DeepLearning Lab1: Dataset and DataLoader

學號:31383404 姓名:周彥宏

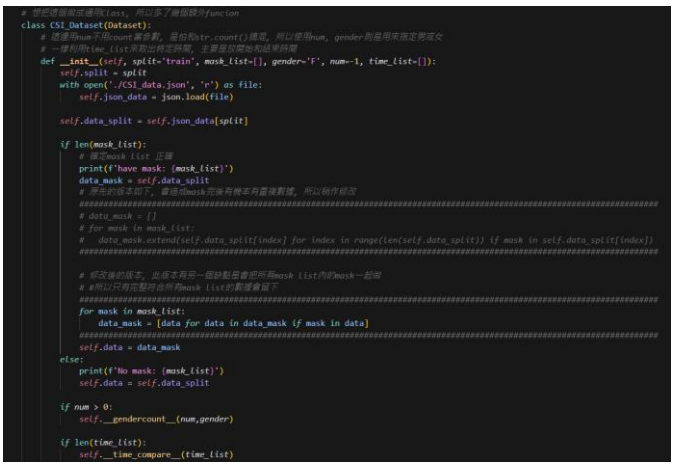
Github link

這次的 Lab 實作中,主要目的是想最大程度減小 rewrite Class CSI_DataSet 的次數,讓我在面對每個 requirement 都可以直接用已經寫好的**通用** Class 去做。

Step 1. 觀察 CSI_data.json

1. Dataset 中分成三個不同的資料集, 分別是"train", "val", "test".
2. 主要的 format 為:
   "CLASS_NAME/npy/THE_GENDER_AND_COUNT/POSITION/TIME/FILE_NAME"
3. 在"val" and "test"中, Gender and count 這欄只會出現**單一男性**, 或是**單一女性**, 這在 requirement 2,3,5 都很重要.
4. 只有在"train"內, 會出現"Env(i)"的 Class name, 其餘都是"val_set" or "test_set".
5. Time 中的 format 為 "date"_"time", eg. 240509_111208(2024/05/09 11:12:08), 這種 format 會讓我們在做 requirement 4,5 的時候很好判斷.
6. 承 5. 如果利用判斷數字"240509" or "111208", 會出現一種錯判是數字有機率會出現在後面的 filename, 在觀察時沒有觀察出一個特定的規律, 所以認定是 random number 所組成的 filename.

Step 2. Rewrite Class



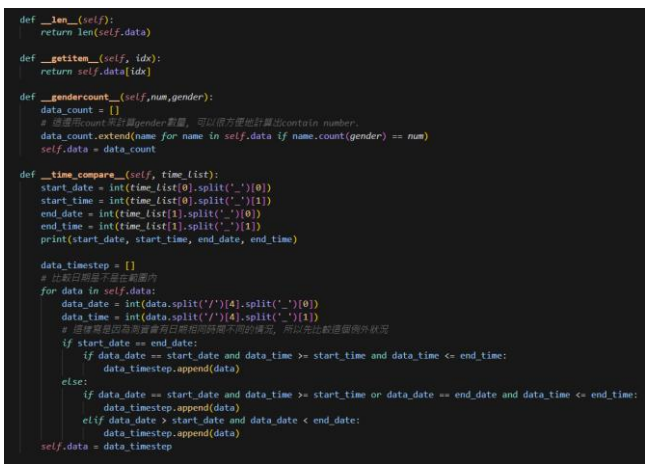
```
class CSI_Dataset(Dataset):
    def __init__(self, split='train', mask_list=[], gender='F', num=-1, time_list=[]):
        self.split = split
        with open('./CSI_data.json', 'r') as file:
            self.json_data = json.load(file)

        self.data_split = self.json_data[split]

        if len(mask_list):
            print(f'have mask: {mask_list}')
            data_mask = self.data_split
            for mask in mask_list:
                data_mask = [data for data in data_mask if mask in data]
            self.data = data_mask
        else:
            print(f'No mask: {mask_list}')
            self.data = self.data_split

        if num > 0:
            self.__gendercount__(num,gender)

        if len(time_list):
            self.__time_compare__(time_list)
```



```
    def __len__(self):
        return len(self.data)

    def __getitem__(self, idx):
        return self.data[idx]

    def __gendercount__(self,num,gender):
        data_count = []
        data_count.extend(name for name in self.data if name.count(gender) == num)
        self.data = data_count

    def __time_compare__(self, time_list):
        start_date = int(time_list[0].split('_')[0])
        start_time = int(time_list[0].split('_')[1])
        end_date = int(time_list[1].split('_')[0])
        end_time = int(time_list[1].split('_')[1])
        print(start_date, start_time, end_date, end_time)

        data_timestep = []
        for data in self.data:
            data_date = int(data.split('/')[4].split('_')[0])
            data_time = int(data.split('/')[4].split('_')[1])
            if start_date == end_date:
                if data_date == start_date and data_time >= start_time and data_time <= end_time:
                    data_timestep.append(data)
            else:
                if data_date == start_date and data_time >= start_time or data_date == end_date and data_time <= end_time:
                    data_timestep.append(data)
                elif data_date > start_date and data_date < end_date:
                    data_timestep.append(data)
        self.data = data_timestep
```

