

Psychoinformatics - Week 13 (Exercises)

by your name (your email)

1. 進一步研究CNN (3 points)

1.1 為何ResNet50會判斷小女孩照片為ping-pong_bal, bubble, or Band_Aid? (3 points)

```
In [1]: import numpy as np
import urllib.request
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions
from PIL import Image
import matplotlib.pyplot as plt
import json
```

```
In [ ]: model = ResNet50(weights='imagenet')

urllib.request.urlretrieve('http://mil.psy.ntu.edu.tw/~tren/girl.jpg', 'girl.jpg')
img = image.load_img('girl.jpg', target_size=(224, 224)) # Or use cv2.resize
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])
```

Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels.h5

102967424/102967424 ————— 0s 0us/step

1/1 ————— 6s 6s/step

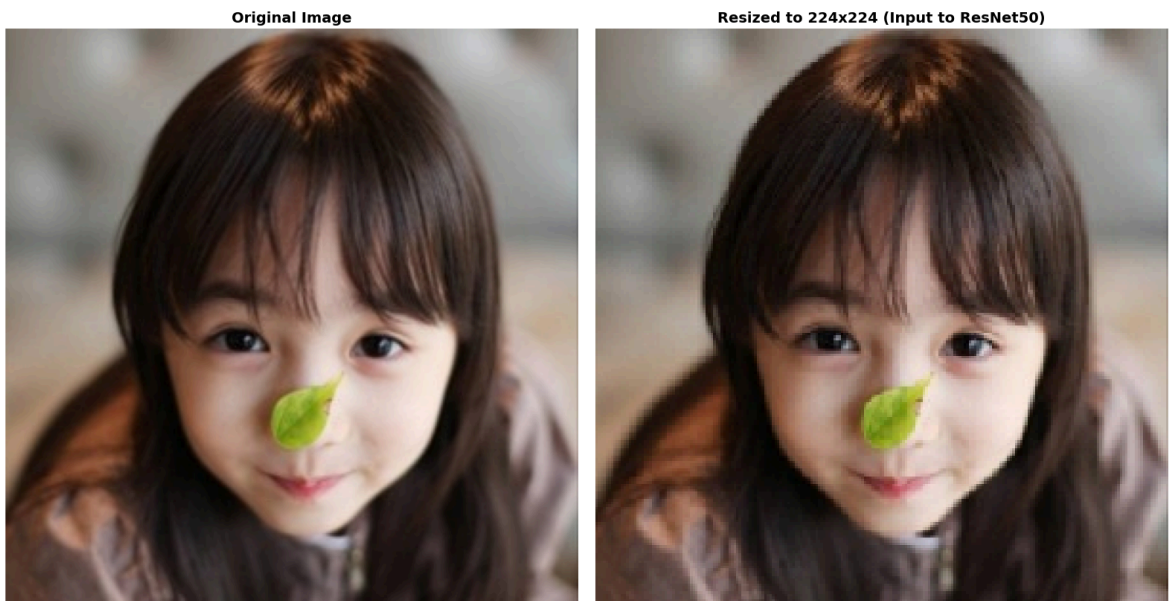
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json

35363/35363 ————— 0s 0us/step

Predicted: [('n03942813', 'ping-pong_ball', np.float32(0.17008679)), ('n09229709', 'bubble', np.float32(0.10647205)), ('n02786058', 'Band_Aid', np.float32(0.103644624))]

```
In [6]: # 載入圖片
img_display = Image.open('girl.jpg')
img_resized = image.load_img('girl.jpg', target_size=(224, 224))
# 創建subplot
fig, axes = plt.subplots(1, 2, figsize=(16, 8))
# 左邊：原始圖片
axes[0].imshow(img_display)
axes[0].set_title('Original Image', fontsize=14, fontweight='bold')
axes[0].axis('off')
# 右邊：resize後的圖片
axes[1].imshow(img_resized)
```

```
axes[1].set_title('Resized to 224x224 (Input to ResNet50)', fontsize=14,  
axes[1].axis('off')  
plt.tight_layout()  
plt.show()
```



```
In [7]: print(model.summary())
```

