📘 Week 1 Homework: Dataset Splitting & Avoiding Data Leakage

Objective
The goal of this assignment is to help students understand the correct procedure for dataset splitting and the risks of data leakage. You are required to implement a custom dataset loader that splits the MNIST dataset into **training / validation / test sets** following standard practice, ensuring there is no data leakage.

Dataset Description

- The provided MNIST dataset is organized into **10 folders**, one for each digit label (0–9).
- The number of images in each folder may vary.
- Your program must read the images from these folders and assign the correct labels (0–9).

Requirements

1. **Custom Dataset Loader**
   - Implement a `dataset_loader.py` to complete the dataset splitting process.
   - **Do not use** built-in random splitting functions such as `torch.utils.data.random_split` or `sklearn.model_selection.train_test_split`.
2. **Splitting Ratio**
   - Split the dataset into three subsets:
     - Training set: **70%**
     - Validation set: **15%**
     - Test set: **15%**
   - Each class (0–9) must be split according to the same ratio to avoid class imbalance.
3. **Avoiding Data Leakage**
   - Splitting must be done **immediately after loading the data**. Do not perform normalization or other preprocessing before splitting.
   - The training, validation, and test sets must be completely independent—no duplicate images across splits.
4. **Output Format**
   - Your program should print the number of samples in each split.
   - Each split must be saved as a text file:
     - `train_list.txt`
     - `val_list.txt`
     - `test_list.txt`

Each line should follow the format:
`image_path label`

5. **Bonus (Optional)**
    - Implement a function `check_data_leakage()` to verify that no samples appear in more than one split.
    - Write a short explanation in `README.md` describing what data leakage is and how you avoided it.

Example Output (Class 0)
Suppose the folder `digit_0/` contains 100 images. With a 70/15/15 split:

- Train: 70 images
- Validation: 15 images
- Test: 15 images

Corresponding sample outputs in `train_list.txt`, `val_list.txt`, and `test_list.txt` might look like:

`train_list.txt` (excerpt)

```
digit_0/img_0001.png 0
digit_0/img_0002.png 0
...
digit_0/img_0070.png 0
digit_1/img_0001.png 1
digit_1/img_0002.png 1
...
digit_1/img_0070.png 1
...
digit_9/img_0070.png 9
```

`val_list.txt` (excerpt)

```
digit_0/img_0071.png 0
digit_0/img_0072.png 0
...
digit_0/img_0085.png 0
```

`test_list.txt` (excerpt)

```
digit_0/img_0086.png 0
```

```
digit_0/img_0087.png 0
...
digit_0/img_0100.png 0
```

In the output, `image_path` must match the actual file path, and `label` must be an integer (0–9).

At the end, your program should also print the split sizes to the terminal, for example:

```
Train set size: 42000
Validation set size: 9000
Test set size: 9000
```

Submission

- `dataset_loader.py`
- `train_list.txt`, `val_list.txt`, `test_list.txt`
- **PDF file**: screenshots of your code and a short explanation
- *(Optional)* `README.md`

Deadline
📅 Tuesday, Sep 9, 11:59 am