

✓ VGG16

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

```
print(model.summary())
```

Model: "vgg16"

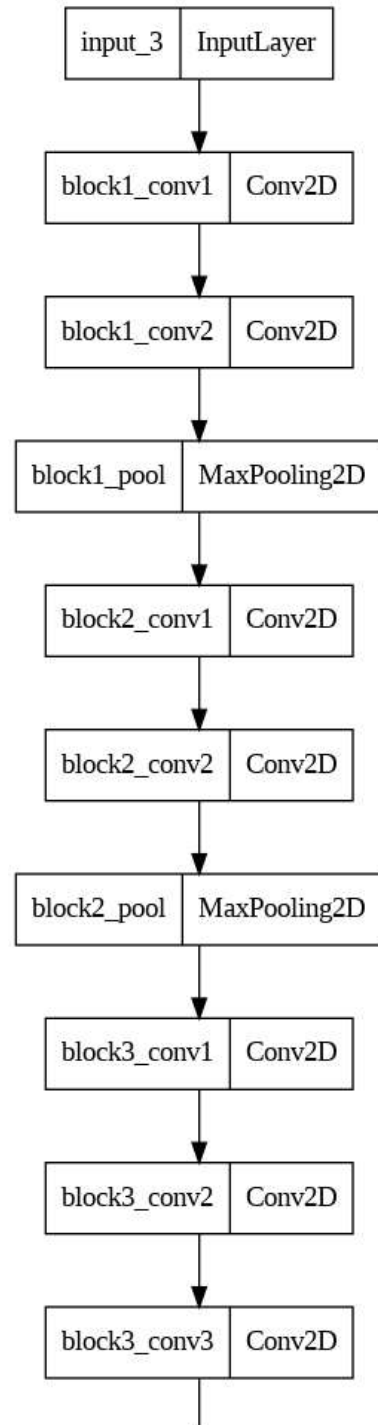
Layer (type)	Output Shape	Param #
=====		
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(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

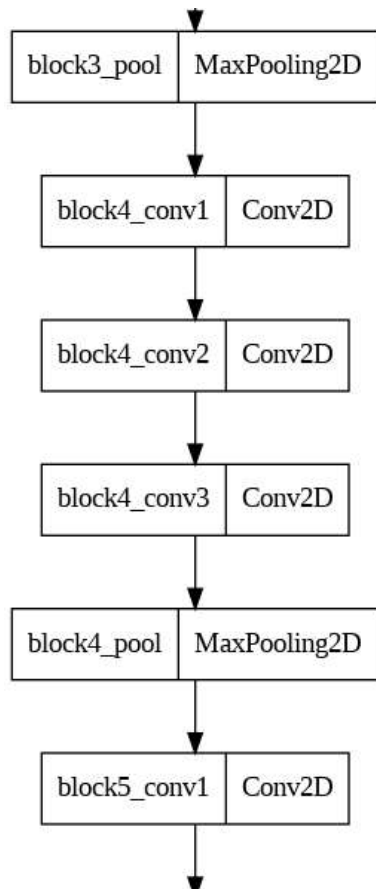
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

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

None

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





```
# from keras.preprocessing.image import load_img
# #load an image from file
# image = load_img('test1.jpg', target_size =(224,224))
```

⊥

```
from keras.preprocessing.image import load_img
from keras.preprocessing import image
from google.colab import files
#load an image from file
uploaded = files.upload()
# image = load_img('uploaded', target_size =(224,224))
for fn in uploaded.keys():
    path='/content/' + fn      #Save the image to content folder
    image=image.load_img(path, target_size=(224, 224))    #load the image
```

pexels-pixabay-70083.jpg

- **pexels-pixabay-70083.jpg**(image/jpeg) - 419539 bytes, last modified: 25/12/2023 - 100% done
Saving pexels-pixabay-70083.jpg to pexels-pixabay-70083.jpg

|

```
from keras.preprocessing.image import img_to_array
#convert the image pixels to numpy array
image = img_to_array(image)
```

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

```
from keras.applications.vgg16 import preprocess_input
#prepare the image from the VGG model
image = preprocess_input(image)
```

```
#predict the probability across all output class
yhat = model.predict(image)
```

```
1/1 [=====] - 0s 481ms/step
```

```
from keras.applications.vgg16 import decode_predictions
#connect the probabilities to class labels
label = decode_predictions(yhat)
#retrieve the most likely result, e.g highest probability
label = label[0][0]
#predict the classification
print('%s(%.2f%%)'%(label[1],label[2]*100))
```

```
tree_frog(97.23%)
```