

Training Part

1/ Dataset

The datasets used in this project are Wildtrack and MultiviewX, which can be downloaded using the following links

Wildtrack:

http://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/Wildtrack/Wildtrack_dataset_full.zip

MultiviewX:

https://onedrive.live.com/?authkey=%21ABoYWoW6dtDI_ZA&id=DFB1B9D32643ECDC%2182813&cid=DFB1B9D32643ECDC&parId=root&parQt=sharedby&o=OneUp

2/ Dataset preprocessing (Wildtrack) Step 6: One-click processing

In order to train the model, a series of operations need to be performed on the data set. All data processing codes are in the **wild&multi_scripts** folder.

1) read_json.py

Modify the corresponding image path and annotation path, and use the video frame image and pedestrian frame position information to obtain pedestrian image data. Each subfolder under the processed folder represents the pedestrian ID, and the image names in the folder correspond to the time number and view number respectively.

In the Wildtrack dataset, pedestrian IDs need to be reranked. The reason is that there are more than 1200 pedestrian IDs in the annotations, but there are less than 300 different pedestrians in reality. The classification loss used in MGN will set the classification head based on the number of pedestrian categories, so the pedestrian IDs need to be limited to 0~296. Rename using the ID mapping relationship in data.json.

Change the following addresses to correspond to Image_subsets, annotation_positions and output folder paths respectively

```
24 images = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/Image_subsets/'
25 source = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/annotations_positions'
26 target = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed'
27
```

This section uncomments the wildtrack section to conform to the naming conventions of the wildtrack dataset.

```
51 # 用于wildtrack数据集
52 s_path = images+'C'+str(camera_num)+'/'+fil[:8]+' .png'
53 t_path = target+id+'/'+'fil[4:8]+'_'+str(camera_num)+' .png'
54
55 # 用于MultiviewX数据集
56 s_path = images+'C'+str(camera_num)+'/'+fil[1:5]+' .png'
57 t_path = target+id+'/'+'fil[1:5]+'_'+str(camera_num)+' .png'
58
```

2) rename_png.py

Process the data in the processed folder. Name the final pedestrian image as xxxx_cy_tzzzz.png, where x, y, and z correspond to the pedestrian ID, view number, and time number, respectively.

Here processed corresponds to the output folder of the previous step

```
5 source = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed'
6 target = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed_rename'
7
```

3) rerank_id.py

Process the data in the **split/full_before_rerank/** folder and rename their character ids according to the mapping relationship in data.json.

The renamed results will be output to the split/full/ folder (because some ids in data.json have no mapping relationship, such as 1200, 0623, etc., so after the rerank conversion, our data will be reduced by about 1100)

Processed_folder corresponds to the output folder of the previous step

```
# processed文件夹的路径
processed_folder = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed_rename'

# 输出文件夹的路径
output_folder = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed_rename_rerank'
```

4) camera_split.py

Create subfolders based on different perspectives and put the pictures into them accordingly

Source corresponds to the output folder of the previous step

```
source = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/processed_rename_rerank'
target = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/camera_split'
```

5) split_rerank.py

For folders with different perspectives, separate the train test val set according to the ratio of 0.45 0.45 0.1

The source_folder here corresponds to the output folder of the previous step

```
41 source_folder = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/camera_split'
42 destination_folder = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/Wildtrack_split_rerank'
```

This is the last step of data processing. We will rename the generated folder to Wildtrack_split_rerank (Note that the W is capitalized)

6) one_step_wildtrack_split_rerank.py

To simplify the operation, we have integrated a separate program to process the data in one click to generate Wildtrack_split_rerank

We only need to change the path here to correspond to the Wildtrack_dataset folder, and we can generate the processed Wildtrack_split_rerank dataset in this directory.

```
1 import json
2 import os
3 import shutil
4 import random
5 from math import floor
6 from PIL import Image
7
8 # 输入你Wildtrack_dataset的路径
9 base = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/'
10
11
```

3/ Train the model for each camera (using Wildtrack as an example)

1) Upload the Wildtrack_split7 to the codes_to_orange/datasets folder.

2) Change the parameters of the following files

codes_to_orange/start_wildsplit.sh

Modify to train different sub-models. Here, the WildSplit7 folder is used, indicating that the sub-model of the 7th camera is trained.

```
$ start_wildsplit.sh
1 # export CUDA_VISIBLE_DEVICES='2'
2
3 CUDA_VISIBLE_DEVICES=2
4 python3 tools/train_net.py --config-file ./configs/WildSplit7/mgn_R50-ibn.yml #MODE
5
```

codes_to_orange/fastreid/data/common.py

Modified to correspond to different datasets (returns 297 for Wildtrack dataset)

```
52 @property
53 def num_classes(self):
54     # return len(self.pids)
55     # 对于wildsplit数据集, 确保num_classes = 297
56     return 297
57     # 对于multisplit数据集, 确保num_classes = 350
58     # return 350
```

codes_to_orange/fastreid/data/datasets/bases.py

Modified to correspond to different datasets (returns 297 for Wildtrack dataset)

```
87 def parse_data(self, data):
88     """Parses data list and returns the number of person IDs
89     and the number of camera views.
90     Args:
91         data (list): contains tuples of (img_path(s), pid, camid)
92     """
93     pids = set()
94     cams = set()
95     for info in data:
96         pids.add(info[1])
97         cams.add(info[2])
98     # return len(pids), len(cams)
99     # 对于wildsplit数据集, 确保num_pids = 297
100     return 297, len(cams)
101     # 对于multisplit数据集, 确保num_pids = 350
102     # return 350, len(cams)
```

