Dataset/

│

├── Training\_Set/

│ ├── Cats/

│ │ ├── cat\_image1.jpg

│ │ ├── cat\_image2.jpg

│ │ └── ...

│ └── Dogs/

│ ├── dog\_image1.jpg

│ ├── dog\_image2.jpg

│ └── ...

│

└── Test\_Set/

├── Cats/

│ ├── cat\_image1.jpg

│ ├── cat\_image2.jpg

│ └── ...

└── Dogs/

├── dog\_image1.jpg

├── dog\_image2.jpg

└── …

Make sure you have images of cats in the Cats folder and images of dogs in the Dogs folder, both within the Training\_Set and Test\_Set folders.

**CODE SAMPLE**

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

from keras.preprocessing.image import ImageDataGenerator

# Initialising the CNN

classifier = Sequential()

# Step1: Convolution

classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))

# Step2: Pooling

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

# Adding another convolutional layer

classifier.add(Conv2D(32, (3, 3), activation='relu'))

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

# Step3: Flattening

classifier.add(Flatten())

# Step4: Full Connection

classifier.add(Dense(units=128, activation='relu'))

classifier.add(Dense(units=1, activation='sigmoid'))

# Compiling the CNN

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

# Image preprocessing and augmentation

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(

'Dataset/Training\_Set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='binary')

test\_set = test\_datagen.flow\_from\_directory(

'Dataset/Test\_Set',

target\_size=(64, 64),

batch\_size=32,

class\_mode='binary')

# Fitting the CNN to the images

classifier.fit\_generator(

training\_set,

steps\_per\_epoch=8000, # Number of training images

epochs=25,

validation\_data=test\_set,

validation\_steps=2000) # Number of test images