMPLEMENTED MODEL IN PYTHON

MODEL RESULTS



Batch Size	No. Of Epochs	Testing accuracy
100	250	76.82%
70	300	82.28%
60	300	93.47%
Final Accuracy	**300**	**93.47%**



Information About Our Dataset

COVID-19 DATASET

DATASET NAME

Covid-19 Image Dataset

kaggle DATASET LINK

https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset

TOTAL NUMBER OF SAMPLES

Total number: 317

DIMENSIONS OF IMAGES

It’s not constant

NUMBER OF CLASSES AND THEIR LABELS

We have 3 classes: (Covid – Viral Pneumonia – Normal)

COVID-19 DATASET

TRAINING SET ~ (80%)

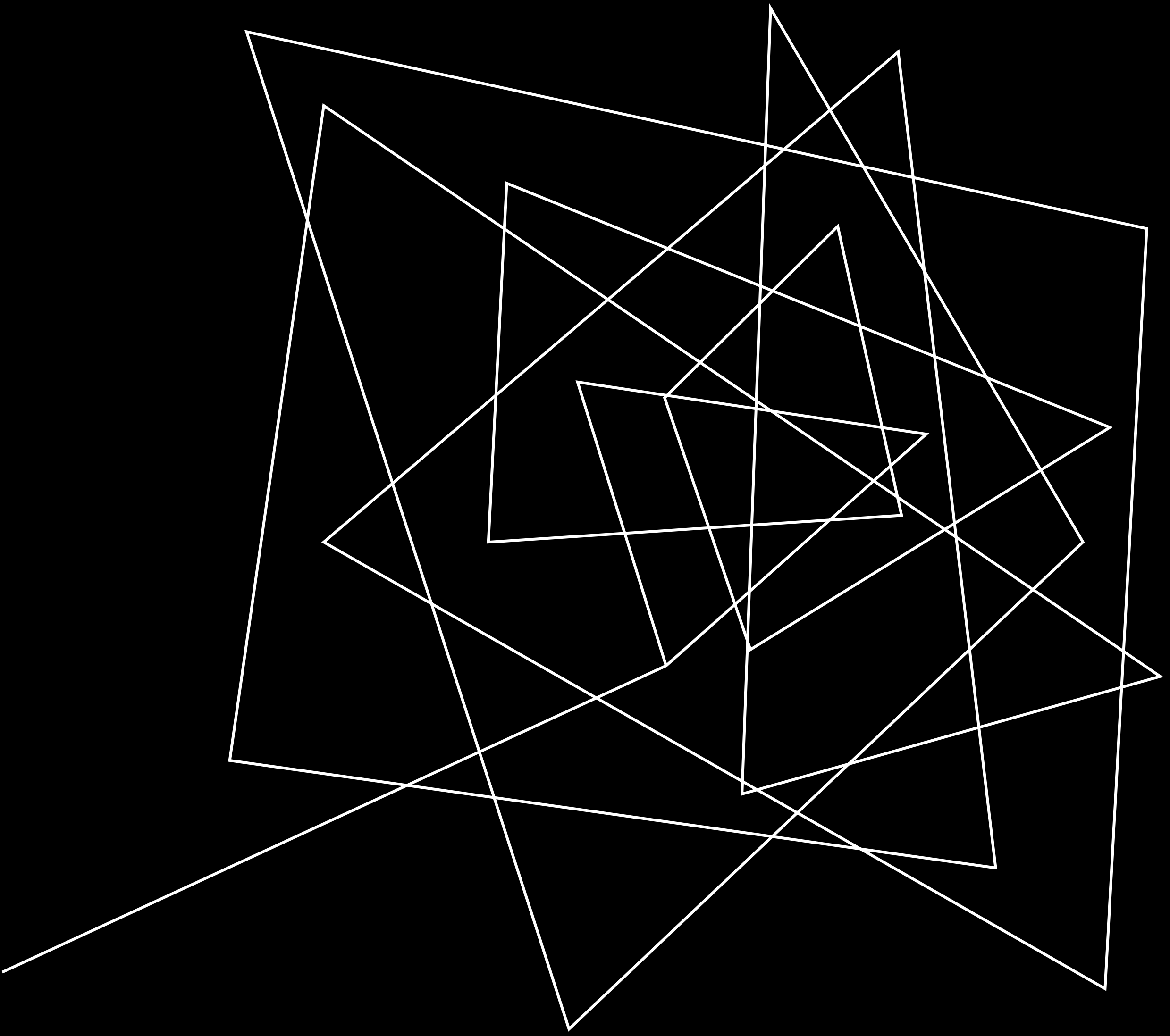
Number of images: 251

- Covid: 111
- Normal: 70
- Viral Pneumonia: 70

TESTING SET ~ (20%)