3) Go back to the **codes_to_orange** root directory and use the command `./start_wildsplit.sh` to start the training program. Train different sub-models by changing the camera number in **start_wildsplit.sh**.

The trained model will be saved in **codes_to_orange/logs/wildsplit1** (the last number corresponds to the camera number 1-7)

Demo implementation part

(person No. 0134 as an example)

To implement the Demo, you need to complete the training and get the sub-model of each camera! !

1/ Data processing

1) Generate the **Wildtrack_by_pid_and_time** dataset using the **transform_wild_pid_and_time.py** file in the data processing folder.

Source selects the **Wildtrack_splited_rerank** dataset

```
34 source_directory = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_dataset/Wildtrack_splited_rerank'
35 target_directory = 'X:/COCO/Multicamera/Wildtrack/Wildtrack_dataset_full/Wildtrack_by_pid_and_time'
```

Move **Wildtrack_by_pid_and_time** to the **codes_to_orange/datasets** folder

2) Use **mv_files_transfer.ipynb** in **codes_to_orange/scripts** to build **gallerypid_mmoe** and **gallery** alternatives (here we take person 0134 as an example)

root selects **Wildtrack_by_pid_and_time** in dataset

In target, the number after gallery determines the person ID you want to query
time decides in which frames to search for people (select the frames where the people to be searched appear)

```
root = '../datasets/Wildtrack_by_pid_and_time/'
target = '../datasets/Orange_demo_wild/mv/gallery0134_mmoe'

files = os.listdir(root)
for file in files:
    time = file.split('.')[1]
    if 575 <= int(time) <= 1995:
        shutil.copytree(join(root, file), join(target, file))
```

After the above steps, the following two folders should appear in **datasets/Orange_demo_wild/mv**

```
✓ Orange_demo_wild
  > baseline
  ✓ mv
    > gallery0011
    > gallery0011_mmoe
    > gallery0081
    > gallery0081_mmoe
    > gallery0095
    > gallery0095_mmoe
    > gallery0113
    > gallery0113_mmoe
    > gallery0134
    > gallery0134_mmoe
```

Create a folder named **query0134** under the **mv** folder at the same level as gallery0134, then randomly select a folder **starting with 0134** in **gallery0134_mmoe** and move it to the **query0134** folder as the image to be queried.

```
\---query0134
  \---0134_t0870
    0134_c1_t0870.png
    0134_c2_t0870.png
    0134_c3_t0870.png
    0134_c6_t0870.png
```

2/ Calculate similarity distance

- 1) Modify the **mv_eval.py** file in **codes_to_orange/mtl** to adapt to the demo
name is changed to the **person ID** to be queried
Change eval_mode to 1

```
308 > if __name__ == "__main__":
309     args = default_argument_parser().parse_args()
310     dataset = 'wild'
311     is_origin_model = 0
312     eval_mode = 1 # 仅在orange_demo启用, 设置为1
313     name = '0134' # 仅在orange_demo下有意义
314     if dataset == 'wild':
315         root = 'datasets/Wildtrack_eval_rerank'
316         n_cam = 7
317     elif dataset == 'multi':
318         root = 'datasets/MultiviewX_eval'
319         n_cam = 6
```

Uncomment this code and save it in the path shown in the figure below. The character ID is in front of the folder name, and the video frame range we selected is after the underline, which needs to correspond to the video frame selected in step 1_2.

```
334     array_save_path = 'orange_demo/mv_video/0134_t0575to1995/list'
335     import numpy as np
336     # dist_flat = dist.flatten()
337     for idx, sub_dist in enumerate(dist[0]):
338         np.save(array_save_path + str(idx) + '.npy', sub_dist)
```

