In [1]:

```
from zipfile import ZipFile
file_name = "CovidDataset.zip"
with ZipFile(file_name,'r') as zip:
  zip.extractall()
  print("Completed")
```

```
Completed
```

In [2]:

```
from tensorflow.keras.layers import *
from tensorflow.keras.models import *
import tensorflow.keras as tf
```

In [3]:

```
#Training model
model = Sequential()   ## creating a blank model
model.add(Conv2D(32,kernel_size=(3,3),activation='relu',input_shape=(224,224,3)))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))    ### reduce the overfitting

model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(128,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())    ### input layer
model.add(Dense(64,activation='relu'))    ## hidden layer of ann
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))   ## output layer

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

In [4]:

```
#Moulding train images
from tensorflow.keras.preprocessing import image
train_datagen = image.ImageDataGenerator(rescale = 1./255, shear_range = 0.2,zoom_range =

test_dataset = image.ImageDataGenerator(rescale=1./255)
```

In [5]:

```
#Reshaping test and validation images
train_generator = train_datagen.flow_from_directory(
    'CovidDataset/Train',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
validation_generator = test_dataset.flow_from_directory(
    'CovidDataset/Val',
    target_size = (224,224),
    batch_size = 32,
    class_mode = 'binary')
```

```
Found 224 images belonging to 2 classes.
Found 60 images belonging to 2 classes.
```

In [6]:

```
#### Train the model
history = model.fit_generator(
    train_generator,
    steps_per_epoch=7,
    epochs = 20,
    validation_data = validation_generator,
    validation_steps=1
)
```

```
C:\Users\lenovo\anaconda3\lib\site-packages\tensorflow\python\keras\engine\training.py:1844: UserWarning: `Mode
ted and will be removed in a future version. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '


Epoch 1/20
7/7 [==============================] - 18s 2s/step - loss: 0.6996 - accuracy: 0.5467 - val_loss: 0.6908 - val_a
Epoch 2/20
7/7 [==============================] - 13s 2s/step - loss: 0.6304 - accuracy: 0.6372 - val_loss: 0.4715 - val_a
Epoch 3/20
7/7 [==============================] - 14s 2s/step - loss: 0.4054 - accuracy: 0.8273 - val_loss: 0.5076 - val_a
Epoch 4/20
7/7 [==============================] - 14s 2s/step - loss: 0.2744 - accuracy: 0.9304 - val_loss: 0.1168 - val_a
Epoch 5/20
7/7 [==============================] - 14s 2s/step - loss: 0.2393 - accuracy: 0.9198 - val_loss: 0.1054 - val_a
Epoch 6/20
7/7 [==============================] - 14s 2s/step - loss: 0.1583 - accuracy: 0.9501 - val_loss: 0.1210 - val_a
Epoch 7/20
7/7 [==============================] - 15s 2s/step - loss: 0.1164 - accuracy: 0.9450 - val_loss: 0.0063 - val_a
Epoch 8/20
7/7 [==============================] - 14s 2s/step - loss: 0.1558 - accuracy: 0.9343 - val_loss: 0.2393 - val_a
Epoch 9/20
7/7 [==============================] - 15s 2s/step - loss: 0.1248 - accuracy: 0.9743 - val_loss: 0.0708 - val_a
Epoch 10/20
7/7 [==============================] - 14s 2s/step - loss: 0.0741 - accuracy: 0.9660 - val_loss: 0.0468 - val_a
Epoch 11/20
7/7 [==============================] - 14s 2s/step - loss: 0.1206 - accuracy: 0.9700 - val_loss: 0.0299 - val_a
Epoch 12/20
7/7 [==============================] - 14s 2s/step - loss: 0.0601 - accuracy: 0.9869 - val_loss: 0.0189 - val_a
Epoch 13/20
7/7 [==============================] - 14s 2s/step - loss: 0.1582 - accuracy: 0.9472 - val_loss: 0.1167 - val_a
Epoch 14/20
7/7 [==============================] - 14s 2s/step - loss: 0.1105 - accuracy: 0.9545 - val_loss: 0.0143 - val_a
Epoch 15/20
7/7 [==============================] - 14s 2s/step - loss: 0.1152 - accuracy: 0.9445 - val_loss: 0.0460 - val_a
Epoch 16/20
7/7 [==============================] - 14s 2s/step - loss: 0.0663 - accuracy: 0.9843 - val_loss: 0.0356 - val_a
Epoch 17/20
7/7 [==============================] - 13s 2s/step - loss: 0.1117 - accuracy: 0.9741 - val_loss: 0.1193 - val_a
Epoch 18/20
7/7 [==============================] - 14s 2s/step - loss: 0.0669 - accuracy: 0.9739 - val_loss: 0.0819 - val_a
Epoch 19/20
7/7 [==============================] - 14s 2s/step - loss: 0.0507 - accuracy: 0.9832 - val_loss: 0.1975 - val_a
Epoch 20/20
7/7 [==============================] - 15s 2s/step - loss: 0.0590 - accuracy: 0.9776 - val_loss: 0.1255 - val_a
```

In [7]:

```
from tensorflow.keras.preprocessing import image
import numpy as np
img = image.load_img('82.jpg',target_size=(224,224))
img = image.img_to_array(img)
img = np.expand_dims(img,axis=0)   ### flattening
ypred = model.predict(img)
if ypred[0][0] == 1:
  print("Covid Negative")
else:
  print("Covid Positive")
```

```
Covid Positive
```

In [8]:

```
ypred[0][0]
```

```
0.0
```

In [9]:

```
#### save the model
model.save("covid_model.h5")
```

In [10]:

```
import tensorflow.keras as tf
mymodel = tf.models.load_model("covid_model.h5")
```

In [ ]: