```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import numpy as np
import pandas as pd


import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/dataset'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Show hidden output

```
import keras
from keras.models import Sequential
from keras.layers import Conv2D,Flatten,Dense,MaxPooling2D,Dropout
from sklearn.metrics import accuracy_score
```

```
import ipywidgets as widgets
import io
from PIL import Image
import tqdm
from sklearn.model_selection import train_test_split
import cv2
from sklearn.utils import shuffle
import tensorflow as tf
```

```


X_train = []
Y_train = []
image_size = 150
labels = ['COVID','Normal']
for i in labels:
    folderPath = os.path.join('/content/drive/MyDrive/dataset/train',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)

for i in labels:
    folderPath = os.path.join('/content/drive/MyDrive/dataset/val',i)
    for j in os.listdir(folderPath):
        img = cv2.imread(os.path.join(folderPath,j))
        img = cv2.resize(img,(image_size,image_size))
        X_train.append(img)
        Y_train.append(i)

X_train = np.array(X_train)
Y_train = np.array(Y_train)
```

```
X_train,Y_train = shuffle(X_train,Y_train,random_state=101)
X_train.shape
```

```
(8264, 150, 150, 3)
```

```
X_train,X_test,y_train,y_test = train_test_split(X_train,Y_train,test_size=0.1,random_state=101)
```

```
y_train_new = []
for i in y_train:
    y_train_new.append(labels.index(i))
y_train=y_train_new
y_train = tf.keras.utils.to_categorical(y_train)

y_test_new = []
for i in y_test:
    y_test_new.append(labels.index(i))
y_test=y_test_new
y_test = tf.keras.utils.to_categorical(y_test)
```

```
model = Sequential()
model.add(Conv2D(32,(3,3),activation = 'relu',input_shape=(150,150,3)))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(Dropout(0.3))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Conv2D(128,(3,3),activation='relu'))
model.add(Conv2D(256,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
model.add(Dropout(0.3))
model.add(Flatten())
model.add(Dense(512,activation = 'relu'))
model.add(Dense(512,activation = 'relu'))
model.add(Dropout(0.3))
model.add(Dense(2,activation='softmax'))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
conv2d_1 (Conv2D)	(None, 146, 146, 64)	18,496
max_pooling2d (MaxPooling2D)	(None, 73, 73, 64)	0
dropout (Dropout)	(None, 73, 73, 64)	0
conv2d_2 (Conv2D)	(None, 71, 71, 64)	36,928
conv2d_3 (Conv2D)	(None, 69, 69, 64)	36,928
dropout_1 (Dropout)	(None, 69, 69, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 34, 34, 64)	0
dropout_2 (Dropout)	(None, 34, 34, 64)	0
conv2d_4 (Conv2D)	(None, 32, 32, 128)	73,856
conv2d_5 (Conv2D)	(None, 30, 30, 128)	147,584
conv2d_6 (Conv2D)	(None, 28, 28, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
dropout_3 (Dropout)	(None, 14, 14, 128)	0
conv2d_7 (Conv2D)	(None, 12, 12, 128)	147,584
conv2d_8 (Conv2D)	(None, 10, 10, 256)	295,168
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 256)	0
dropout_4 (Dropout)	(None, 5, 5, 256)	0
flatten (Flatten)	(None, 6400)	0
dense (Dense)	(None, 512)	3,277,312
dense_1 (Dense)	(None, 512)	262,656
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 2)	1,026

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

```
history = model.fit(X_train,y_train,epochs=50,validation_split=0.3,batch_size=32)
```

```
Epoch 1/50
163/163 ━━━━━━━━━━━━━━━━━━━━ 1246s 8s/step - accuracy: 0.5433 - loss: 13.2476 - val_accuracy: 0.6927 - val_loss: 0.6717
Epoch 2/50
```

```
163/163 ━━━━━━━━━━━━━━━━━━━━ 1228s 7s/step - accuracy: 0.6778 - loss: 0.6048 - val_accuracy: 0.7240 - val_loss: 0.6039
Epoch 3/50
163/163 ━━━━━━━━━━━━━━━━━━━━ 0s 7s/step - accuracy: 0.6911 - loss: 0.5720
```

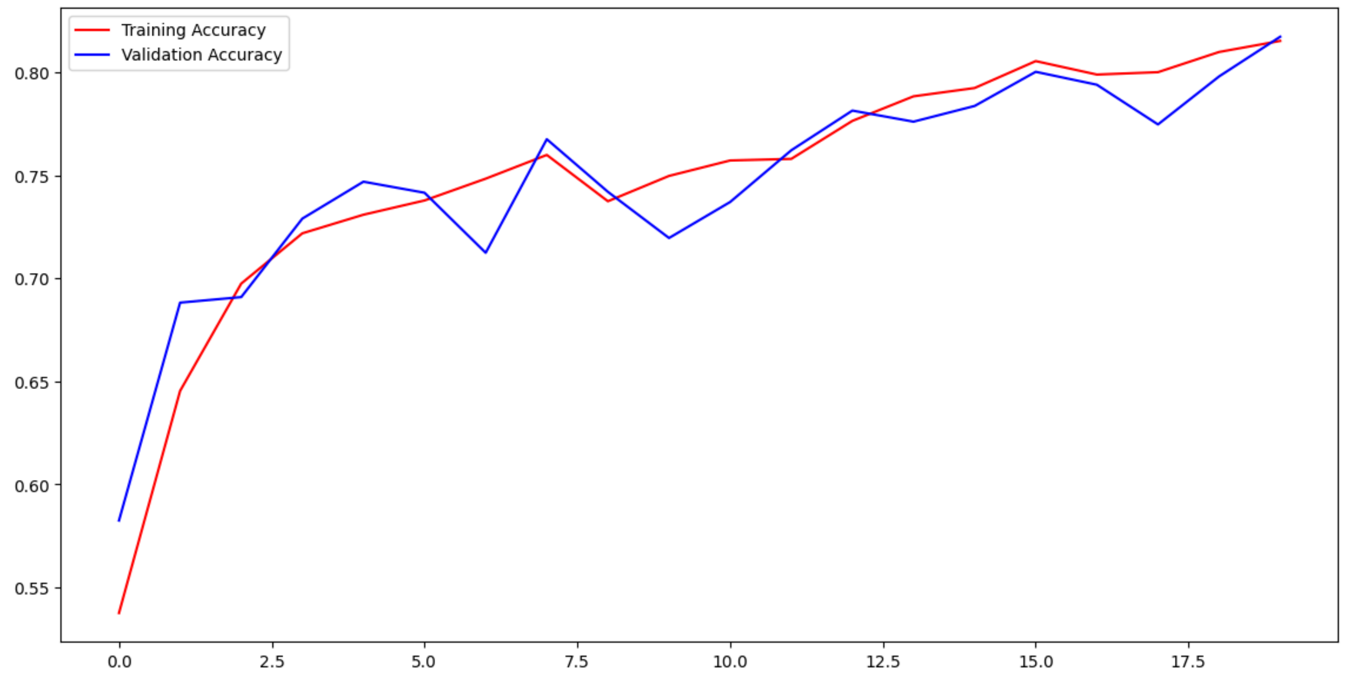
Double-click (or enter) to edit

```
model.save('/content/drive/MyDrive/New_Sequential_3.keras')
```