Model: "resnet50"

Layer (type)	Output Shape	Param #	Connected to
input_layer (InputLayer)	(None, 224, 224, 3)	0	–
conv1_pad (ZeroPadding2D)	(None, 230, 230, 3)	0	input_layer
conv1_conv (Conv2D)	(None, 112, 112, 64)	9,472	conv1_pad[0]
conv1_bn (BatchNormalizatio...	(None, 112, 112, 64)	256	conv1_conv
conv1_relu (Activation)	(None, 112, 112, 64)	0	conv1_bn[0]
pool1_pad (ZeroPadding2D)	(None, 114, 114, 64)	0	conv1_relu
pool1_pool (MaxPooling2D)	(None, 56, 56, 64)	0	pool1_pad[0]
conv2_block1_1_conv (Conv2D)	(None, 56, 56, 64)	4,160	pool1_pool
conv2_block1_1_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block1_1_conv
conv2_block1_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_1_bn
conv2_block1_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block1_1_relu
conv2_block1_2_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block1_2_conv
conv2_block1_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block1_2_bn
conv2_block1_0_conv (Conv2D)	(None, 56, 56, 256)	16,640	pool1_pool
conv2_block1_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block1_2_relu
conv2_block1_0_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block1_3_conv
conv2_block1_3_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block1_0_bn
conv2_block1_add (Add)	(None, 56, 56, 256)	0	conv2_block1_3_bn, conv2_block1_0_conv
conv2_block1_out (Activation)	(None, 56, 56, 256)	0	conv2_block1_add
conv2_block2_1_conv (Conv2D)	(None, 56, 56, 64)	16,448	conv2_block1_out
conv2_block2_1_bn	(None, 56, 56, 64)	256	conv2_block2_1_conv

(BatchNormalizatio...	64)		
conv2_block2_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block
conv2_block2_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block
conv2_block2_2_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block
conv2_block2_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block
conv2_block2_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block
conv2_block2_3_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block
conv2_block2_add (Add)	(None, 56, 56, 256)	0	conv2_block conv2_block
conv2_block2_out (Activation)	(None, 56, 56, 256)	0	conv2_block
conv2_block3_1_conv (Conv2D)	(None, 56, 56, 64)	16,448	conv2_block
conv2_block3_1_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block
conv2_block3_1_relu (Activation)	(None, 56, 56, 64)	0	conv2_block
conv2_block3_2_conv (Conv2D)	(None, 56, 56, 64)	36,928	conv2_block
conv2_block3_2_bn (BatchNormalizatio...	(None, 56, 56, 64)	256	conv2_block
conv2_block3_2_relu (Activation)	(None, 56, 56, 64)	0	conv2_block
conv2_block3_3_conv (Conv2D)	(None, 56, 56, 256)	16,640	conv2_block
conv2_block3_3_bn (BatchNormalizatio...	(None, 56, 56, 256)	1,024	conv2_block
conv2_block3_add (Add)	(None, 56, 56, 256)	0	conv2_block conv2_block
conv2_block3_out (Activation)	(None, 56, 56, 256)	0	conv2_block
conv3_block1_1_conv (Conv2D)	(None, 28, 28, 128)	32,896	conv2_block
conv3_block1_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block
conv3_block1_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block

conv3_block1_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block1_2_conv
conv3_block1_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block1_2_bn
conv3_block1_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block1_2_relu
conv3_block1_0_conv (Conv2D)	(None, 28, 28, 512)	131,584	conv2_block1_0_conv
conv3_block1_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block1_3_conv
conv3_block1_0_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block1_0_bn
conv3_block1_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block1_3_bn
conv3_block1_add (Add)	(None, 28, 28, 512)	0	conv3_block1_add conv3_block1_0_conv
conv3_block1_out (Activation)	(None, 28, 28, 512)	0	conv3_block1_out
conv3_block2_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block2_1_conv
conv3_block2_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block2_1_bn
conv3_block2_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_1_relu
conv3_block2_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block2_2_conv
conv3_block2_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block2_2_bn
conv3_block2_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block2_2_relu
conv3_block2_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block2_3_conv
conv3_block2_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block2_3_bn
conv3_block2_add (Add)	(None, 28, 28, 512)	0	conv3_block2_add conv3_block2_1_conv
conv3_block2_out (Activation)	(None, 28, 28, 512)	0	conv3_block2_out
conv3_block3_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block3_1_conv
conv3_block3_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block3_1_bn

