**Coding Tasks**

**1. CNN Implementation**

Description:

In this task, you are asked to develop a CNN model by giving the dataset below. (<https://mavsuta-my.sharepoint.com/:f:/g/personal/chaowei_wang_mavs_uta_edu/Eo6iRY2sJKdOp5V7xCmDAt8BcU7LoCuIE0k2ETqrxpM1-A?e=4RUuDa>)

There are two files in the dataset:

* data.csv is the training data. Column ACCX is collected from one channel of accelerator. Column Light indicate environment light condition.
* label.csv is the label for the training data (data.csv).
  + There are 55 labels, and 137500 rows in training data. Therefore, each label corresponds to 2500 rows of training data. In other words, the input size of one instance is 2500\*2.

You are free to choose any deep learning library you are familiar with. After finish training, please save the output as a file named “output.txt”. **The model’s performance is not important.** However, it is important the model runs with no error.

Deliverables:

* Source code. The code should be organized in a clear structure and easy to follow.
* The output files.
* A README.txt with necessary instructions included.

**2. Contrastive Learning Implementation**

Description:

In this task, you will continually use the dataset in Task 1, but you will **only use Column ACCX for training**. Your goal is to train a model in the manner of contrastive learning. Below is a step-by-step instruction:

1. Read the paper “[A Simple Framework for Contrastive Learning of Visual Representations](https://arxiv.org/abs/2002.05709)” to get the idea about what contrastive learning is. You are welcome to read any other materials to help you get understood.
2. Run the [code](https://github.com/google-research/simclr) provided in the paper on your local machine successfully and save the printout as a txt file.
3. Using the dataset provided in Task 1 (only Column ACCX is used) as input for the original model and training the model in a contrastive learning manner.

*Hint: The accelerometer data need to be converted to spectrograms so that data augmentation can be done in the same way as images.*

1. After finishing above steps, you will get a well-trained encoder for feature extraction. Now attach two fully connected layer for fine-tuning by using the label dataset (label.csv). Please save the printout as a txt file.

Deliverables:

* Source code. The code should be organized in a clear structure and easy to follow.
* The output files in Step 2 and 4.
* A README.txt with necessary instructions included.

**Submission guidance**

* Submission deadline: **August 22nd by 11:59 PM**
* Submit all your documents as a **zip file** named by your last name, e.g., “Wang.zip”
* You can upload your submission to GitHub, Google Drive or OneDrive, etc. and share a download link with me.
* **Please start earlier as the task may be challenging.**

If you have any questions, please feel free to contact me at [chaowei.wang@mavs.uta.edu](http://chaowei.wang@mavs.uta.edu).