In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
import tensorflow
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [3]:
```
train_Datagen = ImageDataGenerator(rescale= 1./255,
                                   shear_range= 0.2,
                                   zoom_range= 0.2,
                                   horizontal_flip= True)
```

In [4]:
```
training_set = train_Datagen.flow_from_directory('train',
                                                 target_size= (128,128),
                                                 batch_size= 32,
                                                 class_mode= 'binary')
```

Found 40 images belonging to 2 classes.

In [5]:
```
test_Datagen = ImageDataGenerator(rescale= 1./255)
```

In [6]:
```
testing_set = test_Datagen.flow_from_directory('test',
                                               target_size= (128,128),
                                               batch_size= 32,
                                               class_mode= 'binary')
```

Found 20 images belonging to 2 classes.

In [7]:
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Dropout
```

In [8]:
```
model = Sequential()
```

In [9]:
```
model.add(Conv2D(filters = 64,
                 kernel_size = (5,5),
                 activation = 'relu',
                 padding = 'same',
                 input_shape = (128 , 128, 3)))
```

In [10]:
```
model.add(MaxPool2D(pool_size = (2,2), strides = 2))
```

In [11]:
```
model.add(Conv2D(filters = 32,
                 kernel_size = (5,5),
                 activation = 'relu'))
```

In [12]:
```
model.add(MaxPool2D(pool_size = (2,2), strides = 2))
```

In [13]:
```
model.add(Flatten())
```

In [14]:
```
model.add(Dense(units = 128,
                activation = 'relu'))
model.add(Dropout(0.4))
```

In [15]:
```
model.add(Dense(units = 64, activation = 'relu'))
model.add(Dense(units = 1, activation = 'sigmoid'))
```

In [16]:
```
model.compile(optimizer= tensorflow.keras.optimizers.Adam(),
              loss= tensorflow.keras.losses.BinaryCrossentropy(),
              metrics= ['accuracy'])
```

In [17]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 128, 128, 64)      4864

 max_pooling2d (MaxPooling2D  (None, 64, 64, 64)       0
 )

 conv2d_1 (Conv2D)           (None, 60, 60, 32)        51232

 max_pooling2d_1 (MaxPooling  (None, 30, 30, 32)       0
 2D)

 flatten (Flatten)           (None, 28800)             0

 dense (Dense)               (None, 128)               3686528

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 3,750,945
Trainable params: 3,750,945
Non-trainable params: 0
_________________________________________________________________
```

In [18]:
```
model.fit(training_set, validation_data= testing_set, verbose=1, epoch= 50)
```

```
Epoch 1/50
2/2 [==============================] - 1s 762ms/step - loss: 0.7012 - accuracy: 0.6000
- val_loss: 1.3307 - val_accuracy: 0.5000
Epoch 2/50
2/2 [==============================] - 1s 397ms/step - loss: 1.3980 - accuracy: 0.4750
- val_loss: 0.7116 - val_accuracy: 0.5000
Epoch 3/50
2/2 [==============================] - 1s 1s/step - loss: 0.7128 - accuracy: 0.4750 -
val_loss: 0.6949 - val_accuracy: 0.4500
Epoch 4/50
2/2 [==============================] - 1s 946ms/step - loss: 0.6893 - accuracy: 0.4500
- val_loss: 0.7023 - val_accuracy: 0.4500
Epoch 5/50
2/2 [==============================] - 1s 895ms/step - loss: 0.6887 - accuracy: 0.5250
- val_loss: 0.7030 - val_accuracy: 0.4000
Epoch 6/50
2/2 [==============================] - 1s 350ms/step - loss: 0.6891 - accuracy: 0.5000
- val_loss: 0.7046 - val_accuracy: 0.4000
Epoch 7/50
2/2 [==============================] - 1s 372ms/step - loss: 0.6735 - accuracy: 0.6000
- val_loss: 0.7339 - val_accuracy: 0.3000
Epoch 8/50
2/2 [==============================] - 1s 909ms/step - loss: 0.6909 - accuracy: 0.4750
- val_loss: 0.7648 - val_accuracy: 0.3000
Epoch 9/50
2/2 [==============================] - 1s 886ms/step - loss: 0.7245 - accuracy: 0.4000
- val_loss: 0.7035 - val_accuracy: 0.4500
Epoch 10/50
2/2 [==============================] - 1s 361ms/step - loss: 0.6953 - accuracy: 0.5750
- val_loss: 0.7143 - val_accuracy: 0.3500
Epoch 11/50
2/2 [==============================] - 1s 376ms/step - loss: 0.6670 - accuracy: 0.5500
- val_loss: 0.7162 - val_accuracy: 0.3500
Epoch 12/50
2/2 [==============================] - 1s 886ms/step - loss: 0.6814 - accuracy: 0.5500
- val_loss: 0.7093 - val_accuracy: 0.5000
Epoch 13/50
2/2 [==============================] - 1s 367ms/step - loss: 0.7011 - accuracy: 0.5000
- val_loss: 0.6975 - val_accuracy: 0.3500
Epoch 14/50
2/2 [==============================] - 1s 362ms/step - loss: 0.6770 - accuracy: 0.6250
- val_loss: 0.6937 - val_accuracy: 0.5000
Epoch 15/50
2/2 [==============================] - 1s 897ms/step - loss: 0.7205 - accuracy: 0.5500
- val_loss: 0.6977 - val_accuracy: 0.5000
Epoch 16/50
2/2 [==============================] - 1s 337ms/step - loss: 0.7193 - accuracy: 0.4000
- val_loss: 0.6933 - val_accuracy: 0.5000
Epoch 17/50
2/2 [==============================] - 1s 384ms/step - loss: 0.6822 - accuracy: 0.5500
- val_loss: 0.6929 - val_accuracy: 0.5000
Epoch 18/50
2/2 [==============================] - 1s 897ms/step - loss: 0.6899 - accuracy: 0.5000
- val_loss: 0.6968 - val_accuracy: 0.5000
Epoch 19/50
2/2 [==============================] - 1s 367ms/step - loss: 0.6889 - accuracy: 0.5000
- val_loss: 0.7005 - val_accuracy: 0.5000
Epoch 20/50
2/2 [==============================] - 1s 902ms/step - loss: 0.6726 - accuracy: 0.6000
- val_loss: 0.7050 - val_accuracy: 0.4000
Epoch 21/50
2/2 [==============================] - 1s 364ms/step - loss: 0.6752 - accuracy: 0.6000
- val_loss: 0.7118 - val_accuracy: 0.4500
Epoch 22/50
2/2 [==============================] - 1s 382ms/step - loss: 0.6830 - accuracy: 0.5500
- val_loss: 0.7088 - val_accuracy: 0.4000
Epoch 23/50
2/2 [==============================] - 1s 371ms/step - loss: 0.6867 - accuracy: 0.6000
- val_loss: 0.6971 - val_accuracy: 0.4500
Epoch 24/50
2/2 [==============================] - 1s 377ms/step - loss: 0.6788 - accuracy: 0.7750
- val_loss: 0.7038 - val_accuracy: 0.4500
Epoch 25/50
2/2 [==============================] - 1s 359ms/step - loss: 0.6746 - accuracy: 0.6250
- val_loss: 0.7157 - val_accuracy: 0.5000
Epoch 26/50
2/2 [==============================] - 1s 882ms/step - loss: 0.7268 - accuracy: 0.4250
- val_loss: 0.7030 - val_accuracy: 0.3500
Epoch 27/50
2/2 [==============================] - 1s 880ms/step - loss: 0.6678 - accuracy: 0.6250
- val_loss: 0.7050 - val_accuracy: 0.4000
Epoch 28/50
2/2 [==============================] - 1s 869ms/step - loss: 0.7038 - accuracy: 0.5000
- val_loss: 0.6966 - val_accuracy: 0.5500
Epoch 29/50
2/2 [==============================] - 1s 915ms/step - loss: 0.6646 - accuracy: 0.6500
- val_loss: 0.7006 - val_accuracy: 0.4500
Epoch 30/50
2/2 [==============================] - 1s 895ms/step - loss: 0.6599 - accuracy: 0.6500
- val_loss: 0.7050 - val_accuracy: 0.4500
Epoch 31/50
2/2 [==============================] - 1s 905ms/step - loss: 0.6588 - accuracy: 0.6500
- val_loss: 0.6990 - val_accuracy: 0.4500
Epoch 32/50
2/2 [==============================] - 1s 351ms/step - loss: 0.6290 - accuracy: 0.7000
- val_loss: 0.6970 - val_accuracy: 0.5000
Epoch 33/50
2/2 [==============================] - 1s 873ms/step - loss: 0.5953 - accuracy: 0.7000
- val_loss: 0.7084 - val_accuracy: 0.3500
Epoch 34/50
2/2 [==============================] - 1s 348ms/step - loss: 0.6657 - accuracy: 0.6500
- val_loss: 0.7148 - val_accuracy: 0.3500
Epoch 35/50
2/2 [==============================] - 1s 343ms/step - loss: 0.6615 - accuracy: 0.5250
- val_loss: 0.6927 - val_accuracy: 0.6000
Epoch 36/50
2/2 [==============================] - 1s 338ms/step - loss: 0.6227 - accuracy: 0.7250
- val_loss: 0.6837 - val_accuracy: 0.5000
Epoch 37/50
2/2 [==============================] - 1s 872ms/step - loss: 0.6134 - accuracy: 0.8000
- val_loss: 0.6923 - val_accuracy: 0.5000
Epoch 38/50
2/2 [==============================] - 1s 863ms/step - loss: 0.6214 - accuracy: 0.6250
- val_loss: 0.7706 - val_accuracy: 0.6000
Epoch 39/50
2/2 [==============================] - 1s 873ms/step - loss: 0.6098 - accuracy: 0.7000
- val_loss: 0.6789 - val_accuracy: 0.6000
Epoch 40/50
2/2 [==============================] - 1s 376ms/step - loss: 0.5523 - accuracy: 0.8250
- val_loss: 0.6816 - val_accuracy: 0.5500
Epoch 41/50
2/2 [==============================] - 1s 345ms/step - loss: 0.5477 - accuracy: 0.8500
- val_loss: 0.7141 - val_accuracy: 0.5500
Epoch 42/50
2/2 [==============================] - 1s 922ms/step - loss: 0.5387 - accuracy: 0.7250
- val_loss: 0.7078 - val_accuracy: 0.5000
Epoch 43/50
2/2 [==============================] - 1s 366ms/step - loss: 0.4727 - accuracy: 0.7750
- val_loss: 0.7521 - val_accuracy: 0.4500
Epoch 44/50
2/2 [==============================] - 1s 906ms/step - loss: 0.4184 - accuracy: 0.7750
- val_loss: 0.7644 - val_accuracy: 0.5000
Epoch 45/50
2/2 [==============================] - 1s 892ms/step - loss: 0.5171 - accuracy: 0.7500
- val_loss: 0.7293 - val_accuracy: 0.6500
Epoch 46/50
2/2 [==============================] - 1s 363ms/step - loss: 0.4783 - accuracy: 0.8500
- val_loss: 0.7691 - val_accuracy: 0.6500
Epoch 47/50
2/2 [==============================] - 1s 342ms/step - loss: 0.3470 - accuracy: 0.8750
- val_loss: 0.8358 - val_accuracy: 0.6000
Epoch 48/50
2/2 [==============================] - 1s 356ms/step - loss: 0.3992 - accuracy: 0.8250
- val_loss: 1.0483 - val_accuracy: 0.5000
Epoch 49/50
2/2 [==============================] - 1s 864ms/step - loss: 0.4094 - accuracy: 0.8500
- val_loss: 0.9379 - val_accuracy: 0.5000
Epoch 50/50
2/2 [==============================] - 1s 922ms/step - loss: 0.3530 - accuracy: 0.8750
- val_loss: 0.7940 - val_accuracy: 0.6500
<keras.callbacks.History at 0x1e55d692dc0>
```

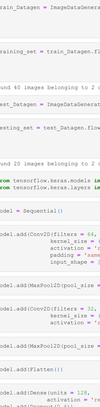
In [19]:
```
import numpy as np
```

In [20]:
```
from tensorflow.keras.preprocessing import image
```

In [21]:
```
test_img = image.load_img('cat.jpg', target_size=(128, 128, 3))
```

In [22]:
```
import matplotlib.pyplot as plt
plt.imshow(test_img)
```

Out[22]:
```
<matplotlib.image.AxesImage at 0x1e55f0920a0>
```



In [23]:
```
test_img = image.img_to_array(test_img)
test_img = np.expand_dims(test_img, axis=0)
test_img = test_img/255
```

In [24]:
```
result = model.predict(test_img)
```

In [25]:
```
training_set.class_indices
```

Out[25]:
```
{'cats': 0, 'dogs': 1}
```

In [26]:
```
result
```

Out[26]:
```
array([[0.18541053]], dtype=float32)
```

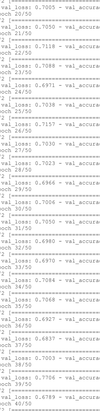
In [27]:
```
if np.round(result[0][0]) == 1 :
    prediction = "It's Dog image.."
else :
    prediction = "It's Cat image.."

print(prediction)
```

```
It's Cat image..
```

In [28]:
```
test_img2 = image.load_img('dog.jpg', target_size=(128, 128, 3))
```

In [29]:
```
plt.imshow(test_img2)
```

Out[29]:
```
<matplotlib.image.AxesImage at 0x1e55f1176a0>
```



In [30]:
```
test_img2 = image.img_to_array(test_img2)
test_img2 = np.expand_dims(test_img2, axis=0)
test_img2 = test_img2/255
```

In [31]:
```
result2 = model.predict(test_img2)
```

In [32]:
```
result2
```

Out[32]:
```
array([[0.8717852]], dtype=float32)
```

In [33]:
```
if np.round(result2[0][0]) == 1 :
    predict = "It's Dog image.."
else :
    predict = "It's Cat image.."

print(predict)
```

```
It's Dog image..
```

In [ ]:
```

```