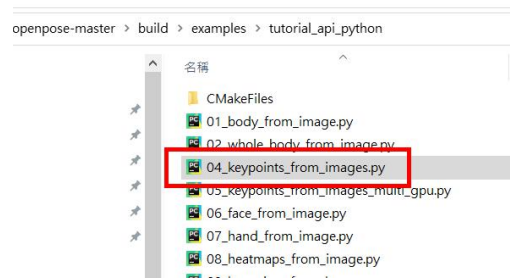


openpose\_Python\_api\_批量圖片下載調用篇  
開啟 04\_keypoints\_from\_images.py 檔：



檔案路徑讀取位置修改：

```
# Flags
# parser.add_argument("--image_dir", default=r"請輸入自己的圖片資料夾讀取路徑" help="")
parser = argparse.ArgumentParser()
parser.add_argument("--image_dir", default=r"C:\Users\user\Desktop\input_test" help="Process a direct")
parser.add_argument("--no_display", default=False, help="Enable to disable the visual display.")
args = parser.parse_known_args()
```

param 調整：

如果遇到 out of memory 可以調整參數 param["net\_resolution"] = "-1x160"

param["model\_pose"] = "COCO" 切換 COCO 模式只取 18 個節點(少了 7 個腳步的節點)

```
# Custom Params (refer to include/openpose/flags.hpp for more parameters)
params = dict()
params["model_folder"] = "../../../models/"
params["net_resolution"] = "-1x160"
params["model_pose"] = "COCO"
```

對每張圖做等比例縮放的方式：

調整 `scale_percent = 10` 的值，可以調整成不同比例

```
# Process and display images 柏翰_等比例縮放圖片：
# i = 1
for imagePath in imagePaths:
    datum = op.Datum()
    # imageToProcess = cv2.imread(imagePath)
    pics = cv2.imread(imagePath)
    scale_percent = 10
    # print(pics.shape)
    # width = pics.shape[1]
    width = pics.shape[1]
    height = pics.shape[0]
    width = int(width * scale_percent / 100)
    height = int(height * scale_percent / 100)
    dim = (width, height)
    print(dim)
    imageToProcess = cv2.resize(pics, dim, interpolation=cv2.INTER_CUBIC)
    datum.cvInputData = imageToProcess
    opWrapper.emplaceAndPop(op.VectorDatum([datum]))
```

承上每張圖的迴圈：

(1) `keypoint_np_x_y_str`：取出每張圖各節點的 x,y 的 np.array

(2) `pic_fn`：取出每張圖的檔名：

```
# 取出每張圖各節點的x,y,score 的np.array
keypoint_np_x_y_score_str = str(datum.poseKeypoints)
# 取出每張圖各節點的x,y 的np.array
keypoint_np_x_y_str = str(datum.poseKeypoints[0][:, [0, 1]])
# 取出每張圖的檔名：
path_list = imagePath.split("\\")
pic_fn_type = path_list[-1]
pic_fn = pic_fn_type.split(".")[0]
print("圖片檔名:", pic_fn)
```

檔案路徑存取路徑修改：

承上每張圖的迴圈：

store\_fn(原始檔名篇)：

base =base：為存檔路徑，請改成自己的目標位置

```
# Store_fn(原始檔名)
# base：為存檔路徑，請改成自己的目標位置
base = r"C:\Users\user\Desktop\save_tran_pics"
if not os.path.exists(base):
    os.makedirs(base)
# 將x,y的骨架依檔案名稱存成json檔
fn = base + "/" + pic_fn + ".json"
with open(fn, "w", encoding="utf-8") as f:
    f.write(keypoint no x y str)
# 將有骨架圖的圖片依檔案名稱存檔
fn = base + "/" + pic_fn + ".jpg"
cv2.imwrite(fn, datum.cvOutputData)
```

imshow 圖片：

key = cv2.waitKey(15) 預設，會連續開完圖片才停止

key = cv2.waitKey(0) 調整 waitKey(0)，需要按任一鍵才會跳下一張圖片

```
if not args[0].no_display:
    cv2.imshow("OpenPose 1.7.0 - Tutorial Python API", datum.cvOutputData)
    key = cv2.waitKey(15)
    # 圖片跟圖片間會暫停，案任一件下一頁
    # key = cv2.waitKey(0)
    if key == 27: break
```

