


```

In [1]: !unzip '/content/drive/My Drive/Covid19Pred/Dataset_kaggle.zip'

!pip install split_folders

import split_folders
split_folders.ratio('/content/Dataset_kaggle', output="output", seed=1337, ratio=0.8)

import pandas as pd
import numpy as np
import os
import tensorflow as tf
import keras
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
#from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import preprocess_input
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications import ResNet101
#from tensorflow.keras.applications.ResNet152 import preprocess_input

image_size = [224,224]
data_path = '/content/output'

resnet = ResNet101(input_shape= image_size+[3],weights='imagenet',include_top=False)

x = resnet.output
x = GlobalAveragePooling2D()(x)

x = Dense(1024,activation='relu')(x)
x = Dense(1024,activation='relu')(x)
x = Dense(512, activation='relu')(x)

preds = Dense(2,activation='softmax')(x)

model = Model(inputs = resnet.input,outputs=preds)

for layer in resnet.layers:
    layer.trainable = False

train_datagen=ImageDataGenerator(preprocessing_function=preprocess_input) #include
test_datagen=ImageDataGenerator(preprocessing_function=preprocess_input)
train_generator=train_datagen.flow_from_directory('/content/output/train', # this
                                                target_size=(224,224),
                                                color_mode='rgb',
                                                batch_size=32,
                                                class_mode='categorical',
                                                shuffle=True)
test_generator=test_datagen.flow_from_directory('/content/output/val', # this is
                                                target_size=(224,224),
                                                color_mode='rgb',
                                                batch_size=32,
                                                shuffle=False)

```

```

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

print(train_generator.n)
print(train_generator.batch_size)
print(746//32)

step_size_train=train_generator.n//train_generator.batch_size
r = model.fit_generator(generator=train_generator,
                      validation_data=test_generator,
                      steps_per_epoch=step_size_train,
                      epochs=20)

acc=model.evaluate_generator(test_generator)
print(acc[1])

```

```

Archive: /content/drive/my Drive/COVID19Pred/Dataset_kaggle.zip
  creating: Dataset_kaggle/
  creating: Dataset_kaggle/COVID/
 extracting: Dataset_kaggle/COVID/Covid (1).png
 inflating: Dataset_kaggle/COVID/Covid (10).png
 inflating: Dataset_kaggle/COVID/Covid (100).png
 inflating: Dataset_kaggle/COVID/Covid (1000).png
 inflating: Dataset_kaggle/COVID/Covid (1001).png
 inflating: Dataset_kaggle/COVID/Covid (1002).png
 inflating: Dataset_kaggle/COVID/Covid (1003).png
 inflating: Dataset_kaggle/COVID/Covid (1004).png
 inflating: Dataset_kaggle/COVID/Covid (1005).png
 inflating: Dataset_kaggle/COVID/Covid (1006).png
 inflating: Dataset_kaggle/COVID/Covid (1007).png
 inflating: Dataset_kaggle/COVID/Covid (1008).png
 inflating: Dataset_kaggle/COVID/Covid (1009).png
 inflating: Dataset_kaggle/COVID/Covid (101).png
 inflating: Dataset_kaggle/COVID/Covid (1010).png
 inflating: Dataset_kaggle/COVID/Covid (1011).png
 inflating: Dataset_kaggle/COVID/Covid (1012).png

```

In [2]:

```

acc=model.evaluate_generator(test_generator)
print(acc[1])

```

0.9678068161010742

In []: