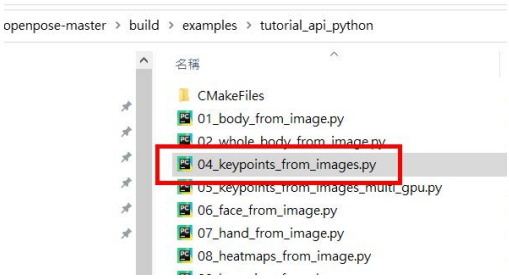
`

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 memeory 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，可以條整成不同比例

```
# 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.shpae[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.arrary

(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.arrary
keypoint_np_x_y_score_str = str(datum.poseKeypoints)
# 取出每張圖各節點的x,y 的np.arrary
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

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