In [14]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Conv2D, MaxPooling2D
```

In [15]:

```
#data
import os

base_dir = 'cats and dogs'

train_dir = 'cats and dogs/train'
test_dir = 'cats and dogs/test'

train_cats_dir = 'cats and dogs/train/cats'
train_dogs_dir = 'cats and dogs/train/dogs'

test_cats_dir = 'cats and dogs/test/cats'
test_dogs_dir = 'cats and dogs/test/dogs'
```

In [3]:

```
print('total train cat images:', len(os.listdir(train_cats_dir)))
print('total train dog images:', len(os.listdir(train_dogs_dir)))
print('total test cat images:', len(os.listdir(test_cats_dir)))
print('total test dog images:', len(os.listdir(test_dogs_dir)))
```

```
total train cat images: 1002
total train dog images: 1000
total test cat images: 500
total test dog images: 500
```

In [4]:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# All images will be rescaled by 1./255
train_datagen = ImageDataGenerator(rescale=1./255)
test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(train_dir,target_size=(150, 150),batch_
test_generator = test_datagen.flow_from_directory(test_dir,target_size=(150, 150),batch_siz
```

```
Found 2002 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
```

In [5]:

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu',input_shape=(150, 150, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 512)               3211776
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
_________________________________________________________________
```

In [6]:

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=['accuracy'])
model.fit_generator(train_generator,epochs=10)
```

```
Epoch 1/10

C:\Users\swapn\anaconda3\lib\site-packages\tensorflow\python\keras\engine\tr
aining.py:1844: UserWarning: `Model.fit_generator` is deprecated and will be
removed in a future version. Please use `Model.fit`, which supports generato
rs.
  warnings.warn('`Model.fit_generator` is deprecated and '

63/63 [==============================] - 48s 754ms/step - loss: 0.7256 - acc
uracy: 0.5003
Epoch 2/10
63/63 [==============================] - 59s 937ms/step - loss: 0.6851 - acc
uracy: 0.5479
Epoch 3/10
63/63 [==============================] - 66s 1s/step - loss: 0.6432 - accura
cy: 0.6268
Epoch 4/10
63/63 [==============================] - 58s 915ms/step - loss: 0.6187 - acc
uracy: 0.6390
Epoch 5/10
63/63 [==============================] - 53s 842ms/step - loss: 0.5662 - acc
uracy: 0.6990
Epoch 6/10
63/63 [==============================] - 50s 785ms/step - loss: 0.5600 - acc
uracy: 0.6979
Epoch 7/10
63/63 [==============================] - 48s 754ms/step - loss: 0.5106 - acc
uracy: 0.7404
Epoch 8/10
63/63 [==============================] - 53s 836ms/step - loss: 0.4784 - acc
uracy: 0.7643
Epoch 9/10
63/63 [==============================] - 51s 804ms/step - loss: 0.4287 - acc
uracy: 0.8017
Epoch 10/10
63/63 [==============================] - 53s 834ms/step - loss: 0.3679 - acc
uracy: 0.8282
```

Out[6]:

```
<tensorflow.python.keras.callbacks.History at 0x22b9adc5c70>
```

In [7]:

```
train_datagen = ImageDataGenerator(
      rescale=1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')

test_datagen = ImageDataGenerator(rescale=1./255)
```

In [8]:

```
train_generator = train_datagen.flow_from_directory(train_dir,target_size=(150, 150),batch_
test_generator = test_datagen.flow_from_directory(test_dir,target_size=(150, 150),batch_siz
```

```
Found 2002 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
```

In [9]:

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu',input_shape=(150, 150, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

In [12]:

```
model.compile(loss='binary_crossentropy',optimizer="rmsprop",metrics=['accuracy'])
model.fit_generator(train_generator,epochs=10)
```

```
Epoch 1/10
63/63 [==============================] - 53s 832ms/step - loss: 0.7218 - acc
uracy: 0.5779
Epoch 2/10
63/63 [==============================] - 48s 754ms/step - loss: 0.6650 - acc
uracy: 0.6070
Epoch 3/10
63/63 [==============================] - 49s 770ms/step - loss: 0.6675 - acc
uracy: 0.6359
Epoch 4/10
63/63 [==============================] - 50s 792ms/step - loss: 0.6424 - acc
uracy: 0.6604
Epoch 5/10
63/63 [==============================] - 50s 791ms/step - loss: 0.6332 - acc
uracy: 0.6369
Epoch 6/10
63/63 [==============================] - 51s 803ms/step - loss: 0.6311 - acc
uracy: 0.6433
Epoch 7/10
63/63 [==============================] - 50s 786ms/step - loss: 0.6179 - acc
uracy: 0.6694
Epoch 8/10
63/63 [==============================] - 48s 759ms/step - loss: 0.6075 - acc
uracy: 0.6785
Epoch 9/10
63/63 [==============================] - 48s 756ms/step - loss: 0.5959 - acc
uracy: 0.6727
Epoch 10/10
63/63 [==============================] - 50s 789ms/step - loss: 0.5934 - acc
uracy: 0.7033
```

