```
from google.colab import drive
drive.mount('/content/drive')
Drive already mounted at /content/drive; to attempt to forcibly re
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataG
from tensorflow.keras.applications.vgg16 import VGG16, prepro
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
imageSize = [224, 224]

trainPath = r"/content/drive/MyDrive/dataset1/body/training"

testPath = r"/content/drive/MyDrive/dataset1/body/validation"
# adding preprocessing layers to the front of vgg

vgg = VGG16(input_shape=imageSize + [3], weights='imagenet',i
Downloading data from https://storage.googleapis.com/tensorflo
58889256/58889256 [==============================] - 0s 0
# don't train existing weights
for layer in vgg.layers:
  layer.trainable = False
# our layers - you can add more if you want
x = Flatten()(vgg.output)
prediction = Dense(3, activation='softmax')(x)
# create a model object
model = Model(inputs=vgg.input, outputs=prediction)
# view the structure of the model
model.summary()
Model: "model"
```

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
dense (Dense)	(None, 3)	75267

```
Total params: 14,789,955
Trainable params: 75,267
Non-trainable params: 14,714,688

# tell the model what cost and optimization method to use
model.compile(
```