遇到'NoneType' object is not subscriptable 的 bug：

因為圖片人物太小沒有產生骨架，`print(keypoint_np_x_y_score_str)` 印出 None  
導致 `keypoint_np_x_y_str = str(datum.poseKeypoints[0][:,[0, 1]])` 取 np.array 的列的時候造成 'NoneType' error。

```
111
112     # 取出每張圖各節點的x,y,score 的np.array
113     keypoint_np_x_y_score_str = str(datum.poseKeypoints)
114     print("keypoint_np_x_y_score_str:", keypoint_np_x_y_score_str)
115
116     # 取出每張圖各節點的x,y 的np.array
117     print("where is bug")
118     keypoint_np_x_y_str = str(datum.poseKeypoints[0][:,[0, 1]])
119     print("where is bug")
120
121     # 取出每張圖的檔名：
122     filename = os.path.basename(imagePath)
123     try:
124         for imagePath in imagePaths:
125             datum = op.Datum()
126             # imageToProcess = cv2.imread(imagePath)
127             pics = cv2.imread(imagePath)
128             scale_percent = 10
129             # print(pics.shape)
130             # width = pics.shape[1]
131             width = pics.shape[1]
132             height = pics.shape[0]
133             width = int(width * scale_percent / 100)
134             height = int(height * scale_percent / 100)
135             dim = (width, height)
136             print(dim)
137             imageToProcess = cv2.resize(pics, dim, interpolation=cv2.INTER_CUBIC)
138             datum.cvInputData = imageToProcess
139             opWrapper.emplaceAndPop(op.VectorDatum([datum]))
140             wrong_pic_fn_list.append(filename)
141             count += 1
142     except Exception as e:
143         print(e)
144         wrong_pic_fn_list.append(filename)
145         count += 1
146     print("Process finished with exit code -1")
```

04\_keypoints\_from\_images x

Starting OpenPose Python Wrapper...

Auto-detecting all available GPUs... Detected 1 GPU(s), using 1 of them starting at GPU 0.

(400, 600)

keypoint\_np\_x\_y\_score\_str: None

where is bug

'NoneType' object is not subscriptable

Process finished with exit code -1

解決並抓出沒產生骨架的圖片：

先在迴圈外把設 `count`

`wrong_pic_fn_list = []`

```
count = 0
wrong_pic_fn_list = []
for imagePath in imagePaths:
    datum = op.Datum()
    # imageToProcess = cv2.imread(imagePath)
    pics = cv2.imread(imagePath)
    scale_percent = 10
    # print(pics.shape)
    # width = pics.shape[1]
    width = pics.shape[1]
    height = pics.shape[0]
    width = int(width * scale_percent / 100)
    height = int(height * scale_percent / 100)
    dim = (width, height)
    print(dim)
    imageToProcess = cv2.resize(pics, dim, interpolation=cv2.INTER_CUBIC)
    datum.cvInputData = imageToProcess
    opWrapper.emplaceAndPop(op.VectorDatum([datum]))
    wrong_pic_fn_list.append(filename)
    count += 1
except Exception as e:
    print(e)
    wrong_pic_fn_list.append(filename)
    count += 1
print("Process finished with exit code -1")
```

把[取每張圖的檔名]coding 往上放

因為要從 `except` 裡面把錯誤的照片 `append` 出來，最後要把所有錯誤的照片抓出來並排除要建的標籤，以免放入訓練資料集的時候有誤

`try:`

如果 `keypoint_np_x_y_score_str == None`

`keypoint_np_x_y_str` 會跳 `'NoneType'`錯誤

並走入 `except` 裡面

走過 `except` 的事件

透過 `continue` 回到 `for imagePath in imagePaths:` 迴圈，往下一張圖片繼續進行

如果圖片正確不經過 `except`。

```
# 取出每張圖的檔名：
path_list = imagePath.split("\\")
pic_fn_type = path_list[-1]
pic_fn = pic_fn_type.split(".")[0]
print("圖片檔名:", pic_fn)
# print(type(pic_fn))

try:
    # 取出每張圖各節點的x,y,score 的np.array
    keypoint_np_x_y_score_str = str(datum.poseKeypoints)
    # 取出每張圖各節點的x,y 的np.array
    keypoint_np_x_y_str = str(datum.poseKeypoints[0][:, [0, 1]])
    # print("keypoint_np_x_y_score_str:", keypoint_np_x_y_score_str)
except TypeError as e:
    print(e)
    print('這是張不合格照片:', pic_fn)
    count += 1
    print('第', count, '次錯誤照片')
    wrong_pic_fn_list.append(pic_fn)
    continue
```

在走完每一張照片的迴圈印出 `wrong_pic_fn_list`

```
if not args[0].no_display:
    cv2.imshow("OpenPose 1.7.0 - Tutorial Python API", datum.cvOutputData)
    key = cv2.waitKey(15)
    # 圖片跟圖片間會暫停，案任一件下一頁
    # key = cv2.waitKey(0)
    if key == 27: break

end = time.time()
print("OpenPose demo successfully finished. Total time: " + str(end - start) + " seconds")
print("wrong_pic_fn_list:", wrong_pic_fn_list)
```

最後當迴圈走完，印出 `wrong_pic_fn_list`：

- (1)將出現在清單裡的圖片刪除 (2)excerl 標籤也去除
- 因為此圖片的人物太小，無法產生出骨架。

```
OpenPose demo successfully finished. Total time: 113.56067824363708 seconds
wrong_pic_fn_list: ['1143']

Process finished with exit code 0
```

