In [ ]:
```
import keras
from keras.preprocessing.image import ImageDataGenerator
```

In [ ]:
```
#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 [ ]:
```
#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')
```

```
Found 436 images belonging to 2 classes.
```

In [ ]:
```
#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 [ ]:
```
#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 [ ]:
```
#initializing the model
model=Sequential()
```

In [ ]:
```
#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 [ ]:
```
#add hidden layer
model.add(Dense(150,activation='relu'))
#add output layer
model.add(Dense(1,activation='sigmoid'))
```

In [ ]:
```
#configure the learning process
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=["accuracy"])
```

In [ ]:
```
#Training the model
model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4)
```

```
Epoch 1/10
14/14 [==============================] - 27s 2s/step - loss: 0.6515 - accuracy: 0.6445 - val_loss: 0.6824 - val_accuracy: 0.5950
Epoch 2/10
14/14 [==============================] - 27s 2s/step - loss: 0.6512 - accuracy: 0.6445 - val_loss: 0.6798 - val_accuracy: 0.5950
Epoch 3/10
14/14 [==============================] - 25s 2s/step - loss: 0.6510 - accuracy: 0.6445 - val_loss: 0.6803 - val_accuracy: 0.5950
Epoch 4/10
14/14 [==============================] - 25s 2s/step - loss: 0.6511 - accuracy: 0.6445 - val_loss: 0.6791 - val_accuracy: 0.5950
Epoch 5/10
14/14 [==============================] - 25s 2s/step - loss: 0.6509 - accuracy: 0.6445 - val_loss: 0.6803 - val_accuracy: 0.5950
Epoch 6/10
14/14 [==============================] - 25s 2s/step - loss: 0.6510 - accuracy: 0.6445 - val_loss: 0.6810 - val_accuracy: 0.5950
Epoch 7/10
14/14 [==============================] - 25s 2s/step - loss: 0.6509 - accuracy: 0.6445 - val_loss: 0.6805 - val_accuracy: 0.5950
Epoch 8/10
14/14 [==============================] - 25s 2s/step - loss: 0.6511 - accuracy: 0.6445 - val_loss: 0.6796 - val_accuracy: 0.5950
```

```
Epoch 9/10
14/14 [==============================] - 25s 2s/step - loss: 0.6510 - accuracy: 0.6445 - val_loss: 0.6804 - val_accuracy: 0.5950
Epoch 10/10
14/14 [==============================] - 25s 2s/step - loss: 0.6511 - accuracy: 0.6445 - val_loss: 0.6808 - val_accuracy: 0.5950
```

Out[ ]:

In [ ]: