In [1]:
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
import numpy as np
from tensorflow.keras.preprocessing import image
```

```
WARNING:tensorflow:From C:\Users\Vaishnavi\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The
name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_soft
max_cross_entropy instead.
```

In [2]:
```
train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=Tr
```

In [3]:
```
train_generator = train_datagen.flow_from_directory(r"C:\Users\Public\Documents\train", target_size=(6
```

```
Found 557 images belonging to 2 classes.
```

In [4]:
```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(units=128, activation='relu'))
model.add(Dense(units=1, activation='sigmoid'))
```

```
WARNING:tensorflow:From C:\Users\Vaishnavi\anaconda3\Lib\site-packages\keras\src\backend.py:873: The
name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\Vaishnavi\anaconda3\Lib\site-packages\keras\src\layers\pooling\max_p
ooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
```

In [5]:
```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

```
WARNING:tensorflow:From C:\Users\Vaishnavi\anaconda3\Lib\site-packages\keras\src\optimizers\__init__.
py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

In [24]:
```
model.fit(train_generator, epochs=100, batch_size=16)
```

```
Epoch 1/100
18/18 [==============================] - 9s 489ms/step - loss: 0.2266 - accuracy: 0.8995
Epoch 2/100
18/18 [==============================] - 7s 389ms/step - loss: 0.2103 - accuracy: 0.9264
Epoch 3/100
18/18 [==============================] - 7s 398ms/step - loss: 0.1854 - accuracy: 0.9372
Epoch 4/100
18/18 [==============================] - 8s 421ms/step - loss: 0.1996 - accuracy: 0.9282
Epoch 5/100
18/18 [==============================] - 8s 422ms/step - loss: 0.1885 - accuracy: 0.9228
Epoch 6/100
18/18 [==============================] - 8s 430ms/step - loss: 0.1608 - accuracy: 0.9479
Epoch 7/100
18/18 [==============================] - 7s 405ms/step - loss: 0.1393 - accuracy: 0.9497
Epoch 8/100
18/18 [==============================] - 8s 417ms/step - loss: 0.1603 - accuracy: 0.9425
Epoch 9/100
18/18 [==============================] - 8s 466ms/step - loss: 0.1915 - accuracy: 0.9246
Epoch 10/100
18/18 [==============================] - 7s 406ms/step - loss: 0.1450 - accuracy: 0.9551
```

In [7]:
```
def load_and_preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(64, 64))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.
    return img_array
```

In [8]:
```
new_image_path =r"C:\Users\Public\Documents\test\cats\cat_56.jpg"

new_image = load_and_preprocess_image(new_image_path)
prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 211ms/step
```

In [9]:
```
img = plt.imread(new_image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

In [10]:
```
if prediction[0][0] > 0.5:
    print("It's a Dog!")
else:
    print("It's a Cat!")
```

```
It's a Cat!
```

In [11]:
```
new_image_path =r"C:\Users\Public\Documents\test\cats\cat_106.jpg"

new_image = load_and_preprocess_image(new_image_path)
prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 63ms/step
```

In [12]:
```
img = plt.imread(new_image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

In [13]:
```
if prediction[0][0] > 0.5:
    print("It's a Dog!")
else:
    print("It's a Cat!")
```

```
It's a Cat!
```

In [14]:
```
new_image_path = r"C:\Users\Public\Documents\test\dogs\dog_283.jpg"
```

In [15]:
```
new_image = load_and_preprocess_image(new_image_path)
prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 55ms/step
```

In [16]:
```
img = plt.imread(new_image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

In [17]:
```
if prediction[0][0] > 0.5:
    print("It's a Dog!")
else:
    print("It's a Cat!")
```

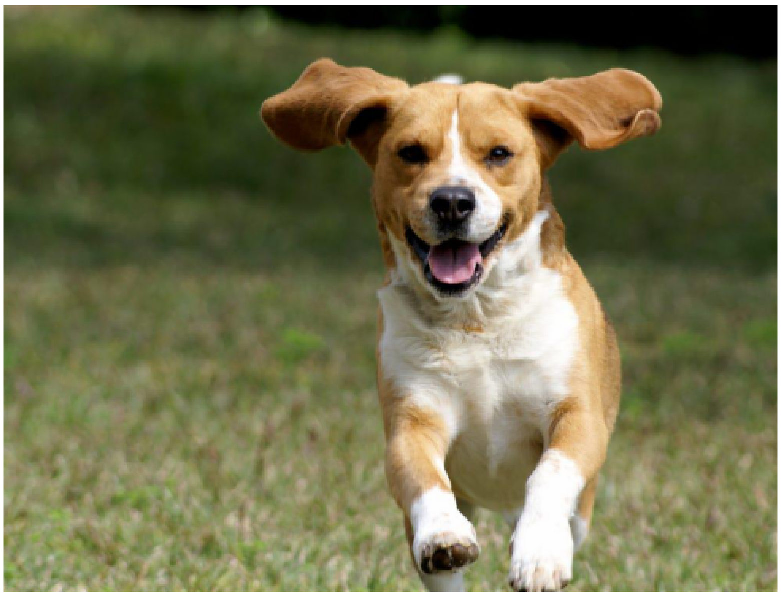
```
It's a Dog!
```

```
In [25]: new_image_path = r"C:\Users\Public\Documents\test\dogs\dog_155.jpg"
```

```
In [26]: new_image = load_and_preprocess_image(new_image_path)
         prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 30ms/step
```

```
In [27]: img = plt.imread(new_image_path)
         plt.imshow(img)
         plt.axis('off')
         plt.show()
```



```
In [28]: if prediction[0][0] > 0.5:
             print("It's a Dog!")
         else:
             print("It's a Cat!")
```

```
It's a Dog!
```

```
In [29]: new_image_path = r"C:\Users\Public\Documents\test\dogs\dog_520.jpg"
```

```
In [30]: new_image = load_and_preprocess_image(new_image_path)
         prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 29ms/step
```

In [31]:
```
img = plt.imread(new_image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

In [32]:
```
if prediction[0][0] > 0.5:
    print("It's a Dog!")
else:
    print("It's a Cat!")
```

```
It's a Dog!
```

In [33]:
```
new_image_path =r"C:\Users\Public\Documents\test\cats\cat_418.jpg"

new_image = load_and_preprocess_image(new_image_path)
prediction = model.predict(new_image)
```

```
1/1 [==============================] - 0s 30ms/step
```

In [34]:
```
img = plt.imread(new_image_path)
plt.imshow(img)
plt.axis('off')
plt.show()
```

In [35]:
```
if prediction[0][0] > 0.5:
    print("It's a Dog!")
else:
    print("It's a Cat!")
```

```
It's a Cat!
```

In [ ]: