Kaggle Link - https://www.kaggle.com/datasets/pranavraikokte/covid19-image-dataset

In [5]:

```python
import tensorflow as tf
import keras as ks
```

In [7]:

```python
# Preprocessing
from keras.preprocessing.image import ImageDataGenerator
train_gen=ImageDataGenerator(rescale=1/255,shear_range=0.2,zoom_range=0.2)
train_set=train_gen.flow_from_directory('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my
resume Projects\\Covid19-dataset\\train',target_size=(64,64))
test_gen=ImageDataGenerator(rescale=1/255)
test_set=test_gen.flow_from_directory('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my r
esume Projects\\Covid19-dataset\\test',target_size=(64,64))
```

```
Found 251 images belonging to 3 classes.
Found 66 images belonging to 3 classes.
```

In [10]:

```python
train_set.class_indices
```

Out[10]:

```
{'Covid': 0, 'Normal': 1, 'Viral Pneumonia': 2}
```

In [23]:

```python
#Now Building Model
from keras.models import Sequential
classifier=Sequential()
from keras.layers import Conv2D
classifier.add(Conv2D(filters=32,kernel_size=3,activation='relu',input_shape=[64,64,3]))
from keras.layers import MaxPooling2D
classifier.add(MaxPooling2D(pool_size=2,strides=2))
classifier.add(Conv2D(filters=32,kernel_size=3,activation='relu',input_shape=[64,64,3]))
classifier.add(MaxPooling2D(pool_size=2,strides=2))
from keras.layers import Flatten
classifier.add(Flatten())
```

In [24]:

```python
from keras.layers import Dense
classifier.add(Dense(units=128,activation='relu'))
classifier.add(Dense(units=3,activation='softmax'))
classifier.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy']
)
classifier.fit(x=train_set,validation_data=test_set,epochs=15,batch_size=32)
```

```
Epoch 1/15
8/8 [==============================] - 13s 2s/step - loss: 1.2439 - accuracy: 0.3347 - va
l_loss: 1.0259 - val_accuracy: 0.4242
Epoch 2/15
8/8 [==============================] - 12s 1s/step - loss: 1.0003 - accuracy: 0.5657 - va
l_loss: 0.9072 - val_accuracy: 0.7273
Epoch 3/15
8/8 [==============================] - 12s 2s/step - loss: 0.8155 - accuracy: 0.7689 - va
l_loss: 0.6650 - val_accuracy: 0.7727
Epoch 4/15
8/8 [==============================] - 12s 1s/step - loss: 0.6158 - accuracy: 0.7809 - va
l_loss: 0.4811 - val_accuracy: 0.8030
Epoch 5/15
8/8 [==============================] - 11s 1s/step - loss: 0.3975 - accuracy: 0.8606 - va
l_loss: 0.6285 - val_accuracy: 0.7121
Epoch 6/15
8/8 [==============================] - 11s 1s/step - loss: 0.3870 - accuracy: 0.8446 - va
l loss: 0.2797 - val accuracy: 0.9091
```

```
Epoch 7/15
8/8 [==============================] - 11s 1s/step - loss: 0.2607 - accuracy: 0.8964 - va
l_loss: 0.2317 - val_accuracy: 0.9091
Epoch 8/15
8/8 [==============================] - 12s 1s/step - loss: 0.2293 - accuracy: 0.9402 - va
l_loss: 0.2263 - val_accuracy: 0.9091
Epoch 9/15
8/8 [==============================] - 13s 2s/step - loss: 0.3041 - accuracy: 0.8805 - va
l_loss: 0.3272 - val_accuracy: 0.8333
Epoch 10/15
8/8 [==============================] - 12s 1s/step - loss: 0.2453 - accuracy: 0.9363 - va
l_loss: 0.2866 - val_accuracy: 0.8636
Epoch 11/15
8/8 [==============================] - 12s 1s/step - loss: 0.2004 - accuracy: 0.9203 - va
l_loss: 0.2042 - val_accuracy: 0.8788
Epoch 12/15
8/8 [==============================] - 12s 1s/step - loss: 0.2028 - accuracy: 0.9243 - va
l_loss: 0.2806 - val_accuracy: 0.8788
Epoch 13/15
8/8 [==============================] - 12s 1s/step - loss: 0.2631 - accuracy: 0.9084 - va
l_loss: 0.3574 - val_accuracy: 0.8636
Epoch 14/15
8/8 [==============================] - 12s 2s/step - loss: 0.2263 - accuracy: 0.9084 - va
l_loss: 0.1957 - val_accuracy: 0.9091
Epoch 15/15
8/8 [==============================] - 12s 1s/step - loss: 0.1912 - accuracy: 0.9124 - va
l_loss: 0.2348 - val_accuracy: 0.8939
```

Out[24]:

```
<keras.callbacks.History at 0x26f87f1b6a0>
```

In [ ]:

```
# I give any one of the image for prediction
```

Making Single Prediction

In [25]:

```python
import numpy as np
from PIL import Image
test_image=Image.open('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my resume Projects\
\Covid19-dataset\\0101.jpeg')
test_image=test_image.resize((64,64))
test_image=np.array(test_image)
test_image=np.expand_dims(test_image,axis=0)
result=classifier.predict(test_image)
result
```

```
1/1 [==============================] - 0s 76ms/step
```

Out[25]:

```
array([[0., 1., 0.]], dtype=float32)
```

In [26]:

```python
if result[0][0]==1:
    print('Covid')
elif result[0][1]==1:
    print('Normal')
else:
    print('Viral Pneumonia')
```

```
Normal
```

In [27]:

```python
# 2nd picture
test_image=Image.open('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my resume Projects\
```

```python
                \Covid19-dataset\\0101.jpeg')
test_image=test_image.resize((64,64))
test_image=np.array(test_image)
test_image=np.expand_dims(test_image,axis=0)
result=classifier.predict(test_image)
result

if result[0][0]==1:
    print('Covid')
elif result[0][1]==1:
    print('Normal')
else:
    print('Viral Pneumonia')
```

```
1/1 [==============================] - 0s 19ms/step
Normal
```

In [28]:

```python
# 3rd picture
test_image=Image.open('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my resume Projects\
\Covid19-dataset\\0114.jpeg')
test_image=test_image.resize((64,64))
test_image=np.array(test_image)
test_image=np.expand_dims(test_image,axis=0)
result=classifier.predict(test_image)
result

if result[0][0]==1:
    print('Covid')
elif result[0][1]==1:
    print('Normal')
else:
    print('Viral Pneumonia')
```

```
1/1 [==============================] - 0s 20ms/step
Viral Pneumonia
```

In [29]:

```python
# 4rth picture
test_image=Image.open('C:\\Users\\admin\\Desktop\\Intership tasks\\1 my resume Projects\
\Covid19-dataset\\0115.jpeg')
test_image=test_image.resize((64,64))
test_image=np.array(test_image)
test_image=np.expand_dims(test_image,axis=0)
result=classifier.predict(test_image)
result

if result[0][0]==1:
    print('Covid')
elif result[0][1]==1:
    print('Normal')
else:
    print('Viral Pneumonia')
```

```
1/1 [==============================] - 0s 18ms/step
Viral Pneumonia
```

In [ ]:

In [ ]: