Transfer Learning

In [ ]:
```
# import vgg16 model from keras application
from keras.applications.vgg16 import VGG16, preprocess_input, decode_predictions
from keras.preprocessing import image
import numpy as np
```

In [ ]:
```
# create the model
model = VGG16(weights='imagenet')
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels.h5
553467096/553467096 [==============================] - 20s 0us/step
```

In [ ]:
```
# check model summary
model.summary()
```

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 fc1 (Dense)                 (None, 4096)              102764544

 fc2 (Dense)                 (None, 4096)              16781312

 predictions (Dense)         (None, 1000)              4097000

=================================================================
Total params: 138357544 (527.79 MB)
Trainable params: 138357544 (527.79 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [ ]:
```
# Preprocessing for Predition
img_path = 'Cat.jpg'
img = image.load_img(img_path,target_size=(224,224))
x= image.img_to_array(img)
x = np.expand_dims(x,axis=0)
x= preprocess_input(x)
```

In [ ]:
```
# Predict
preds = model.predict(x)
```

```
1/1 [==============================] - 1s 946ms/step
```

In [ ]:
```
# decode the prediction
print('Prediction :', decode_predictions(preds, top=3)[0])
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
35363/35363 [==============================] - 0s 0us/step
Prediction : [('n02124075', 'Egyptian_cat', 0.2028058), ('n02127052', 'lynx', 0.08188627), ('n02123045', 'tabby', 0.06460435)]
```

In [ ]:
```
# Preprocessing for Predition
img_path = 'dog.jpg'
img = image.load_img(img_path,target_size=(224,224))
x= image.img_to_array(img)
x = np.expand_dims(x,axis=0)
x= preprocess_input(x)
```

In [ ]:
```
# Predict
preds = model.predict(x)
```

```
1/1 [==============================] - 1s 559ms/step
```

In [ ]:
```
# decode the prediction
print('Prediction :', decode_predictions(preds, top=3)[0])
```

```
Prediction : [('n02099601', 'golden_retriever', 0.51487076), ('n02099712', 'Labrador_retriever', 0.4799755), ('n02104029', 'kuvasz', 0.0007487086)]
```

In [ ]:
```
# Preprocessing for Predition
img_path = 'elephant.jpg'
img = image.load_img(img_path,target_size=(224,224))
x= image.img_to_array(img)
x = np.expand_dims(x,axis=0)
x= preprocess_input(x)
```

In [ ]:
```
# Predict
preds = model.predict(x)
```

```
1/1 [==============================] - 1s 869ms/step
```

In [ ]:
```
# decode the prediction
print('Prediction :', decode_predictions(preds, top=3)[0])
```

```
Prediction : [('n02504458', 'African_elephant', 0.76421064), ('n01871265', 'tusker', 0.20635526), ('n02504013', 'Indian_elephant', 0.029307453)]
```