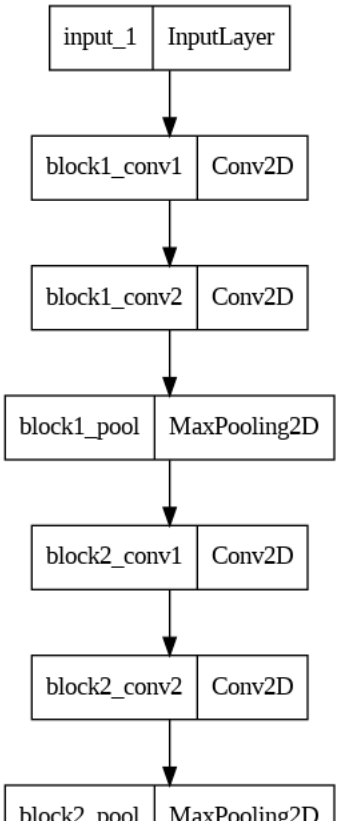
```
from keras.applications.vgg16 import VGG16
model= VGG16()
print(model.summary())
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernel
553467096/553467096 [==============================] - 7s 0us/step
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)
```

```
from tensorflow.keras.utils import plot_model
plot_model(model, to_file='vgg.png')
```




```
from tensorflow.keras.utils import load_img
#load an image from file
image = load_img('/content/drive/MyDrive/Colab Notebooks/bird_1.jpg',target_size=(224,224))
```

```
from tensorflow.keras.utils import img_to_array
#convert the image in pixel to a numpy array
image = img_to_array(image)
```

```
#reshapeing data for the model
image = image.reshape((1,image.shape[0], image.shape[1],image.shape[2]))
```

```
from keras.applications.vgg16 import preprocess_input
#perpare the image for the VGG16 model
image = preprocess_input(image)
```

```
#perdict the probalibility across all output classes
yhat = model.predict(image)
```

```
1/1 [==============================] - 8s 8s/step
```

```
yhat
```

```
array([[2.00670050e-10, 2.65897859e-10, 6.10448801e-12, 1.90773265e-12,
        7.85185556e-13, 4.06526812e-10, 1.28962396e-11, 1.04403534e-06,
        3.15370744e-05, 1.64790185e-06, 2.69491210e-07, 2.13480092e-07,
        3.29109838e-07, 2.79485647e-07, 4.35953496e-09, 2.95582722e-05,
        1.03316881e-06, 5.34186825e-07, 6.54838971e-07, 9.18402421e-09,
        3.83511804e-07, 3.89485923e-03, 1.22179072e-05, 2.92428558e-06,
        9.13861215e-01, 1.06926488e-10, 8.86842405e-11, 6.20556870e-11,
        2.93198688e-10, 8.41678203e-12, 4.89270673e-11, 2.72457445e-10,
        6.48458995e-11, 1.52224604e-11, 2.89407075e-11, 3.00841185e-10,
        1.22345689e-09, 1.44574299e-08, 9.21927465e-11, 1.46833363e-08,
        1.81321469e-09, 2.21009300e-09, 2.54114885e-09, 6.81236187e-08,
        1.40473755e-09, 8.99046948e-11, 4.80812501e-10, 7.70016051e-10,
        5.52919897e-11, 2.31438246e-09, 2.92841557e-10, 6.63106015e-09,
        2.90934030e-11, 1.14238996e-09, 4.89284058e-10, 9.75102030e-12,
        2.61158567e-10, 4.29326297e-10, 3.81770587e-10, 8.68037420e-10,
        8.55559901e-10, 3.24847128e-11, 1.36243725e-10, 1.41315426e-09,
        1.92697802e-10, 8.55892510e-11, 2.81435719e-09, 8.66715102e-11,
        2.16111976e-10, 4.78799222e-10, 1.28971935e-08, 1.66434540e-11,
        1.10510561e-08, 1.27413466e-08, 1.33440003e-09, 2.25245955e-08,
        8.36581415e-10, 3.61644359e-09, 3.47272072e-10, 2.90985969e-09,
        2.65742619e-05, 4.36830214e-06, 4.99521324e-04, 8.10156390e-02,
        1.35285831e-07, 1.03426515e-04, 9.00096275e-05, 1.15631813e-07,