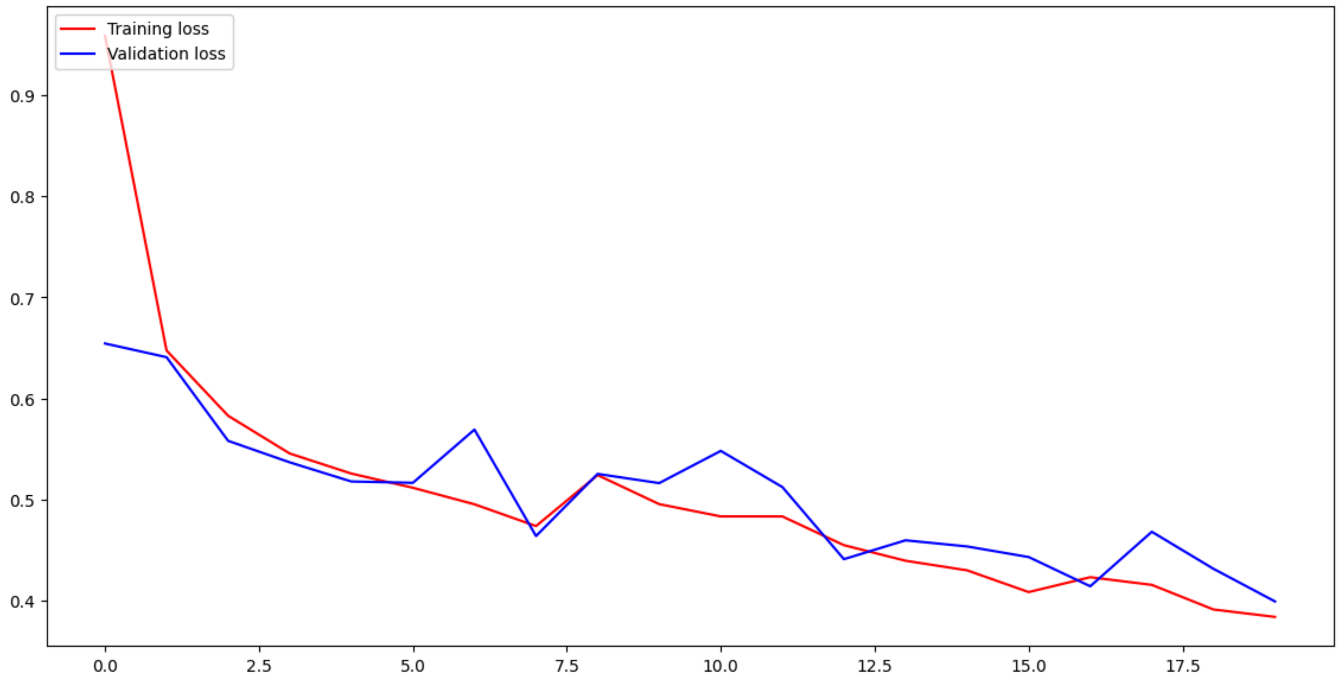
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```


acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
epochs = range(len(acc))
fig = plt.figure(figsize=(14,7))
plt.plot(epochs,acc,'r',label="Training Accuracy")
plt.plot(epochs,val_acc,'b',label="Validation Accuracy")
plt.legend(loc='upper left')
plt.show()
```



```
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
fig = plt.figure(figsize=(14,7))
plt.plot(epochs,loss,'r',label="Training loss")
plt.plot(epochs,val_loss,'b',label="Validation loss")
plt.legend(loc='upper left')
plt.show()
```



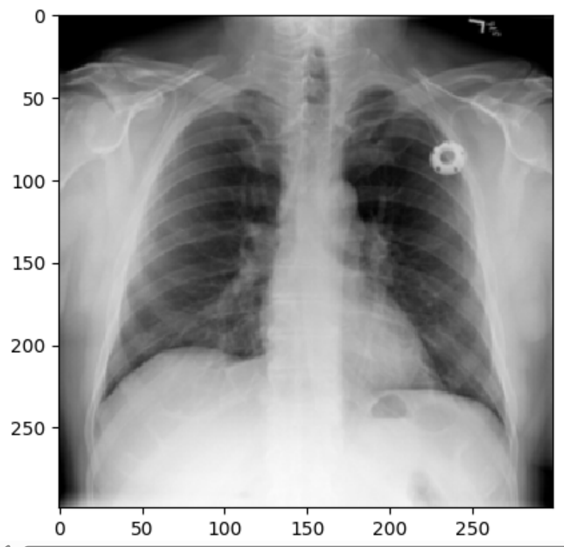
```
img = cv2.imread('/content/drive/MyDrive/dataset/val/Normal/Normal-5637.png')
img = cv2.resize(img,(150,150))
img_array = np.array(img)
img_array.shape
```

```
(150, 150, 3)
```

```
img_array = img_array.reshape(1,150,150,3)
img_array.shape
```

```
(1, 150, 150, 3)
```

```
from tensorflow.keras.preprocessing import image
img = image.load_img('/content/drive/MyDrive/dataset/val/Normal/Normal-5637.png')
plt.imshow(img,interpolation='nearest')
plt.show()
```



```
a=model.predict(img_array)
indices = a.argmax()
indices
if indices==0:
  print("COVID")
else:
  print("Normal")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
Normal
```

Start coding or generate with AI.