- 2) Modify the **preprocessor.py** folder in **codes_to_orange/mtl/utlis**

Modify the two addresses in the figure below. As in the previous step, the character ID is in front of the folder name, and the video frame range we selected is after the underline.

```
49 class PreprocessQuery(object):
50     def __init__(self, query, root=None, transform=None):
51         super(PreprocessQuery, self).__init__()
52         self.query = query
53
54         with open('orange_demo/mv_video/0134_t0575to1995/queries_paths.txt', 'a') as f:
55             for fname, img_names in query.items():
56                 f.write(str(fname))
57                 for img_name in img_names:
58                     f.write(str(img_name[2]))
59                 f.write('\n')
60
61         self.root = root
62         self.transform = transform
63         self.index_dict = {}
64         self.__create_index_dict__()
```

```

100 class PreprocessGallery(object):
101     def __init__(self, gallery, camid, root=None, transform=None):
102         super(PreprocessGallery, self).__init__()
103         self.gallery = gallery[camid]
104
105         with open('orange_demo/mv_video/0134_t0575to1995/galleries_paths.txt', 'a') as f:
106             f.write(str(fname) + '\n')
107
108         self.root = root
109         self.transform = transform
110         self.camid = camid

```

- 3) Run **python3 mtl/mv_eval.py** to calculate the similarity distance. The calculation results and related image paths will be stored in the folder **orange_demo/mv_video/0134_t0575to1995** (taking 0134 as an example)

It should be noted that during the calculation process, we will generate the `gallery_paths.txt` file and the `query_path.txt` file in the folder. Before running the program, we need to delete these two files, otherwise they will be generated repeatedly and cause errors.

The calculation results of the similarity distance will also be saved in the folder **orange_demo/mv_video/0134_t0575to1995**.

3/ Generate Video

Use the **mv_demo_process.ipynb** script in **codes_to_orange/scripts** and modify the **pid** to match the character id

```

[2]: import numpy as np
      from os.path import join
      from glob import glob
      import os

•[164]: pid = '0134'
        root = glob(join('../orange_demo/mv_video', pid + '*'))[0]
        root

[164]: '../orange_demo/mv_video/0236_t0575to1995'

[165]: datas = []
        for dist in sorted(glob(join(root, 'dist*'))):
            data = np.load(dist, allow_pickle=True)
            datas.append(data)

```

After the run is complete, it will save a **match.json** file in the **orange_demo/mv_video/0134_t0575to1995** folder.

Then use **demo_mv_picture.ipynb** in **codes_to_orange/scripts** to select the specified character in the scene.

Change the path here to match the match.json file generated in the previous step

```
[1]: import cv2
import os
import json
import numpy as np
import ast

# 读取JSON文件
json_file_path = 'codes_to_orange/codes_to_orange/orange_demo/mv_video/0134_t0575to1995/matches.json'
with open(json_file_path, 'r') as file:
    data = json.load(file)
    # print(type(data))
i = 0

# 初始化记录结果的变量
matched_results = []
```

Change the path here to match the r2o.json file in the scripts folder

```
]: import json
# 指定JSON文件路径
json_file_path = '/data02/zhangrunxiang/reid1/fast-reid/scripts/r2o.json'

# 加载JSON文件中的数据
with open(json_file_path, 'r') as file:
    json_data = json.load(file)
```

Change the path here to match the annotation_positions folder in the Wildtrack dataset

```
# 指定JSON文件目录路径
json_directory = '/data02/zhangrunxiang/reid1/datasets/Wildtrack_dataset_full/Wildtrack_dataset/annotations_positions/'

# 遍历matched_results, 提取所有的帧数信息
# frame_numbers_to_search = set(item[1] for item in matched_results)
desired_position = {} # 用于存储行人ID、帧数和视角为5的信息

for frame_number in frame_numbers:
    # 构建要查找的JSON文件名
    frame_number = str(frame_number)
    json_file_name = '0000' + frame_number + '.json'
    json_file_path = json_directory + json_file_name
```

Change the two paths here to match the path of **Image_subsets** in the **Wildtrack dataset** and the path where we want to save the framed images (here C6 represents the scene image taken by camera 6, and you can choose scene images from C1 to C7 as needed)

```
# 指定要处理的图像文件夹路径和保存路径
file_path = '/data02/zhangrunxiang/reid1/datasets/Wildtrack_dataset_full/wildtrack_dataset/Image_subsets/C6/'
save_path = '/data02/zhangrunxiang/reid1/datasets/wildtrack_dataset_full/Wildtrack_dataset/Image_subsets/C6/'

# 创建保存路径文件夹
os.makedirs(save_path, exist_ok=True)

start_frame = 700
end_frame = 900
step = 5
```

Finally, run the program to get the processed picture frame image

To generate a video from an image, you can use **pic2video.ipynb** in the scripts folder
Change the following path to match the framed image saved in the previous step

•[3]: # 图片文件夹路径

```
image_folder = '/data02/zhangrunxiang/reid1/datasets/wildtrack_dataset_full/Wildtrack_dataset/Image_subsets/C6/'
```

```
# 输出视频文件名, 保存到与图片文件夹相同位置
```

```
video_name = os.path.join(image_folder, image_folder.split('/')[-1] + '_output.mp4')
```

```
# 获取图片文件夹下所有图片文件名
```

```
images = [img for img in os.listdir(image_folder) if img.endswith(".png")]  
print(len(images))
```