```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# DEfine image size and batch size
IMG_SIZE=325
BATCH_SIZE=32

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

train_datagen =
ImageDataGenerator(rescale=1./255,validation_split=0.2) # Correct the
typo here

train_generator = train_datagen.flow_from_directory(
   '/content/drive/MyDrive/Covid19-dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='training'
)
validation_generator = train_datagen.flow_from_directory(
    '/content/drive/MyDrive/Covid19-dataset',
    target_size=(IMG_SIZE,IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='binary',
    subset='validation'
)
```

```
Found 254 images belonging to 2 classes.
Found 63 images belonging to 2 classes.
```

```
# Define the model
model = keras.Sequential([
    layers.Conv2D(32,
(3,3),activation='relu',input_shape=(IMG_SIZE,IMG_SIZE,3)),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(64,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Conv2D(128,(3,3),activation='relu'),
    layers.MaxPooling2D(2,2),
    layers.Flatten(),
    layers.Dense(128,activation='relu'),
    layers.Dense(1,activation='sigmoid') #output layer
])
```

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

model.fit(train_generator, validation_data=validation_generator,
epochs=5) # Use 'validation_generator'
```

```
Epoch 1/5
8/8 [==============================] - 142s 17s/step - loss: 2.0471 -
accuracy: 0.7047 - val_loss: 0.5219 - val_accuracy: 0.7937
Epoch 2/5
8/8 [==============================] - 71s 9s/step - loss: 0.4833 -
accuracy: 0.7953 - val_loss: 0.6300 - val_accuracy: 0.7302
Epoch 3/5
8/8 [==============================] - 69s 8s/step - loss: 0.4725 -
accuracy: 0.7638 - val_loss: 0.5981 - val_accuracy: 0.7937
Epoch 4/5
8/8 [==============================] - 69s 8s/step - loss: 0.3751 -
accuracy: 0.8346 - val_loss: 0.7135 - val_accuracy: 0.7937
Epoch 5/5
8/8 [==============================] - 67s 8s/step - loss: 0.3240 -
accuracy: 0.8425 - val_loss: 0.7729 - val_accuracy: 0.7937

<keras.src.callbacks.History at 0x792bf3459bd0>
```

```
model.save('model.h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/
training.py:3103: UserWarning: You are saving your model as an HDF5
file via `model.save()`. This file format is considered legacy. We
recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np

model = load_model('model.h5')
test_image_path =
'/content/drive/MyDrive/Covid19-dataset/train/Covid/012.jpeg'
img = image.load_img(test_image_path, target_size=(325, 325)) # Change
target size to 325x325 to match the model's input layer
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)

img_array = img_array / 255.0

predictions = model.predict(img_array)
print(predictions)
```

```
1/1 [==============================] - 0s 158ms/step
[[0.99943167]]
```

```
if predictions < 0.5:
      print('It is a Normal')
else:
      print('It is a viral phneumonia')
```

```
It is a viral phneumonia
```