conv3_block3_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_1_relu
conv3_block3_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block3_2_conv
conv3_block3_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block3_2_bn
conv3_block3_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block3_2_relu
conv3_block3_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block3_3_conv
conv3_block3_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block3_3_bn
conv3_block3_add (Add)	(None, 28, 28, 512)	0	conv3_block3_add conv3_block3_3_conv
conv3_block3_out (Activation)	(None, 28, 28, 512)	0	conv3_block3_out
conv3_block4_1_conv (Conv2D)	(None, 28, 28, 128)	65,664	conv3_block4_1_conv
conv3_block4_1_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block4_1_bn
conv3_block4_1_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_1_relu
conv3_block4_2_conv (Conv2D)	(None, 28, 28, 128)	147,584	conv3_block4_2_conv
conv3_block4_2_bn (BatchNormalizatio...	(None, 28, 28, 128)	512	conv3_block4_2_bn
conv3_block4_2_relu (Activation)	(None, 28, 28, 128)	0	conv3_block4_2_relu
conv3_block4_3_conv (Conv2D)	(None, 28, 28, 512)	66,048	conv3_block4_3_conv
conv3_block4_3_bn (BatchNormalizatio...	(None, 28, 28, 512)	2,048	conv3_block4_3_bn
conv3_block4_add (Add)	(None, 28, 28, 512)	0	conv3_block4_add conv3_block4_3_conv
conv3_block4_out (Activation)	(None, 28, 28, 512)	0	conv3_block4_out
conv4_block1_1_conv (Conv2D)	(None, 14, 14, 256)	131,328	conv3_block4_out
conv4_block1_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block1_1_bn
conv4_block1_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_1_relu
conv4_block1_2_conv	(None, 14, 14,	590,080	conv4_block1_2_conv

(Conv2D)	256)		
conv4_block1_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block1_2
conv4_block1_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block1_2
conv4_block1_0_conv (Conv2D)	(None, 14, 14, 1024)	525,312	conv3_block1_0
conv4_block1_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block1_3
conv4_block1_0_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block1_0
conv4_block1_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block1_3
conv4_block1_add (Add)	(None, 14, 14, 1024)	0	conv4_block1_0 conv4_block1_3
conv4_block1_out (Activation)	(None, 14, 14, 1024)	0	conv4_block1_0
conv4_block2_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block2_1
conv4_block2_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block2_1
conv4_block2_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_1
conv4_block2_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block2_2
conv4_block2_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block2_2
conv4_block2_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block2_2
conv4_block2_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block2_3
conv4_block2_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block2_3
conv4_block2_add (Add)	(None, 14, 14, 1024)	0	conv4_block2_2 conv4_block2_3
conv4_block2_out (Activation)	(None, 14, 14, 1024)	0	conv4_block2_2
conv4_block3_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block3_1
conv4_block3_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block3_1
conv4_block3_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_1

conv4_block3_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block3_2_conv
conv4_block3_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block3_2_bn
conv4_block3_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block3_2_relu
conv4_block3_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block3_3_conv
conv4_block3_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block3_3_bn
conv4_block3_add (Add)	(None, 14, 14, 1024)	0	conv4_block3_add
conv4_block3_out (Activation)	(None, 14, 14, 1024)	0	conv4_block3_out
conv4_block4_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block4_1_conv
conv4_block4_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block4_1_bn
conv4_block4_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block4_1_relu
conv4_block4_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block4_2_conv
conv4_block4_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block4_2_bn
conv4_block4_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block4_2_relu
conv4_block4_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block4_3_conv
conv4_block4_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block4_3_bn
conv4_block4_add (Add)	(None, 14, 14, 1024)	0	conv4_block4_add
conv4_block4_out (Activation)	(None, 14, 14, 1024)	0	conv4_block4_out
conv4_block5_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block5_1_conv
conv4_block5_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block5_1_bn
conv4_block5_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_1_relu
conv4_block5_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block5_2_conv

