Model Development Phase

Date	12th July 2024
Team ID	SWTID1720195938
Project Title	**CovidVision: Advanced COVID-19 Detection from Lung X-Rays with Deep Learning**
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

Initial Model Training Code (5 marks):

Paste the screenshot of the model training code

```
import pathlib
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import random
import shutil
import os
import cv2
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import Xception
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Activation, Conv2D, MaxPool2D, Flatten, Dropout, BatchNormalization
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report,confusion_matrix
from sklearn.model_selection import train_test_split
from google.colab import files
import seaborn as sns
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.Kaggle
```

```
[ ] !kaggle datasets download -d tawsifurrahman/covid19-radiography-database
```

```
Dataset URL: https://www.kaggle.com/datasets/tawsifurrahman/covid19-radiography-database
License(s): copyright-authors
Downloading covid19-radiography-database.zip to /content
100% 778M/778M [00:29<00:00, 38.4MB/s]
100% 778M/778M [00:29<00:00, 27.8MB/s]
```

```
[ ] !unzip '/content/covid19-radiography-database.zip'
```

```
[ ] PATH_TO_METADATA = "/content/COVID-19_Radiography_Dataset/Normal.metadata.xlsx"
    df = pd.read_excel(PATH_TO_METADATA)
    df.head()
```

Show hidden output

```
[ ] !ls /tmp
```

Show hidden output

```
[ ] !mkdir /tmp/Xray_train_data
```

```
[ ] !cp -R "/content/COVID-19_Radiography_Dataset/COVID/images" "/tmp/Xray_train_data/"
```

```
[ ] !mv "/tmp/Xray_train_data/images" "/tmp/Xray_train_data/COVID"
```

```
[ ] !ls -1 "/tmp/Xray_train_data/COVID"| wc -l
```

Show hidden output

```
[ ] !mkdir "/tmp/Xray_train_data/NORMAL"
```

```
[ ] cnt = 0
    for (i, row) in df.iterrows():
        if (cnt < 3616):
            filename = row["FILE NAME"].lower().capitalize() + "." + row["FORMAT"].lower()
            image_path = os.path.join("/content/COVID-19_Radiography_Dataset/Normal/images", filename)
            image_copy_path = os.path.join("/tmp/Xray_train_data/NORMAL", filename)
            shutil.copy2(image_path, image_copy_path)
            cnt += 1

    print(cnt)
```

```
[ ] !ls "/tmp/Xray_train_data"
```

Show hidden output

```
[ ] !ls -1 "/tmp/Xray_train_data/COVID"| wc -l
```

Show hidden output

```
!ls -1 "/tmp/Xray_train_data/NORMAL"| wc -l
```

Show hidden output

```
[ ] train_data_dir= "/tmp/Xray_train_data"

[ ] IMAGE_SIZE = (299, 299)
    IMAGE_SHAPE = IMAGE_SIZE + (3,)
```

```
img_height, img_width=IMAGE_SIZE
batch_size=16
train_datagen = ImageDataGenerator(validation_split=0.3)

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='binary',
    subset='training')
validation_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='binary',
    subset='training')
```

```
xception_model = Sequential()
pre_model = tf.keras.applications.Xception(
    include_top=False,
    input_shape=(299, 299, 3),
    pooling='avg',
    weights='imagenet'
)
for layer in pre_model.layers:
    layer.trainable = False
xception_model.add(pre_model)
xception_model.add(Flatten())
xception_model.add(Dense(512, activation='relu'))
xception_model.add(Dense(1, activation='sigmoid'))
```