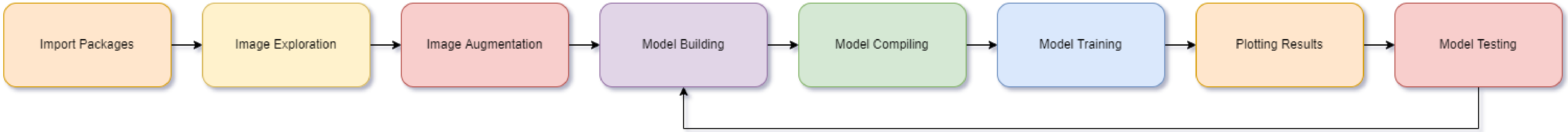
Number of images: 66

- Covid: 26
- Normal: 20
- Viral Pneumonia: 20

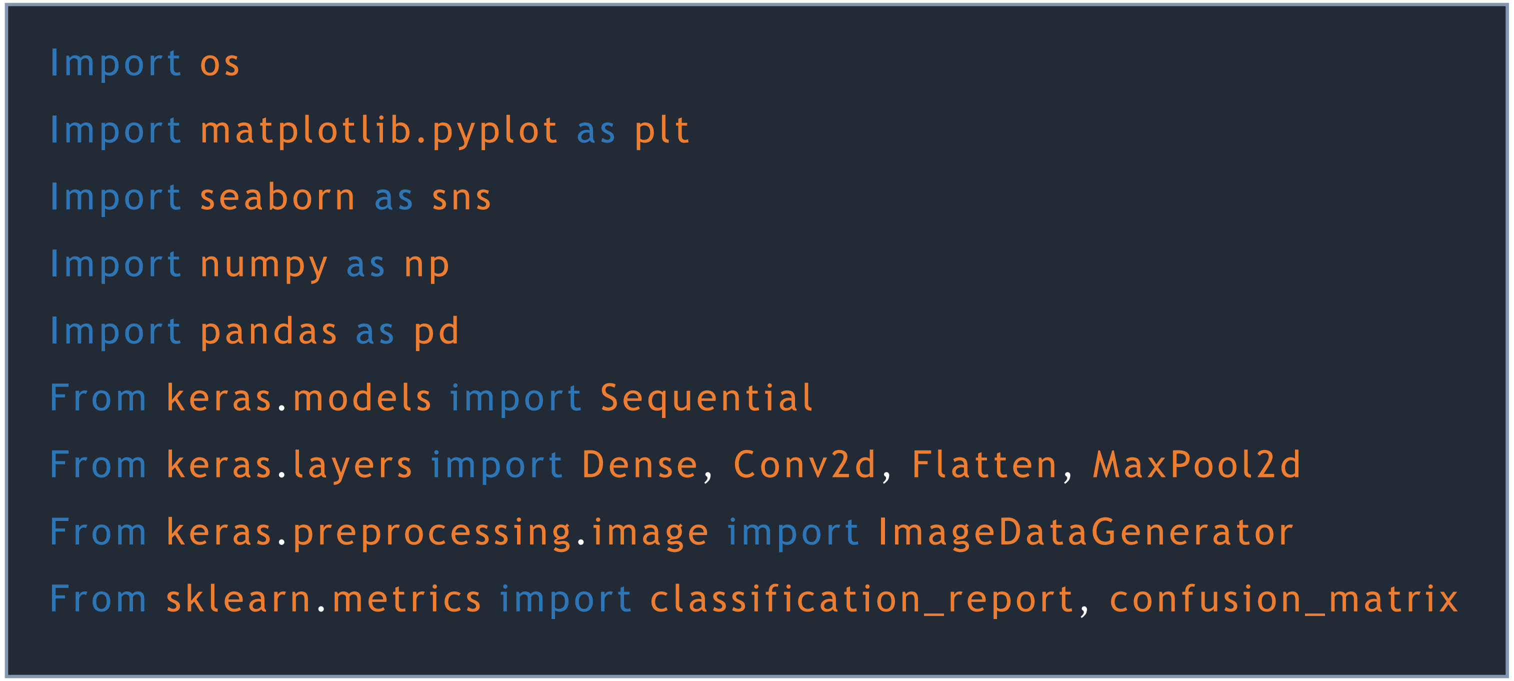


Implementation OF Our Model

PROJECT WORKFLOW



IMPORT PACKAGES

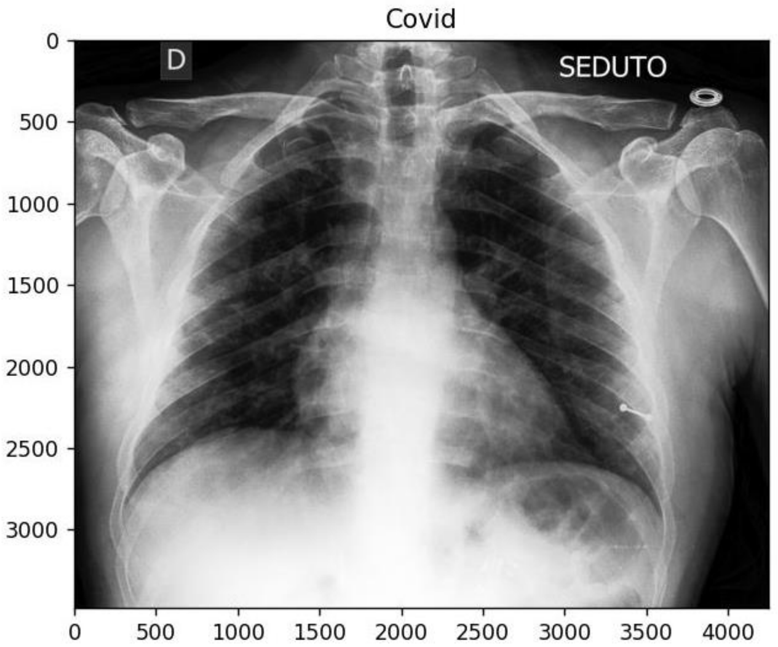


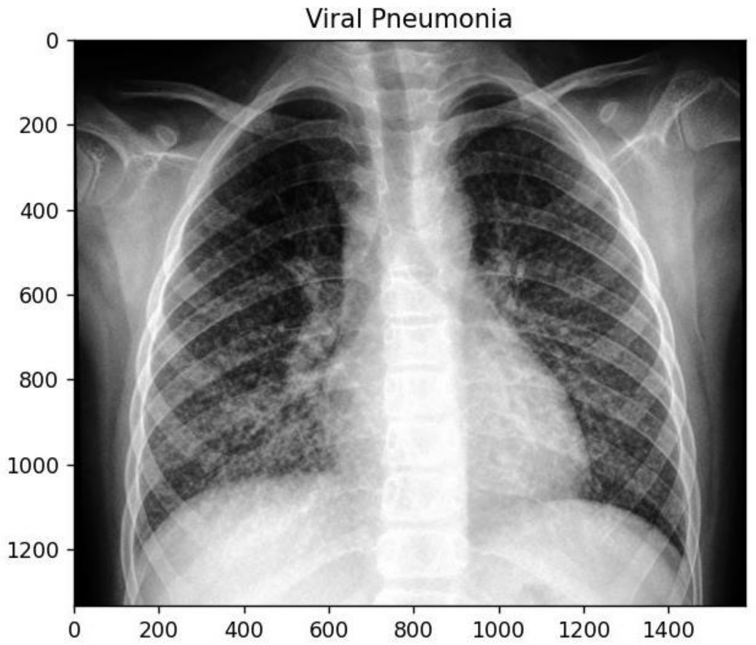
```
Import os
Import matplotlib.pyplot as plt
Import seaborn as sns
Import numpy as np
Import pandas as pd
From keras.models import Sequential
From keras.layers import Dense, Conv2d, Flatten, MaxPool2d
From keras.preprocessing.image import ImageDataGenerator
From sklearn.metrics import classification_report, confusion_matrix
```

IMAGE EXPLORATION

```
train_image_path = “Covid19-dataset/train”
test_image_path = “Covid19-dataset/test”
img = plt.imread(os.path.join(train_image_path, “Covid/01.jpeg”))
plt.imshow(img)
height, width, dim = img.shape
plt.title(‘Covid’)
print(‘size of image (h x w x d)’, height, width, dim)
plt.show()

img = plt.imread(os.path.join(train_image_path, “Viral Pneoumonia/01.jpeg”))
plt.imshow(img)
height, width, dim = img.shape
plt.title(‘Viral Pneumonia’)
print(‘size of image (h x w x d)’, height, width, dim)
plt.show()

img = plt.imread(os.path.join(train_image_path, “Normal/01.jpeg”))
plt.imshow(img)
height, width, dim = img.shape
plt.title(‘Normal’)
print(‘size of image (h x w x d)’, height, width, dim)
plt.show()
```