在設計 Class CSI_Dataset 時, 有先去查閱一下 torch dataset and dataloader 運作, 發現基本上 Dataset 都要在這個 Class 先處理完,就直接拿來做訓練, 所以我將所有 data clear 的 function 都放在這個 Class 內去做, 並讓它變成通用的.

以下會分幾個重點來說: (**BTW 我怕太多 PDF 放不下, 所以 code 都有寫註解**)

1. 在做 mask 時, 原先想法是每個符合 mask list 就利用 extend 丟進去 mask_data, 但是會造成 data 會有**重複情況**發生.eg. mask_list = ['F1', 'F2', 'F3']

2. 承 1. 之後修改成只有完整符合所有 mask_list 的 format 才算.
3. 接著是 gendercount，就是簡單去判斷 string 內，所含的字母”F” or “M”個數，就可以拿出我們需要的資料. 那這次資料集不會在其他地方出現”F” or “M”，所以就沒有做很嚴謹的處理.
4. Time compare 一樣是利用 time_list 傳入起始和結束時間，然後判斷時間有沒有在範圍內，這是可以解決我們剛剛講到 date or time number 會出現在 filename 中的問題.
5. 承 4. 我利用 spilt 方法把 format 分段，取出我想要的 date and time. 因為 requirement 中，會出現**開始結束的日期是同一天**，所以我有稍微修改一下判斷式，先判斷這個例外狀況. 接著判斷**相同日期不同時間**，再來才是**超過開始日期但是還未到結束時間**，這種只要判斷 date，不用判斷 time.

Step 3. Thinking Requirement

分別對每個 Requirement 做不同的 Constructor 傳入參數的設定.(相關 function 介紹都在 step 2).

Requirement 1. 因為只需要 Class name，所以只需要傳入 mask_list = [‘Env3’]，且這個 class 只有在”train”才有，所以 spilt = “train”.

Requirement 2. 因為要女生，先利用 mask_list = [‘F’]把單一男性的拿掉，在利用 gender = ‘F’, num = 2,來拿到需要的 data. 另外”val” and “test”只有單一性別，所以只看”train”.

Requirement 3. 同理於 Requirement 2.但不一樣的是只有單一女性，所以 mask_list = [“Female”]，然後需要看”val” and “test”，所以我採取分別將”train” “val” “test” 利用 spilt 取出後，在透過 output 合併.

```
dataset_train = CSI_Dataset(split='train', mask_list=['Female'])
dataset_val = CSI_Dataset(split='val', mask_list=['Female'])
dataset_test = CSI_Dataset(split='test', mask_list=['Female'])

print(len(dataset_train), len(dataset_val), len(dataset_test))
dataLoader_train = DataLoader(dataset_train, batch_size=len(dataset_train), shuffle=False)
dataLoader_val = DataLoader(dataset_val, batch_size=len(dataset_val), shuffle=False)
dataLoader_test = DataLoader(dataset_test, batch_size=len(dataset_test), shuffle=False)
```

```
have mask: ['Female']
have mask: ['Female']
have mask: ['Female']
229679 3622 4807
```

```
# combine all dataset to output, and check it contain all of "Female"
output = []
output.extend(next(iter(dataLoader_train)))
output.extend(next(iter(dataLoader_val)))
output.extend(next(iter(dataLoader_test)))
print(len(output))
output[233001:233301]
```

Requirement 4. 時間部分, “val” and “test”只有 240509 的資料，所以只看”train”,time_list = [“240506_181307”,”240507_232434”].

Requirement 5. 同理 1 and 2，只看”train”然後只有一個男性，所以只要看”Male(i)”，利用上述修改過的 mask 特性，直接設定 mask_list = [“Env3”, “5_posi”,”Male”]，由於這邊設定 Male 當 mask，所以不會有其他女性在內,因此不用設定 gender and num. time_list = ['240508_090000', '240508_110000'].