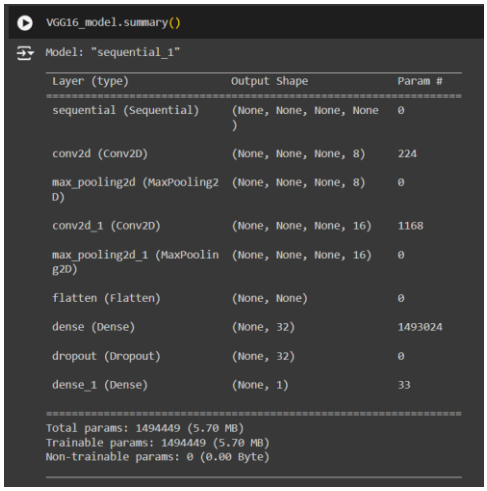
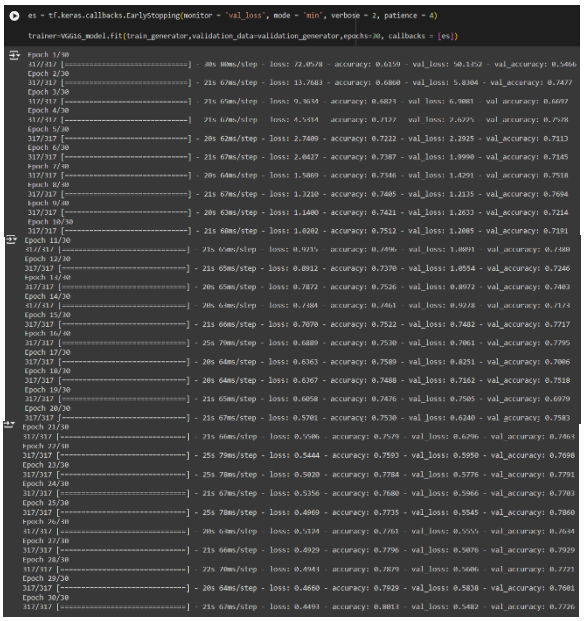
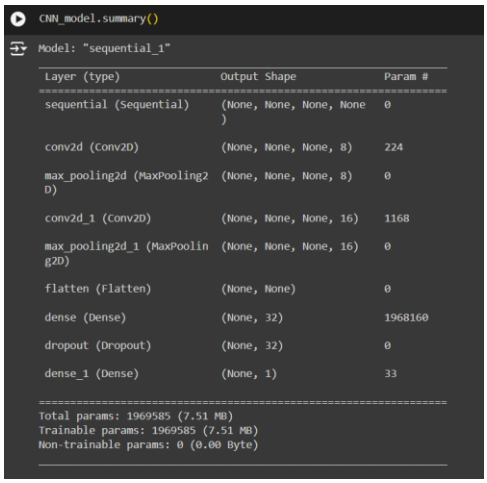
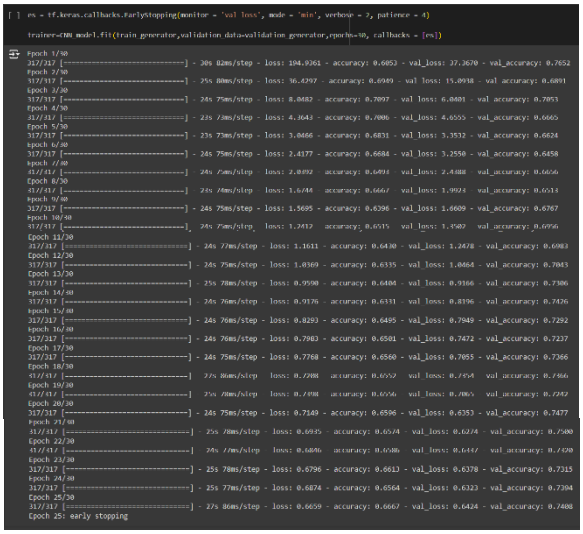
Show hidden output

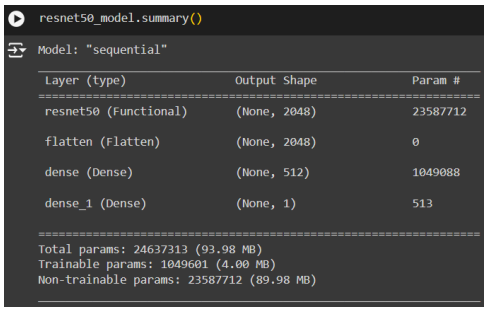
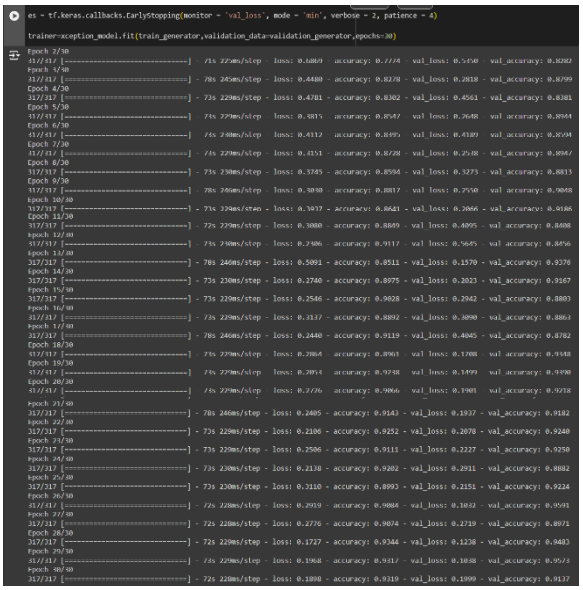
```
[ ] xception_model.compile(optimizer=Adam(learning_rate=0.0001),
                           loss='binary_crossentropy',
                           metrics = ['accuracy'])
```

```
[ ] es = tf.keras.callbacks.EarlyStopping(monitor = 'val_loss', mode = 'min', verbose = 2, patience = 4)

    trainer=xception_model.fit(train_generator,validation_data=validation_generator,epochs=30)
```

Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics
VGG16		
CNN Model		

RESNET 50		
Xception