IMAGE EXPLORATION OUTPUT





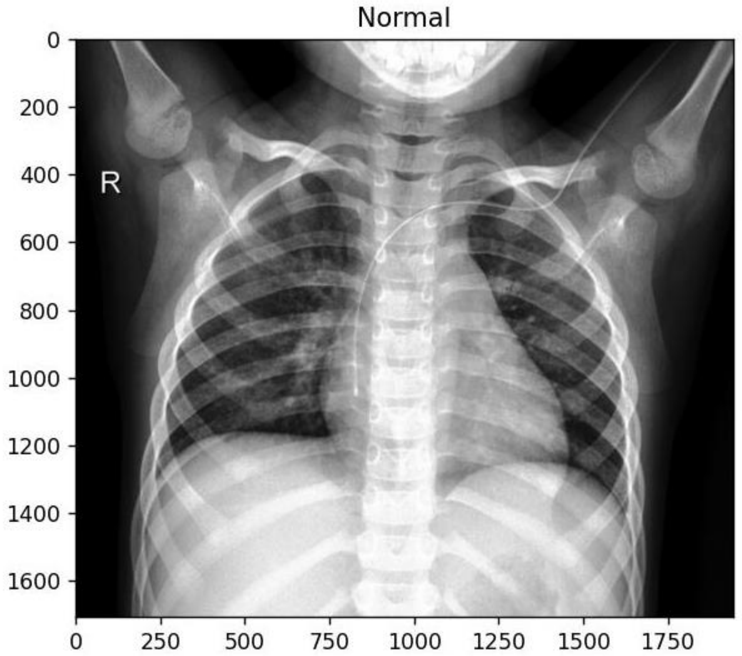
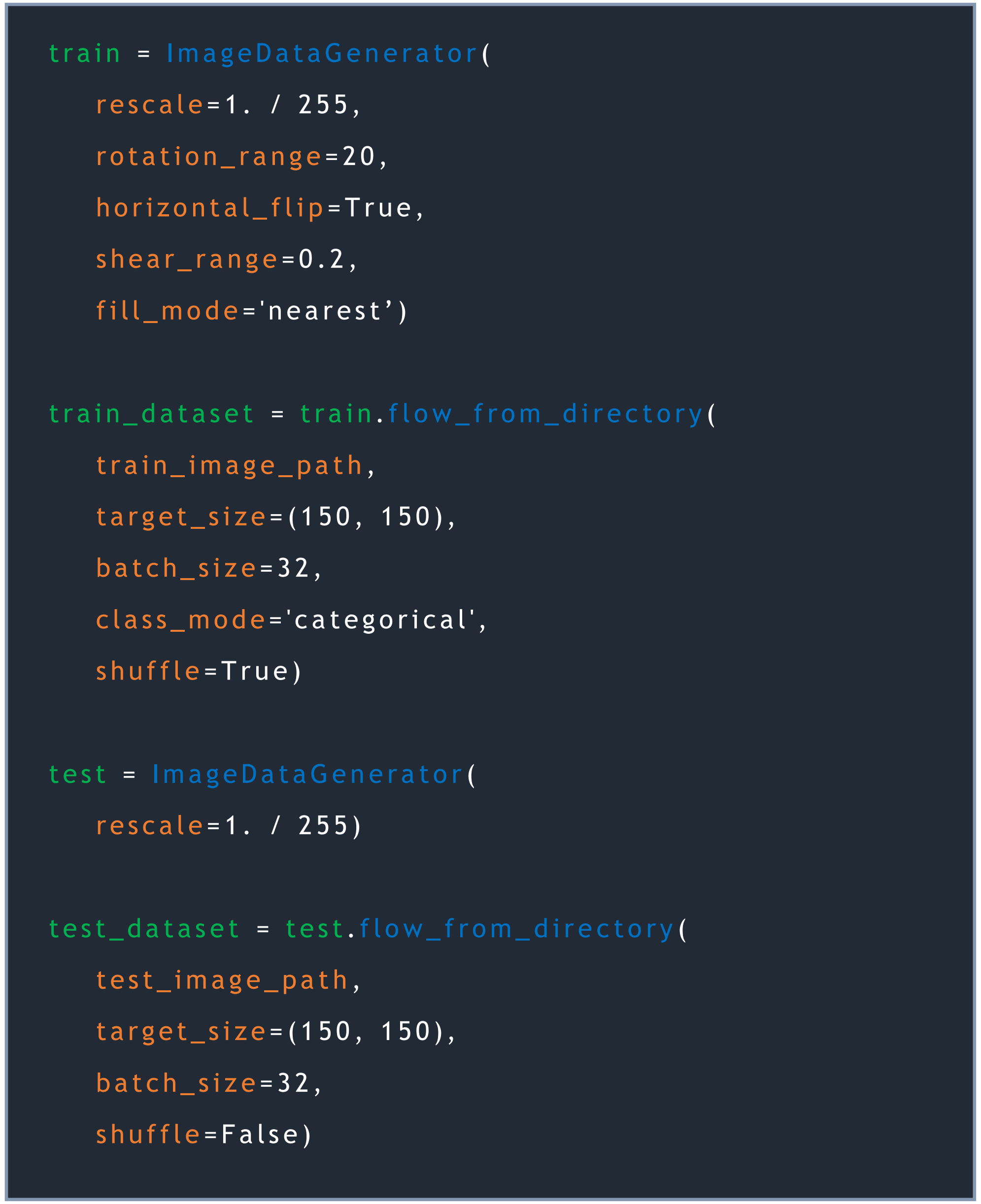
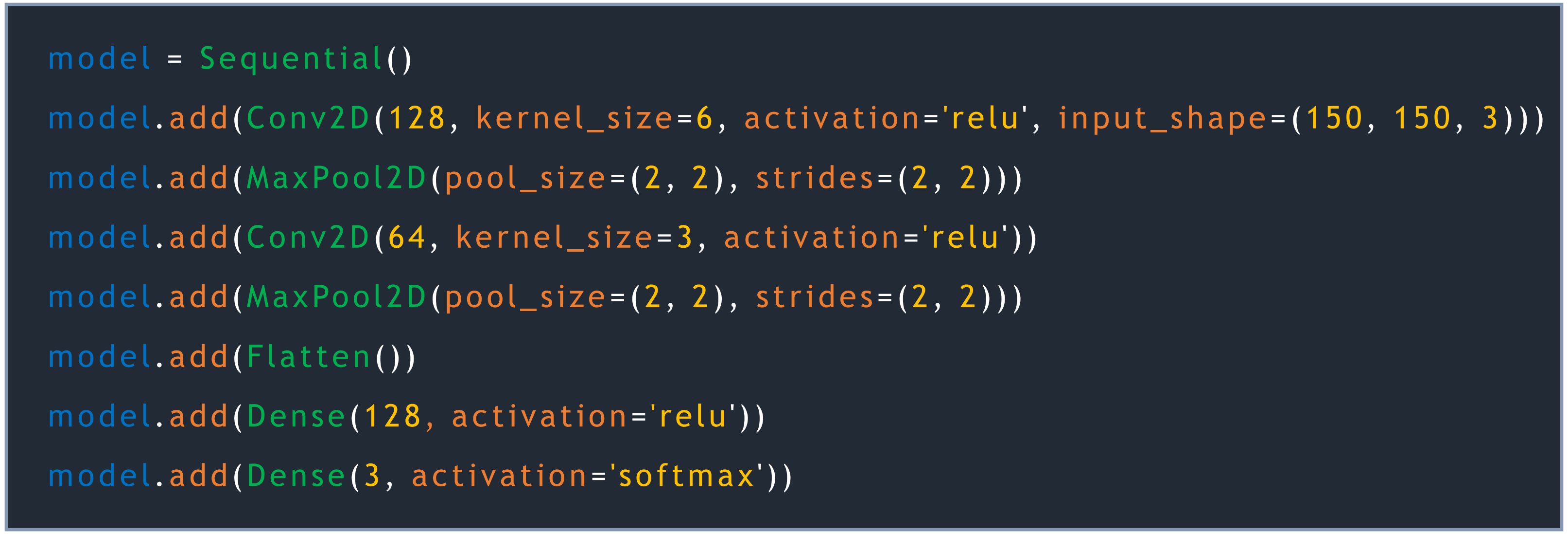


