from tensorflow.keras.preprocessing.image import load\_img

from tensorflow.keras.preprocessing.image import img\_to\_array

from keras.applications.vgg16 import preprocess\_input

from keras.applications.vgg16 import decode\_predictions

from keras.applications.vgg16 import VGG16

# load an image from file

image = load\_img('download.jpg', target\_size=(224, 224))

# convert the image pixels to a numpy array

image = img\_to\_array(image)

# reshape data for the model

image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))

# prepare the image for the VGG model

image = preprocess\_input(image)

# load the model

model = VGG16()

# predict the probability across all output classes

yhat = model.predict(image)

# convert the probabilities to class labels

label = decode\_predictions(yhat)

# retrieve the most likely result, e.g. highest probability

label = label[0][0]

# print the classification

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