Out[12]:

```
<tensorflow.python.keras.callbacks.History at 0x22b9c8c8160>
```

In [14]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(test_loss)
print(test_accuracy)
```

```
32/32 [==============================] - 5s 160ms/step - loss: 0.6058 - accu
racy: 0.6880
0.6058006882667542
0.6880000233650208
```

Transfer Learning

In [15]:

```
# Using Pretrained Model
# VGG16
from tensorflow.keras.applications import VGG16

vgg_conv_base = VGG16(weights='imagenet',
                      include_top=False,
                      input_shape=(150, 150, 3))

vgg_conv_base.summary()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5 (https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5)
58892288/58889256 [==============================] - 20s 0us/step
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 150, 150, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 150, 150, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 150, 150, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 75, 75, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 75, 75, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 75, 75, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 37, 37, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 37, 37, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 18, 18, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 18, 18, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 9, 9, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
=================================================================
Total params: 14,714,688
```

```
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

In [16]:

```
model = Sequential()
model.add(vgg_conv_base)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

In [17]:

```
vgg_conv_base.trainable = False
```

In [18]:

```
model.compile(loss='binary_crossentropy',optimizer="adam",metrics=['accuracy'])
model.fit_generator(train_generator,epochs=5)
```

```
Epoch 1/5
63/63 [==============================] - 146s 2s/step - loss: 0.9811 - accur
acy: 0.6493
Epoch 2/5
63/63 [==============================] - 143s 2s/step - loss: 0.4002 - accur
acy: 0.8060
Epoch 3/5
63/63 [==============================] - 143s 2s/step - loss: 0.3912 - accur
acy: 0.8275
Epoch 4/5
63/63 [==============================] - 143s 2s/step - loss: 0.3425 - accur
acy: 0.8481
Epoch 5/5
63/63 [==============================] - 141s 2s/step - loss: 0.3379 - accur
acy: 0.8563
```

Out[18]:

```
<tensorflow.python.keras.callbacks.History at 0x22b9c06b4c0>
```

In [19]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(test_loss)
print(test_accuracy)
```

```
32/32 [==============================] - 69s 2s/step - loss: 0.2643 - accura
cy: 0.8880
0.2642878293991089
0.8880000114440918
```

In [21]:

```
# Fine Tuning
model = Sequential()
model.add(vgg_conv_base)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

vgg_conv_base.trainable = True

set_trainable = False
for layer in vgg_conv_base.layers:
    if layer.name == 'block5_conv1':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False


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

In [23]:

```
model.fit_generator(train_generator,epochs=10)
```

```
Epoch 1/10
63/63 [==============================] - 142s 2s/step - loss: 0.6932 - accur
acy: 0.4895
Epoch 2/10
63/63 [==============================] - 149s 2s/step - loss: 0.6932 - accur
acy: 0.4865
Epoch 3/10
63/63 [==============================] - 148s 2s/step - loss: 0.6932 - accur
acy: 0.4995
Epoch 4/10
63/63 [==============================] - 154s 2s/step - loss: 0.6932 - accur
acy: 0.4915
Epoch 5/10
63/63 [==============================] - 149s 2s/step - loss: 0.6932 - accur
acy: 0.4885
Epoch 6/10
63/63 [==============================] - 149s 2s/step - loss: 0.6932 - accur
acy: 0.4865
Epoch 7/10
63/63 [==============================] - 144s 2s/step - loss: 0.6932 - accur
acy: 0.4895
Epoch 8/10
63/63 [==============================] - 141s 2s/step - loss: 0.6932 - accur
acy: 0.5005
Epoch 9/10
63/63 [==============================] - 144s 2s/step - loss: 0.6932 - accur
acy: 0.5005
Epoch 10/10
63/63 [==============================] - 149s 2s/step - loss: 0.6932 - accur
acy: 0.4925
```

Out[23]:

```
<tensorflow.python.keras.callbacks.History at 0x22b9eb2fd30>
```

In [24]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(test_loss)
print(test_accuracy)
```

```
32/32 [==============================] - 53s 2s/step - loss: 0.6931 - accura
cy: 0.5000
0.6931486129760742
0.5
```

In [37]:

```
from tensorflow.keras.applications.inception_v3 import InceptionV3

local_weights_file = 'inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'
```

In [38]:

```
inception_conv_base = InceptionV3(input_shape = (150, 150, 3),
                                  include_top = False,
                                  weights = None)

inception_conv_base.load_weights(local_weights_file)

inception_conv_base.trainable = False
```

In [39]:

```
model = Sequential()
model.add(inception_conv_base)
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer="adam",metrics=['accuracy'])
model.fit_generator(train_generator,epochs=10)
```

```
Epoch 1/10
63/63 [==============================] - 55s 764ms/step - loss: 7.3468 - acc
uracy: 0.7559
Epoch 2/10
63/63 [==============================] - 48s 758ms/step - loss: 0.3297 - acc
uracy: 0.9283
Epoch 3/10
63/63 [==============================] - 50s 789ms/step - loss: 0.5835 - acc
uracy: 0.8851
Epoch 4/10
63/63 [==============================] - 55s 870ms/step - loss: 0.4896 - acc
uracy: 0.9224
Epoch 5/10
63/63 [==============================] - 47s 739ms/step - loss: 0.3909 - acc
uracy: 0.9116
Epoch 6/10
63/63 [==============================] - 51s 803ms/step - loss: 0.2618 - acc
uracy: 0.9283
Epoch 7/10
63/63 [==============================] - 49s 773ms/step - loss: 0.2450 - acc
uracy: 0.9304
Epoch 8/10
63/63 [==============================] - 48s 761ms/step - loss: 0.2214 - acc
uracy: 0.9344
Epoch 9/10
63/63 [==============================] - 45s 718ms/step - loss: 0.2132 - acc
uracy: 0.9326
Epoch 10/10
63/63 [==============================] - 43s 679ms/step - loss: 0.2043 - acc
uracy: 0.9368
```

Out[39]:

```
<tensorflow.python.keras.callbacks.History at 0x22bb682a100>
```

In [40]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(test_loss)
print(test_accuracy)
```

```
32/32 [==============================] - 19s 511ms/step - loss: 0.1644 - acc
uracy: 0.9540
0.16438768804073334
0.9539999961853027
```

ResNet model

In [22]:

```
from tensorflow.keras.applications import ResNet50

model = ResNet50(input_shape=(224, 224,3), include_top=False, weights="imagenet")
```

In [23]:

```
for layer in model.layers:
    layer.trainable = False
```

In [25]:

```
from tensorflow.keras.applications import ResNet50
from tensorflow.python.keras.models import Sequential
#from tensorflow.python.keras.layers import Dense, Flatten, GlobalAveragePooling2

model = Sequential()
model.add(ResNet50(include_top=False, weights='imagenet', pooling='max'))
model.add(Dense(1, activation='sigmoid'))
```

In [26]:

```
model.compile(optimizer ='sgd', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

In [27]:

```
model.fit(train_generator, epochs = 5)
```

```
Epoch 1/5
63/63 [==============================] - 531s 8s/step - loss: 6.5539 - accur
acy: 0.6524
Epoch 2/5
63/63 [==============================] - 528s 8s/step - loss: 1.1227 - accur
acy: 0.7821
Epoch 3/5
63/63 [==============================] - 529s 8s/step - loss: 0.4470 - accur
acy: 0.8953
Epoch 4/5
63/63 [==============================] - 489s 8s/step - loss: 0.2802 - accur
acy: 0.9428
Epoch 5/5
63/63 [==============================] - 487s 8s/step - loss: 0.2332 - accur
acy: 0.9496
```

Out[27]:

```
<tensorflow.python.keras.callbacks.History at 0x27d9a4a1a90>
```

In [28]:

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(test_loss)
print(test_accuracy)
```

```
32/32 [==============================] - 36s 1s/step - loss: 0.9564 - accura
cy: 0.5000
0.9563969969749451
0.5
```

In [ ]:

```

```