```
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
```