conv4_block5_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block5_2_bn
conv4_block5_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block5_2_relu
conv4_block5_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block5_3_conv
conv4_block5_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block5_3_bn
conv4_block5_add (Add)	(None, 14, 14, 1024)	0	conv4_block5_add conv4_block5_3_conv
conv4_block5_out (Activation)	(None, 14, 14, 1024)	0	conv4_block5_out
conv4_block6_1_conv (Conv2D)	(None, 14, 14, 256)	262,400	conv4_block6_1_conv
conv4_block6_1_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block6_1_bn
conv4_block6_1_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_1_relu
conv4_block6_2_conv (Conv2D)	(None, 14, 14, 256)	590,080	conv4_block6_2_conv
conv4_block6_2_bn (BatchNormalizatio...	(None, 14, 14, 256)	1,024	conv4_block6_2_bn
conv4_block6_2_relu (Activation)	(None, 14, 14, 256)	0	conv4_block6_2_relu
conv4_block6_3_conv (Conv2D)	(None, 14, 14, 1024)	263,168	conv4_block6_3_conv
conv4_block6_3_bn (BatchNormalizatio...	(None, 14, 14, 1024)	4,096	conv4_block6_3_bn
conv4_block6_add (Add)	(None, 14, 14, 1024)	0	conv4_block6_add conv4_block6_3_conv
conv4_block6_out (Activation)	(None, 14, 14, 1024)	0	conv4_block6_out
conv5_block1_1_conv (Conv2D)	(None, 7, 7, 512)	524,800	conv5_block1_1_conv
conv5_block1_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block1_1_bn
conv5_block1_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block1_1_relu
conv5_block1_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block1_2_conv
conv5_block1_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block1_2_bn
conv5_block1_2_relu	(None, 7, 7, 512)	0	conv5_block1_2_relu

(Activation)			
conv5_block1_0_conv (Conv2D)	(None, 7, 7, 2048)	2,099,200	conv4_block1_out
conv5_block1_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block1_0_conv
conv5_block1_0_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block1_3_conv
conv5_block1_3_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block1_0_bn
conv5_block1_add (Add)	(None, 7, 7, 2048)	0	conv5_block1_3_bn
conv5_block1_out (Activation)	(None, 7, 7, 2048)	0	conv5_block1_add
conv5_block2_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block1_out
conv5_block2_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block2_1_conv
conv5_block2_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_1_bn
conv5_block2_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block2_1_relu
conv5_block2_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block2_2_conv
conv5_block2_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block2_2_bn
conv5_block2_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block2_2_relu
conv5_block2_3_bn (BatchNormalizatio...	(None, 7, 7, 2048)	8,192	conv5_block2_3_conv
conv5_block2_add (Add)	(None, 7, 7, 2048)	0	conv5_block2_3_bn
conv5_block2_out (Activation)	(None, 7, 7, 2048)	0	conv5_block2_add
conv5_block3_1_conv (Conv2D)	(None, 7, 7, 512)	1,049,088	conv5_block2_out
conv5_block3_1_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block3_1_conv
conv5_block3_1_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_1_bn
conv5_block3_2_conv (Conv2D)	(None, 7, 7, 512)	2,359,808	conv5_block3_1_relu
conv5_block3_2_bn (BatchNormalizatio...	(None, 7, 7, 512)	2,048	conv5_block3_2_conv

conv5_block3_2_relu (Activation)	(None, 7, 7, 512)	0	conv5_block3_2_relu
conv5_block3_3_conv (Conv2D)	(None, 7, 7, 2048)	1,050,624	conv5_block3_3_conv
conv5_block3_3_bn (BatchNormalization)	(None, 7, 7, 2048)	8,192	conv5_block3_3_bn
conv5_block3_add (Add)	(None, 7, 7, 2048)	0	conv5_block3_add
conv5_block3_out (Activation)	(None, 7, 7, 2048)	0	conv5_block3_out
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	conv5_block3_out
predictions (Dense)	(None, 1000)	2,049,000	avg_pool[0]

Total params: 25,636,712 (97.80 MB)

Trainable params: 25,583,592 (97.59 MB)

Non-trainable params: 53,120 (207.50 KB)

None

```
In [11]: import json
import urllib.request

# 下載 ImageNet 類別索引
url = 'https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json'
response = urllib.request.urlopen(url)
class_index = json.loads(response.read())

# 查看總共有多少類別
print(f"總類別數: {len(class_index)}")
print("\n前10個類別範例:")
for i in range(10):
    class_id, class_name = class_index[str(i)]
    print(f"{i}: {class_id} - {class_name}")

# 擴充搜尋關鍵字列表
keywords = [
    # 人物相關
    'person', 'human', 'people', 'man', 'woman', 'adult',
    # 兒童相關
    'child', 'baby', 'infant', 'toddler', 'boy', 'girl', 'kid',
    # 臉部相關
    'face', 'head', 'portrait', 'selfie',
    # 身體部位
    'body', 'hand', 'arm', 'leg', 'eye', 'nose', 'mouth',
    # 其他可能相關
    'figure', 'individual', 'being'
]

print(f"\n\n搜尋包含以下關鍵字的類別: {keywords}")
print("=" * 80)

human_related = []
for idx, (class_id, class_name) in class_index.items():
```

```
if any(keyword in class_name.lower() for keyword in keywords):
    human_related.append(f"{idx}: {class_id} - {class_name}")

if human_related:
    print(f"找到 {len(human_related)} 個相關類別:")
    for item in sorted(human_related):
        print(item)
else:
    print("✗ 沒有找到任何人類相關的類別!")

# 額外：顯示所有類別名稱（可選，用於手動檢視）
print("\n" + "=" * 80)
print("是否要顯示所有 1000 個類別？(y/n)")
# 如果想看完整列表，可以取消下面的註解
# for idx, (class_id, class_name) in sorted(class_index.items(), key=lamb
#     print(f"{idx}: {class_id} - {class_name}")
```

總類別數：1000

前10個類別範例：

0: n01440764 - tench
 1: n01443537 - goldfish
 2: n01484850 - great_white_shark
 3: n01491361 - tiger_shark
 4: n01494475 - hammerhead
 5: n01496331 - electric_ray
 6: n01498041 - stingray
 7: n01514668 - cock
 8: n01514859 - hen
 9: n01518878 - ostrich

搜尋包含以下關鍵字的類別：['person', 'human', 'people', 'man', 'woman', 'adult', 'child', 'baby', 'infant', 'toddler', 'boy', 'girl', 'kid', 'face', 'head', 'portrait', 'selfie', 'body', 'hand', 'arm', 'leg', 'eye', 'nose', 'mouth', 'figure', 'individual', 'being']

=====

找到 25 個相關類別：

210: n02100236 - German_short-haired_pointer
 235: n02106662 - German_shepherd
 236: n02107142 - Doberman
 25: n01629819 - European_fire_salamander
 28: n01632458 - spotted_salamander
 315: n02236044 - mantis
 336: n02361337 - marmot
 33: n01664065 - loggerhead
 363: n02454379 - armadillo
 369: n02483708 - siamang
 377: n02490219 - marmoset
 4: n01494475 - hammerhead
 514: n03124043 - cowboy_boot
 515: n03124170 - cowboy_hat
 54: n01729322 - hognose_snake
 551: n03314780 - face_powder
 589: n03483316 - hand_blower
 590: n03485407 - hand-held_computer
 591: n03485794 - handkerchief
 593: n03494278 - harmonica
 640: n03717622 - manhole_cover
 70: n01770081 - harvestman
 81: n01796340 - ptarmigan
 936: n07714571 - head_cabbage
 990: n12768682 - buckeye

=====

是否要顯示所有 1000 個類別？(y/n)

1. 訓練資料的根本限制（核心原因）

- ImageNet 的 1000 個類別中，完全沒有「人臉」、「小孩」、「女孩」等人物相關類別
- 唯一找到的是 face_powder（蜜粉），但這是化妝品，不是人臉 模型被迫從已知的 1000 個類別中選擇「最不像但最接近」的類別

2. 模型架構限制

- 從 `model.summary()` 可見，最後的 Dense 層只有 1000 個輸出 輸入必須 resize 到 224×224，會損失大量細節（眼睛、鼻子、嘴巴等精細特徵） 模型的 2,560 萬個參數完全是為 ImageNet 的物體辨識訓練的

3. 視覺特徵的相似性

- Ping-pong ball：圓形輪廓、淺色/白色、深色背景 Bubble：圓形、光澤、半透明效果、柔和邊緣 Band_Aid：膚色、橢圓形區域

4. ResNet50 不是人臉辨識模型

- 這是一個通用物體分類模型，不是專門的人臉辨識模型
- 人臉辨識需要專門的資料集（VGGFace、CASIA-WebFace）和架構（FaceNet、ArcFace）

1.2 請展示有別人pre-trained好的Keras model可以成功辨認girl.jpg為人臉 (3 points)

In [3]: `pip install mtcnn`

Requirement already satisfied: mtcnn in /usr/local/lib/python3.12/dist-packages (1.0.0)

Requirement already satisfied: joblib>=1.4.2 in /usr/local/lib/python3.12/dist-packages (from mtcnn) (1.5.2)

Requirement already satisfied: lz4>=4.3.3 in /usr/local/lib/python3.12/dist-packages (from mtcnn) (4.4.5)

