```
In [1]:
                       import keras
                       \textbf{from} \ \texttt{keras.preprocessing.image} \ \textbf{import} \ \texttt{ImageDataGenerator}
  In [8]:
                      #Define the parameters/arguments for ImageDataGenerator class
                       train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)
                       test datagen=ImageDataGenerator(rescale=1./255)
In [11]: #Applying ImageDataGenerator functionality to trainset
                       x\_train=train\_datagen.flow\_from\_directory('/content/Dataset/Dataset/train\_set', target\_size=(128,128), batch\_size=32, class\_mode='binary' larget\_size=(128,128), batch\_size=(128,128), batch\_size=(128,128),
                      Found 436 images belonging to 2 classes.
In [12]: #Applying ImageDataGenerator functionality to testset
                       x\_test=test\_datagen.flow\_from\_directory('/content/Dataset/Dataset/test\_set', target\_size=(128,128), batch\_size=32, class\_mode='binary')
                      Found 121 images belonging to 2 classes.
In [17]: | #import model building libraries
                       #To define Linear initialisation import Sequential
                       from keras.models import Sequential
                       #To add Layers import Dense
                       from keras.layers import Dense
                       #To create Convolution kernel import Convolution2D
                       from keras.layers import Convolution2D
                        #import Maxpooling layer
                       from keras.layers import MaxPooling2D
                       #import flatten layer
                        from keras.layers import Flatten
                       import warnings
                       warnings.filterwarnings('ignore')
In [19]: #initializing the model
                       model=Sequential()
In [20]: | #add convolutional layer
                       model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
                       #add maxpooling layer
                       model.add(MaxPooling2D(pool_size=(2,2)))
                       #add flatten layer
                       model.add(Flatten())
  In [ ]:
```