IMAGE AUGMENTATION



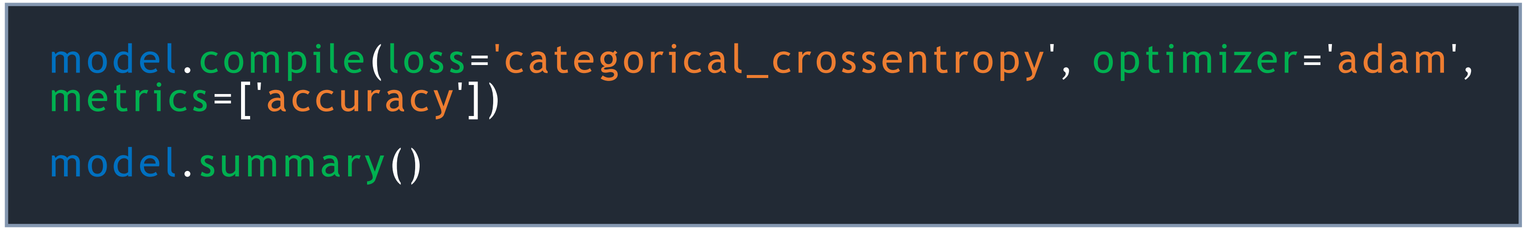
```
train = ImageDataGenerator(
    rescale=1. / 255,
    rotation_range=20,
    horizontal_flip=True,
    shear_range=0.2,
    fill_mode='nearest’)

train_dataset = train.flow_from_directory(
    train_image_path,
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical',
    shuffle=True)

test = ImageDataGenerator(
    rescale=1. / 255)

test_dataset = test.flow_from_directory(
    test_image_path,
    target_size=(150, 150),
    batch_size=32,
    shuffle=False)
```



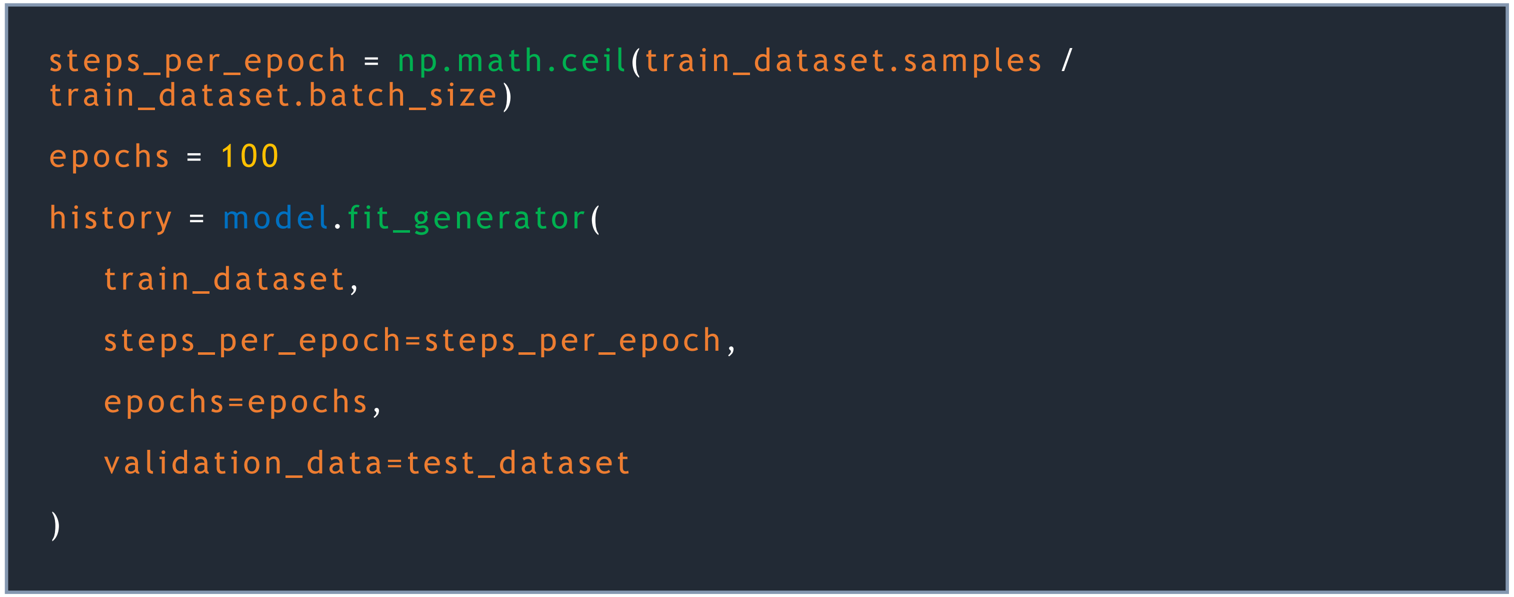
```
model = Sequential()
model.add(Conv2D(128, kernel_size=6, activation='relu', input_shape=(150, 150, 3)))
model.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(3, activation='softmax'))
```

MODEL BUILDING

MODEL COMPILING

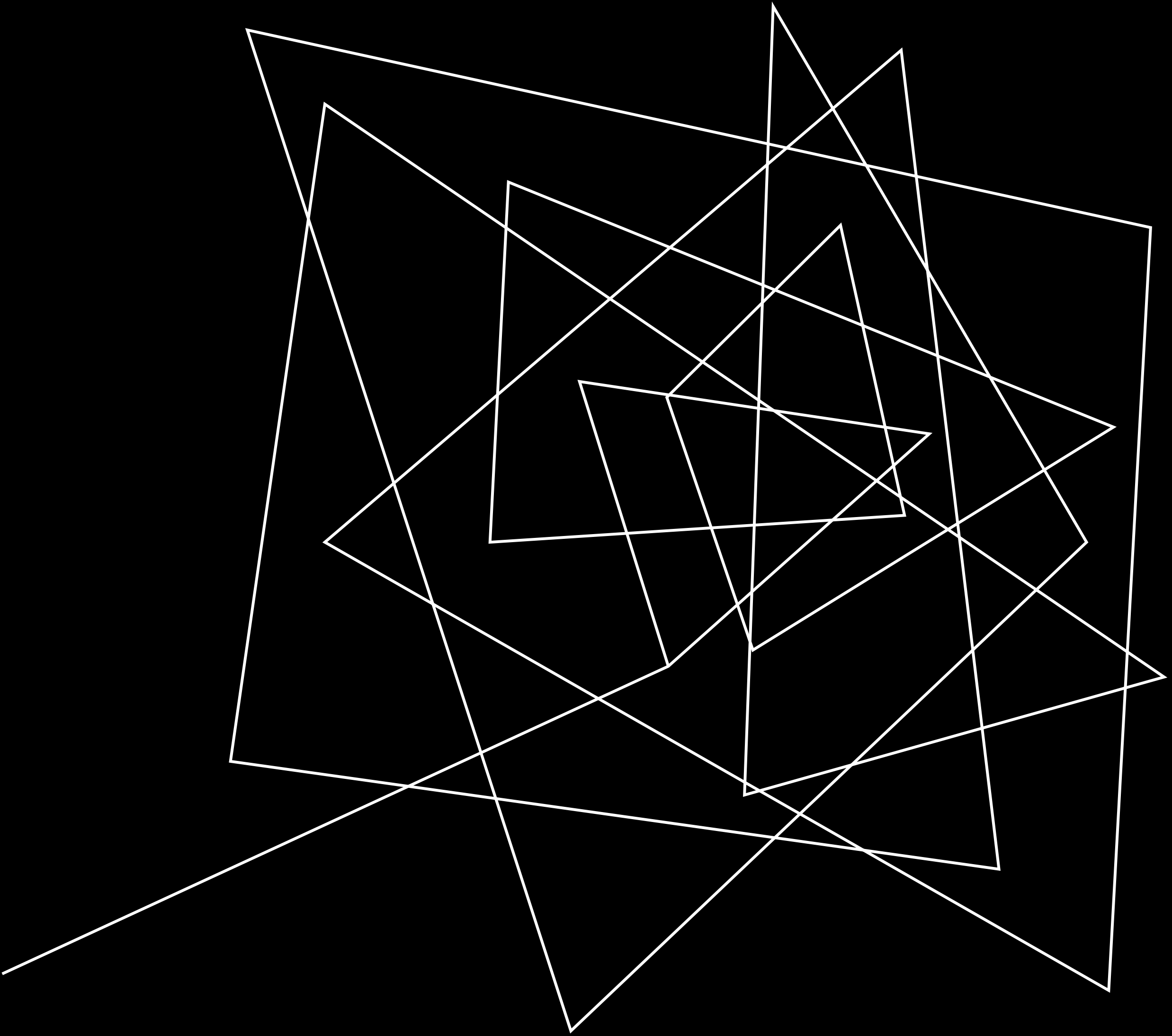


```
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

model.summary()
```

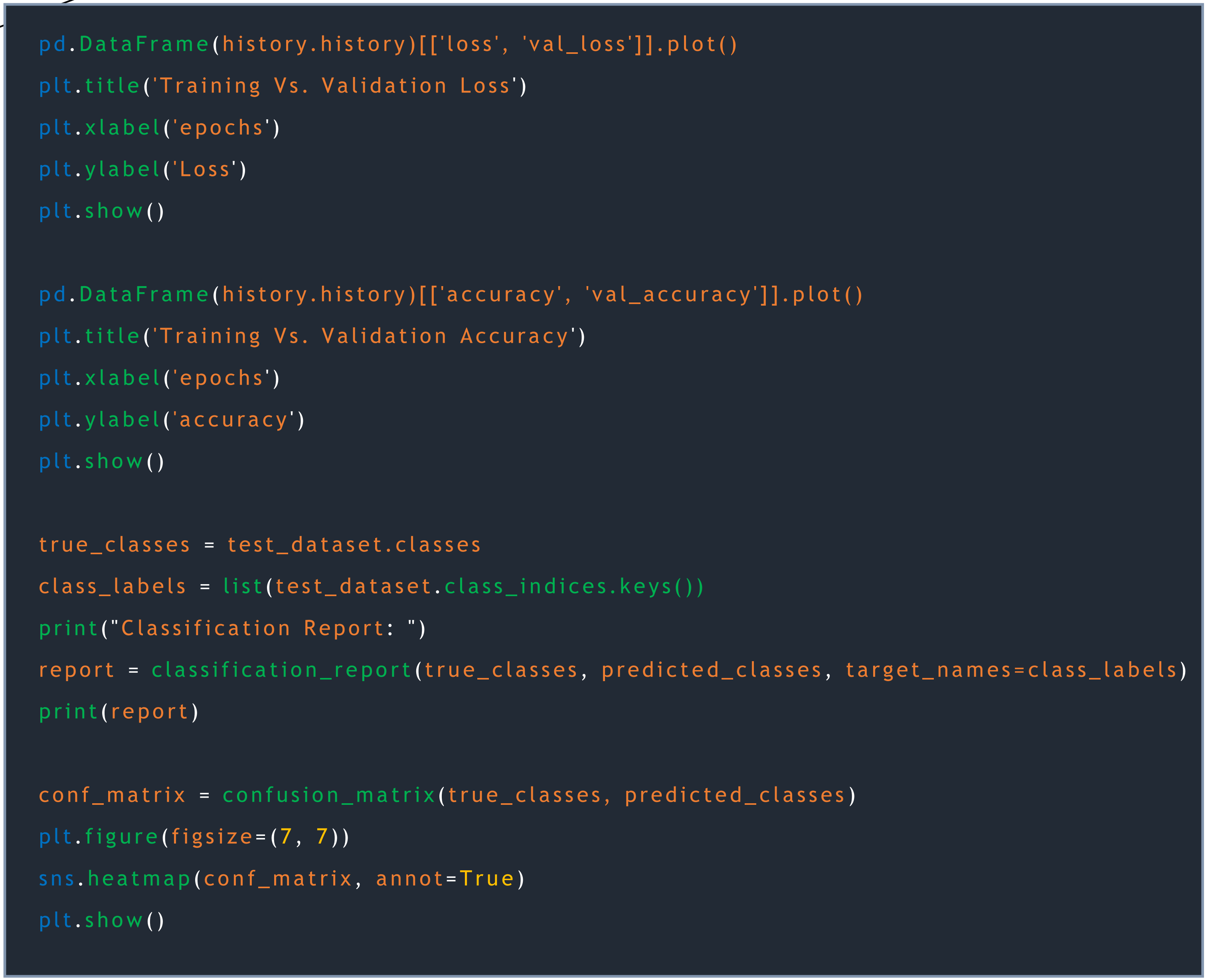


```
steps_per_epoch = np.math.ceil(train_dataset.samples /
train_dataset.batch_size)

epochs = 100

history = model.fit_generator(

    train_dataset,

    steps_per_epoch=steps_per_epoch,

    epochs=epochs,

    validation_data=test_dataset

)
```

MODEL TRAINING(FITTING)



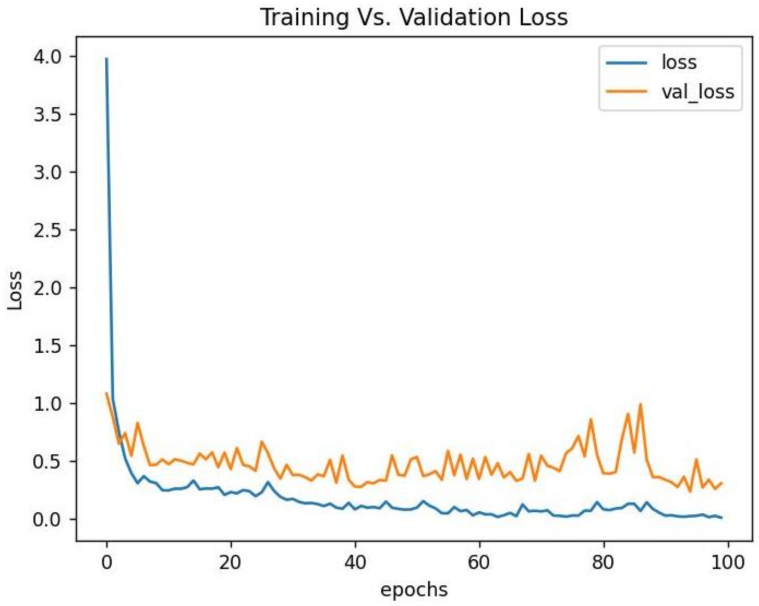
Model Testing
AND Plot Results

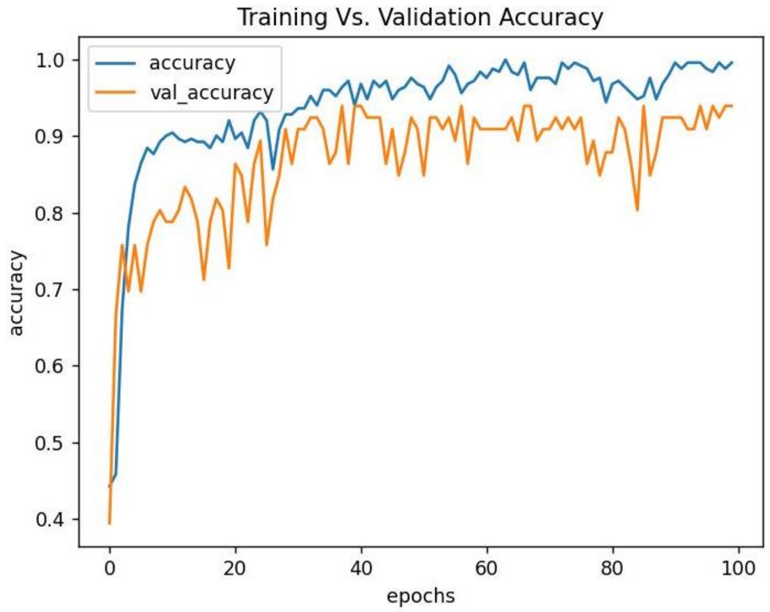


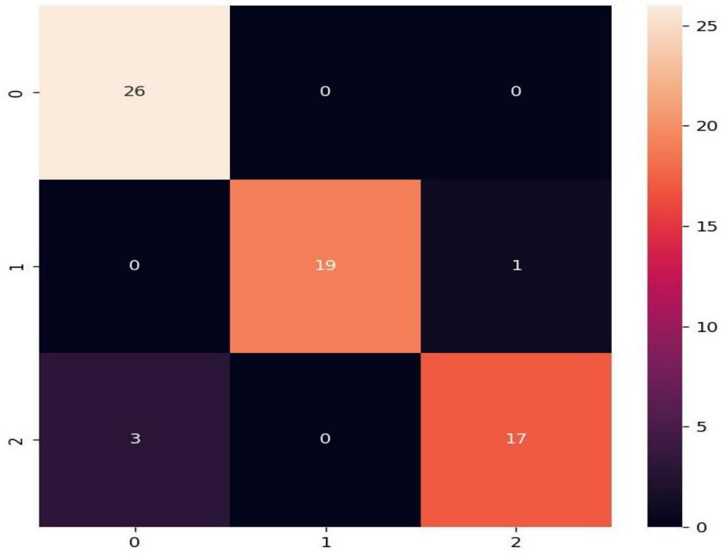
```
pd.DataFrame(history.history)[['loss', 'val_loss']].plot()
plt.title('Training Vs. Validation Loss')
plt.xlabel('epochs')
plt.ylabel('Loss')
plt.show()

pd.DataFrame(history.history)[['accuracy', 'val_accuracy']].plot()
plt.title('Training Vs. Validation Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.show()

true_classes = test_dataset.classes
class_labels = list(test_dataset.class_indices.keys())
print("Classification Report: ")
report = classification_report(true_classes, predicted_classes, target_names=class_labels)
print(report)

conf_matrix = confusion_matrix(true_classes, predicted_classes)
plt.figure(figsize=(7, 7))
sns.heatmap(conf_matrix, annot=True)
plt.show()
```

PLOTTING RESULTS

PLOTTING RESULTS OUTPUT

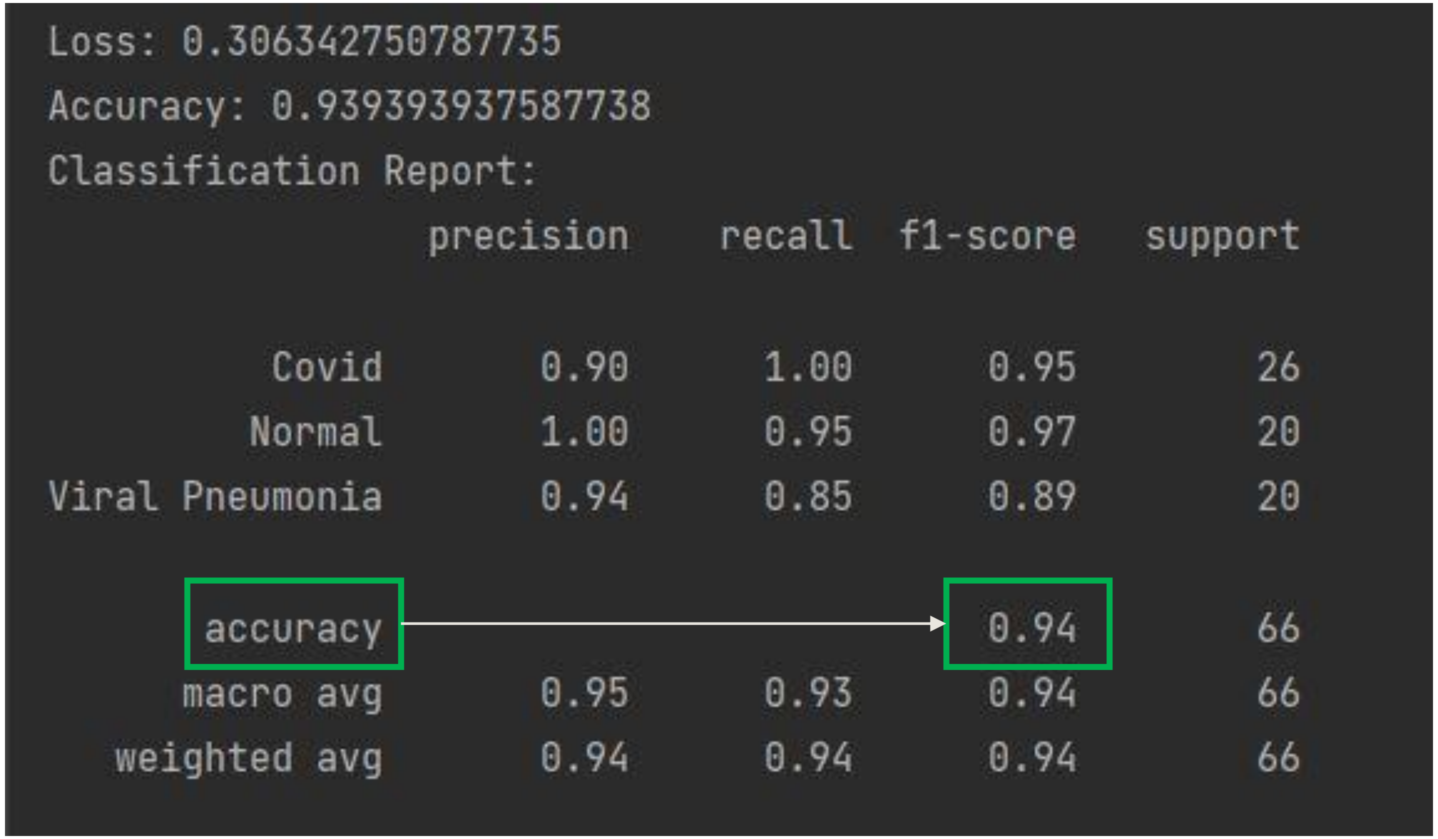

Training Vs. Validation Loss


Training Vs. Validation Accuracy



PLOTTING RESULTS OUTPUT

CLASSIFICATION REPORT



```
Loss: 0.306342750787735
Accuracy: 0.939393937587738
Classification Report:
                 precision    recall  f1-score   support

          Covid       0.90      1.00      0.95        26
         Normal       1.00      0.95      0.97        20
Viral Pneumonia       0.94      0.85      0.89        20

       accuracy                           0.94        66
      macro avg       0.95      0.93      0.94        66
   weighted avg       0.94      0.94      0.94        66
```

```
Import cv2 as cv
classes = ["Covid", "Normal", "Viral Pneumonia"]
model = load_model('FinalCovidClassifier.h5')

def prepareImage(imagePath):
    image_size = 150
    img_array = cv.imread(imagePath)
    img_array = cv.cvtColor(img_array, cv.COLOR_BGR2RGB)
    new_array = cv.resize(img_array, (image_size, image_size))
    plt.imshow(new_array)
    return new_array.reshape(-1, image_size, image_size, 3)

image = prepareImage("F:/normal.jpg")
prediction = model.predict(image)
index = np.argmax(prediction)
print(f"Prediction is {classes[index]}")
plt.show()
```

MODEL TESTING

THANK YOU

GitHub Link: https://github.com/MohammedAly22/Covid-19_Image_Classification