In [1]:

```
# Convolutional Neural Network

# Importing the libraries
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator
tf.__version__

# Part 1 - Data Preprocessing

# Preprocessing the Training set
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
training_set = train_datagen.flow_from_directory('/home/anudeep/Desktop/Section 40 - Convolutional Neural Network
s (CNN)/dataset/training_set',
                                                 target_size = (64, 64),
                                                 batch_size = 32,
                                                 class_mode = 'binary')

# Preprocessing the Test set
test_datagen = ImageDataGenerator(rescale = 1./255)
test_set = test_datagen.flow_from_directory('/home/anudeep/Desktop/Section 40 - Convolutional Neural Networks (CN
N)/dataset/test_set',
                                            target_size = (64, 64),
                                            batch_size = 32,
                                            class_mode = 'binary')

# Part 2 - Building the CNN

# Initialising the CNN
cnn = tf.keras.models.Sequential()

# Step 1 - Convolution
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu', input_shape=[64, 64, 3]))

# Step 2 - Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Adding a second convolutional layer
cnn.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, activation='relu'))

#Pooling
cnn.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2))

# Step 3 - Flattening
cnn.add(tf.keras.layers.Flatten())

# Step 4 - Full Connection
cnn.add(tf.keras.layers.Dense(units=128, activation='relu'))

# Step 5 - Output Layer
cnn.add(tf.keras.layers.Dense(units=1, activation='sigmoid'))

# Part 3 - Training the CNN

# Compiling the CNN
cnn.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

# Training the CNN on the Training set and evaluating it on the Test set
cnn.fit(x = training_set, validation_data = test_set, epochs = 25)

# Part 4 - Making a single prediction

import numpy as np
from keras.preprocessing import image
test_image = image.load_img('/home/anudeep/Desktop/dog.jpeg', target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = cnn.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = 'dog'
else:
    prediction = 'cat'
print(prediction)
```

```
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:516: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:517: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:518: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:519: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:520: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/framework/dtypes.py:525: Futur
eWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future version of nu
mpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:541
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint8 = np.dtype([("qint8", np.int8, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:542
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint8 = np.dtype([("quint8", np.uint8, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:543
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint16 = np.dtype([("qint16", np.int16, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:544
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_quint16 = np.dtype([("quint16", np.uint16, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:545
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  _np_qint32 = np.dtype([("qint32", np.int32, 1)])
/home/anudeep/anaconda3/lib/python3.7/site-packages/tensorboard/compat/tensorflow_stub/dtypes.py:550
: FutureWarning: Passing (type, 1) or '1type' as a synonym of type is deprecated; in a future versio
n of numpy, it will be understood as (type, (1,)) / '(1,)type'.
  np_resource = np.dtype([("resource", np.ubyte, 1)])
Using TensorFlow backend.

Found 8000 images belonging to 2 classes.
Found 2000 images belonging to 2 classes.
WARNING:tensorflow:From /home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/ops/in
it_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype is
deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:From /home/anudeep/anaconda3/lib/python3.7/site-packages/tensorflow/python/ops/nn
_impl.py:180: add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprec
ated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Epoch 1/25
250/250 [==============================] - 224s 896ms/step - loss: 0.6735 - acc: 0.5830 - val_loss:
0.6242 - val_acc: 0.6300
Epoch 2/25
250/250 [==============================] - 78s 312ms/step - loss: 0.5988 - acc: 0.6799 - val_loss: 0
.5814 - val_acc: 0.6875
Epoch 3/25
250/250 [==============================] - 85s 342ms/step - loss: 0.5540 - acc: 0.7176 - val_loss: 0
.5655 - val_acc: 0.7220
Epoch 4/25
250/250 [==============================] - 85s 339ms/step - loss: 0.5314 - acc: 0.7316 - val_loss: 0
.5309 - val_acc: 0.7435
Epoch 5/25
250/250 [==============================] - 84s 334ms/step - loss: 0.5083 - acc: 0.7486 - val_loss: 0
.5002 - val_acc: 0.7580
Epoch 6/25
250/250 [==============================] - 83s 334ms/step - loss: 0.4874 - acc: 0.7665 - val_loss: 0
.4874 - val_acc: 0.7740
Epoch 7/25
250/250 [==============================] - 86s 344ms/step - loss: 0.4752 - acc: 0.7721 - val_loss: 0
.4855 - val_acc: 0.7650
```

```
Epoch 8/25
250/250 [==============================] - 68s 273ms/step - loss: 0.4658 - acc: 0.7753 - val_loss: 0
.4875 - val_acc: 0.7675
Epoch 9/25
250/250 [==============================] - 69s 278ms/step - loss: 0.4509 - acc: 0.7875 - val_loss: 0
.4609 - val_acc: 0.7795
Epoch 10/25
250/250 [==============================] - 80s 319ms/step - loss: 0.4405 - acc: 0.7903 - val_loss: 0
.4709 - val_acc: 0.7750
Epoch 11/25
250/250 [==============================] - 72s 287ms/step - loss: 0.4253 - acc: 0.7996 - val_loss: 0
.4903 - val_acc: 0.7675
Epoch 12/25
250/250 [==============================] - 70s 278ms/step - loss: 0.4159 - acc: 0.8106 - val_loss: 0
.4672 - val_acc: 0.7920
Epoch 13/25
250/250 [==============================] - 71s 284ms/step - loss: 0.4071 - acc: 0.8123 - val_loss: 0
.4596 - val_acc: 0.7860
Epoch 14/25
250/250 [==============================] - 69s 277ms/step - loss: 0.3981 - acc: 0.8190 - val_loss: 0
.4477 - val_acc: 0.7925
Epoch 15/25
250/250 [==============================] - 69s 275ms/step - loss: 0.3851 - acc: 0.8273 - val_loss: 0
.4534 - val_acc: 0.7930
Epoch 16/25
250/250 [==============================] - 68s 272ms/step - loss: 0.3869 - acc: 0.8253 - val_loss: 0
.4612 - val_acc: 0.7970
Epoch 17/25
250/250 [==============================] - 70s 280ms/step - loss: 0.3738 - acc: 0.8336 - val_loss: 0
.5129 - val_acc: 0.7755
Epoch 18/25
250/250 [==============================] - 68s 272ms/step - loss: 0.3617 - acc: 0.8391 - val_loss: 0
.4658 - val_acc: 0.8025
Epoch 19/25
250/250 [==============================] - 70s 279ms/step - loss: 0.3531 - acc: 0.8414 - val_loss: 0
.4983 - val_acc: 0.7800
Epoch 20/25
250/250 [==============================] - 68s 272ms/step - loss: 0.3517 - acc: 0.8416 - val_loss: 0
.4889 - val_acc: 0.7725
Epoch 21/25
250/250 [==============================] - 70s 280ms/step - loss: 0.3393 - acc: 0.8501 - val_loss: 0
.4598 - val_acc: 0.8100
Epoch 22/25
250/250 [==============================] - 68s 272ms/step - loss: 0.3288 - acc: 0.8549 - val_loss: 0
.4953 - val_acc: 0.7905
Epoch 23/25
250/250 [==============================] - 70s 279ms/step - loss: 0.3196 - acc: 0.8612 - val_loss: 0
.4716 - val_acc: 0.8050
Epoch 24/25
250/250 [==============================] - 68s 272ms/step - loss: 0.3168 - acc: 0.8609 - val_loss: 0
.5323 - val_acc: 0.7755
Epoch 25/25
250/250 [==============================] - 69s 274ms/step - loss: 0.3058 - acc: 0.8674 - val_loss: 0
.4909 - val_acc: 0.7990
dog
```

In [13]:

```
test_image = image.load_img('/home/anudeep/Desktop/dog1.jpeg', target_size = (64, 64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis = 0)
result = cnn.predict(test_image)
training_set.class_indices
if result[0][0] == 1:
    prediction = 'dog'
else:
    prediction = 'cat'
print(prediction)
```

```
dog
```

In [ ]:

```
```