*# Importing the Keras libraries and packages*

from keras.models import Sequential

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.layers import Dense

*# Initialising the CNN*

classifier = Sequential()

*# Step 1 - Convolution*

classifier.add(Convolution2D(32, 3, 3, input\_shape = (64, 64, 3), activation = 'relu'))

*# Step 2 - Pooling*

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

*# Adding a second convolutional layer*

classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))

classifier.add(MaxPooling2D(pool\_size = (2, 2)))

*# Step 3 - Flattening*

classifier.add(Flatten())

*# Step 4 - Full connection*

classifier.add(Dense(output\_dim = 128, activation = 'relu'))

classifier.add(Dense(output\_dim = 1, activation = 'sigmoid'))

*# Compiling the CNN*

classifier.compile(optimizer = 'adam', loss = 'binary\_crossentropy', metrics = ['accuracy'])

*# Part 2 - Fitting the CNN to the images*

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale = 1./255,

shear\_range = 0.2,

zoom\_range = 0.2,

horizontal\_flip = True)

test\_datagen = ImageDataGenerator(rescale = 1./255)

path='../input/dataset/dataset/'

from keras.preprocessing.image import ImageDataGenerator

train\_datagen = ImageDataGenerator(rescale = 1./255,

shear\_range = 0.2,

zoom\_range = 0.2,

horizontal\_flip = True)

test\_datagen = ImageDataGenerator(rescale = 1./255)

training\_set = train\_datagen.flow\_from\_directory(path+'training\_set',

target\_size = (64, 64),

batch\_size = 32,

class\_mode = 'binary')

test\_set = test\_datagen.flow\_from\_directory(path+'test\_set',

target\_size = (64, 64),

batch\_size = 32,

class\_mode = 'binary')

classifier.fit\_generator(training\_set,

samples\_per\_epoch = 360,

nb\_epoch = 5,

validation\_data = test\_set,

nb\_val\_samples = 40)