        1.14102636e-06, 6.94320050e-08, 7.45222860e-06, 9.35231981e-07,
        1.25903782e-07, 1.91408872e-05, 2.42723615e-07, 1.30774467e-07,
        2.21728484e-08, 2.62739661e-08, 1.59940143e-07, 2.19477300e-07,
```

```
1.70421650e-08, 3.31464613e-11, 4.64840832e-09, 1.32378511e-10,
3.05676678e-10, 2.81803136e-09, 4.42411663e-10, 3.18830864e-11,
6.09288564e-10, 2.61077909e-10, 6.40391351e-10, 5.72284928e-11,
7.23571536e-10, 3.41845285e-09, 5.68394309e-09, 4.33783720e-09,
1.94871376e-11, 6.73537244e-08, 4.77336365e-11, 5.73495591e-11,
2.06454281e-11, 2.52736061e-11, 1.89273708e-10, 7.40966566e-10,
4.41633663e-09, 5.41364661e-11, 3.71968595e-10, 8.18302137e-09,
1.14927325e-07, 3.33323982e-08, 7.22852833e-10, 4.16734878e-08,
2.08293542e-07, 1.38812786e-04, 3.37712095e-06, 1.02725962e-05,
2.65464344e-08, 6.88065853e-08, 1.42053541e-04, 4.10563672e-08,
1.29625846e-06, 3.12412631e-05, 6.24977758e-08, 2.22075428e-07,
2.19786671e-07, 9.57792054e-07, 7.82635325e-08, 6.79519566e-11,
5.72111108e-12, 6.77419926e-13, 1.36179804e-10, 1.29673064e-10,
2.53057109e-09, 7.38146419e-11, 4.18817603e-09, 1.14443355e-09,
1.17871560e-10, 1.15267706e-09, 6.69854505e-10, 6.61562888e-11,
1.89170124e-10, 2.00323074e-11, 1.06584958e-11, 1.95249372e-11,
2.52957699e-09, 2.62647210e-10, 3.07587809e-11, 4.79923566e-11,
5.30933214e-11, 1.26162372e-10, 2.85725277e-10, 7.88743463e-12,
8.23031573e-11, 1.64736662e-11, 4.02923361e-10, 7.46739129e-11,
1.17012233e-11, 9.83372758e-11, 1.70050030e-10, 1.30794525e-11,
3.76763863e-11, 8.59085025e-11, 3.24578503e-10, 3.44858613e-08,
2.74726752e-11, 1.31043051e-10, 1.09414633e-09, 3.03316572e-09,
1.21548507e-10, 1.45363209e-11, 3.11886246e-11, 1.60075092e-10,
2.36511144e-10, 1.97173083e-10, 3.04587050e-11, 8.71333450e-10,
5.91286575e-10, 7.20595250e-10, 9.58269997e-09, 1.99048791e-10,
9.49809675e-10, 3.53484381e-10, 6.48293641e-10, 7.11021589e-11,
7.81518739e-10, 1.09440101e-09, 8.29089644e-11, 4.08193201e-11,
1.27978392e-10, 5.38106365e-11, 1.11634435e-08, 1.88887461e-10,
7.35909156e-09, 6.85979658e-12, 2.28880670e-09, 1.91340138e-10,
1.12989874e-10, 9.66269842e-10, 7.55593366e-11, 1.91316393e-10,
3.21886184e-10, 8.26811231e-10, 2.20273147e-10, 3.67942100e-11,
3.43978422e-11, 3.12046472e-10, 2.20495393e-11, 4.72401285e-10,
4.27750591e-09, 6.18126050e-10, 2.65927280e-10, 9.94883398e-10,
```

```
from keras.applications.vgg16 import decode_predictions
#convey the probalilities class labels
label = decode_predictions(yhat)
label = label [0][0]
```

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json
35363/35363 [==============================] - 0s 0us/step
```

```
label
```

```
('n01622779', 'great_grey_owl', 0.9138612)
```

```
print('%s (%.2f%%)' % (label[1],label[2]*100))
```

```
great_grey_owl (91.39%)
```