

Program Assignment #2

- **Due Date: Dec. 9 (Wed) 18:00**
- **Submission: Source code & Report file**

Design a handwritten character recognition system.

We provide an example code (*main.py*). You can modify *main.py* to make your own code.

1. Database

This data contains images of five labels; a, b, c, d, and e. Each image is 350×350 pixels. The data is arranged into five folders corresponding to their labels. The images are labeled 0.png, 1.png, and etc.

- The labels of samples are defined in the example code (*main.py*).
 - $a \Leftrightarrow 0 / b \Leftrightarrow 1 / c \Leftrightarrow 2 / d \Leftrightarrow 3 / e \Leftrightarrow 4$
- Hand-out data: 150 samples
 - 30 samples / each label
 - You can split this data into training set and validation set for robustness to the test (unseen) data. Because good performance in hand-out data can be the opposite in test data.
- Test (unseen) data for TA: 300 samples
 - 60 samples / each label
 - This data is not provided.

2. Preprocessing and feature extraction ('preprocess' function)

Implement your own feature extraction and preprocessing algorithm using images.

- You can augment the hand-out data by optional processing tasks.
- (Optional) Rotation: adjust the given characters to the same angle.
- (Optional) Thickness: adjust the given characters to the same thickness.
- If there are other preprocessing tasks that you think would be necessary, design and implement them. It is optional, you do not need to implement optional preprocessing tasks.
- Then, implement your own feature extraction method to distinguish each alphabet.
- In the example code (*main.py*), we just flatten the images.

3. Make a classifier ('classify' function)

Implement your own classification algorithm using your own features. You can use any tools. Rule-based algorithm is also possible. Submit the highest performance algorithm (source code) of your own coding.

- In the example code (*main.py*), we use KNN algorithm.

Precautions

1. Source code (including *main.py*) and report file (*report.docx*) that explain your preprocessing and classifier methods. If you do not refer our example code (*main.py*), explain how to run your algorithm
File name: StudentID.zip (or tar.gz) ex) 20201111.zip (or tar.gz)
2. **Your source code must include 'preprocess' and 'classify' functions.**
 - ✓ Submit only one of the best-performing algorithms. (It is impossible to output the best performance after testing with various algorithms in 'classify' function.)
3. Implement your algorithm using python3.

- A. You can use python external packages such as *numpy*, *librosa*, *skleran*, *scipy*, *cv2*, and *etc*.
- 4. TA will evaluate final performance of your algorithms using the hand-out data and test (unseen) data.
 - A. $\text{PA1 score} = 0.3 * (\text{Accuracy for hand-out data}) + 0.7 * (\text{Accuracy for test data})$