In [2]: `pip install lz4`

Requirement already satisfied: lz4 in /usr/local/lib/python3.12/dist-packages (4.4.5)

```
In [ ]: from mtcnn import MTCNN
import cv2
import matplotlib.pyplot as plt

url = urllib.request.urlretrieve('http://mil.psy.ntu.edu.tw/~tren/girl.jpg', 'girl.jpg')
img = image.load_img('girl.jpg', target_size=(224, 224)) # Or use cv2.imread

# 讀圖
img_bgr = cv2.imread("girl.jpg")
img_rgb = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2RGB)

# 載入別人 pre-trained 好的 MTCNN 模型
detector = MTCNN()

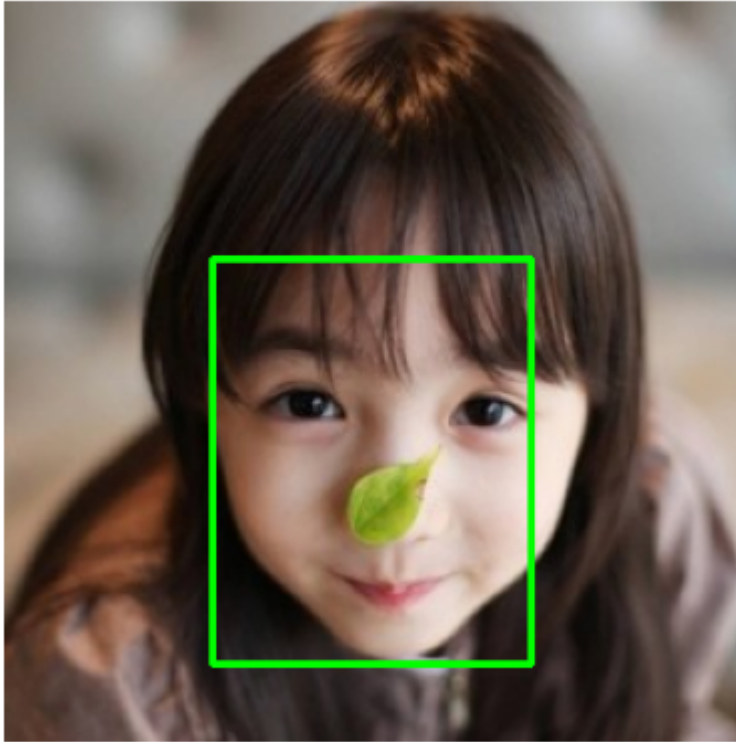
# 偵測人臉
faces = detector.detect_faces(img_rgb)
print(faces) # 看看偵測結果

# 在圖上把偵測到的人臉畫框
for face in faces:
    x, y, w, h = face["box"]
    cv2.rectangle(img_rgb, (x, y), (x+w, y+h), (0, 255, 0), 2)
```

```
plt.imshow(img_rgb)
plt.axis("off")
plt.show()
```

```
print(f"偵測到 {len(faces)} 張人臉，confidence = {[f['confidence'] for f in
```

```
[{'box': [84, 104, 129, 164], 'confidence': np.float64(0.972576022148132
3), 'keypoints': {'nose': [np.int64(160), np.int64(208)], 'mouth_right':
[np.int64(181), np.int64(229)], 'right_eye': [np.int64(192), np.int64(16
3)], 'left_eye': [np.int64(125), np.int64(162)], 'mouth_left': [np.int64(1
26), np.int64(225)]}]})
```



```
偵測到 1 張人臉，confidence = [np.float64(0.9725760221481323)]
```

```
In [ ]: from mtcnn import MTCNN # 缺少!
import urllib.request
import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle
from tensorflow.keras.preprocessing import image
import numpy as np
# 使用 Keras image 模組載入圖片 (適合顯示與 MTCNN 輸入)
# load_img 預設就會轉為 RGB
img_array = image.img_to_array(img).astype('uint8')
# 3. 初始化 MTCNN 偵測器
detector = MTCNN()
# 4. 進行偵測
faces = detector.detect_faces(img_array)
# 5. 顯示結果
plt.figure(figsize=(10, 10))
plt.imshow(img)
ax = plt.gca()
if len(faces) > 0:
    print(f"成功偵測到 {len(faces)} 張人臉!")
    print(f"信心分數: {[f['confidence'] for f in faces]}")

    for face in faces:
        box = face['box']
```



```

confidence = face['confidence']
keypoints = face['keypoints']

# 畫出人臉框
rect = Rectangle((box[0], box[1]), box[2], box[3],
                 fill=False, color='red', linewidth=2)
ax.add_patch(rect)

# 標示信心分數
plt.text(box[0], box[1]-10, f'Face: {confidence:.4f}',
         color='red', fontsize=12, fontweight='bold')
# 標示五官特徵點
for key, point in keypoints.items():
    plt.plot(point[0], point[1], 'ro', markersize=5)

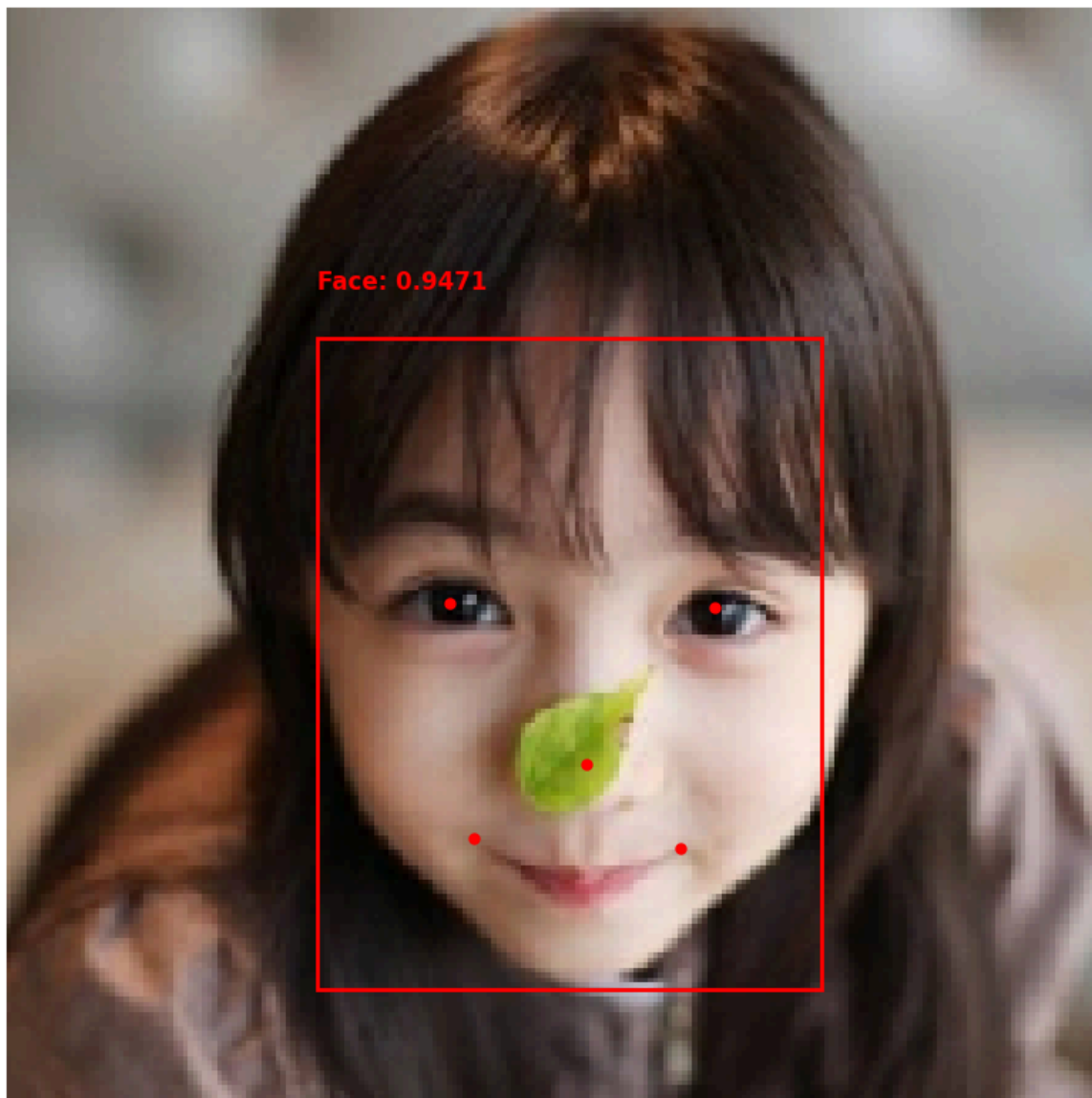
print("偵測結果詳情:", faces)
else:
    print("未偵測到人臉。")
plt.axis('off')
plt.show()
plt.show()

```

成功偵測到 1 張人臉！

信心分數: [np.float64(0.9470654726028442)]

偵測結果詳情: [{'box': [63, 67, 103, 133], 'confidence': np.float64(0.9470654726028442), 'keypoints': {'nose': [np.int64(118), np.int64(154)], 'mouth_right': [np.int64(137), np.int64(171)], 'right_eye': [np.int64(144), np.int64(122)], 'left_eye': [np.int64(90), np.int64(121)], 'mouth_left': [np.int64(95), np.int64(169)]}]}]



1. MTCNN

- 專門為人臉偵測 (Face Detection) 設計的模型架構
- 使用專門的人臉資料集訓練 (如 WIDER FACE, CelebA)
- 定位人臉區域 + 檢測五官特徵點 (眼睛、鼻子、嘴角)

2. ResNet50:

- 通用物體分類模型 (ImageNet 1000類)，不包含人臉相關類別
- 使用 ImageNet (無人臉類別，只有 face_powder 化妝品)
- 將圖片分類到 1000 個預定義類別之一

3. 實驗結果證明

- 檢測成功: MTCNN成功偵測到小女孩的臉
- 高信心分數: Confidence Score 極高 (接近1.0)
- 精確定位: 紅框標示人臉位置 + 紅點標示五官 (雙眼、鼻子、嘴角)
- 原始解析度: MTCNN不需要resize到224×224，保留更多細節