CG4002 Embedded System Design Project

Subcomponent test guidelines for Software components

- **Since the time is limited, please record your setup, walkthroughs and demos emphasizing the points in the rubric below in your videos.** (This is to ensure that all your hard work is captured which may not be possible in our 1-to-1 calls. Please try to keep your video to within 10 mins)
- The number next to the criteria denotes the marks allocated

Software [Machine learning]: [Live+Video]

This assessment is to make sure you are on the right track and can test your software implementation on a dataset of your choice, either taken from an existing dataset or captured by yourself. To meet the criteria for this checkpoint, you will learn how to use libraries and software to perform multi-class classification on a dataset and get familiar with the basics and applications of Machine Learning. We will test your software on the Ultra96 board's processor which you can access remotely.

Your goal is to classify human activities into N moves using Machine Learning:

1. Choose the human activities you would like to classify **[1]**

2. Acquire or produce a dataset: **[1]**
   - The dataset can be acquired from the sources provided in lecture, any online sources or collected on your own.
   - The data can be from any modality, such as inertial signals (accelerometers and gyrometers). Note that the modality of this data should be the same as your design report.

3. Apply machine learning approaches:
   - You should use at least two machine learning algorithms/models to do classification. Note that the selected ML algorithms should be amenable for FPGA acceleration (Work closely with hardware). Options include Neural networks, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), or any others you can think of! The two algorithms for machine learning should be compatible with the FPGA. **[3]**
   - Explain how you segment, and select features, and parameters. **[3]**
   - You should indicate the libraries and software you used for classification. Again, there are many options for you to use, pytorch, tensorflow, scikit-learn, keras, and weka. Show how you have trained your classifiers. **[2]**

0. Evaluation and expected outputs
   You should use some form of model validation to test your approach, for example k-fold cross-validation.
   - Your software should output the following two metrics of your proposed models:
     - Classification accuracy **[1]**
     - Confusion matrix **[1]**

- Try and interpret them (as you might get asked some questions!). **[2]**
- Which model performed better and why? **[2]**
- What do you think makes the activities different and separable in your chosen modalities? **[2]**
- Then, pick one sample from your dataset and show us which activity it is classified into. **[2]**

<u>Software [Dashboard design]: [Live + Video]</u>

1. User survey: Identify your users first, Identify your users' needs, Conduct user survey as a "pilot survey" with participants all from your own groups, Plan for iterative design based on user feedback (e.g. questionnaires, focus groups), determine your software architecture. **[6]**

2. Real-time Visualization
   - Accurately displays the sensor events from emulated queries **[1]**
   - Needs to demonstrate the scalability of dashboard by visualizing all data sent from sensor In addition to being able to code the stream (show competency), you need to be able to justify your decisions. **[3]**
   - Displays "fake" sensor readings and results for classification and relative positions **[3]**

   Real-time scalability: Needs to show ability to receive at least 3 streams of sensor data and display in real-time. Pay attention to system reliability (e.g. the dashboard should be able to stream sensor data for at least 5 mins without noticeable data loss). **[2]**

3. Off-line analytics
   - Needs to take in "test input" logs which consists of [dance move type, relative position] (e.g., Dance Move Type 1 – Position 1, 3, 2) **[2]**
   - Needs to display meaningful statistics based on test logs and ground truth. Ground truth refers to the expected move, and test log refers to the move detected by your system. Calculate offline analytics based on your own inputs of ground truth done during training of the system **[3]**

     e.g.,

Ground truth	Test log,
Move 1,	Move 1
Move 2,	Move 2
Move 3,	Move 1

Note:

I. Because we'll have a limited amount of time for the subcomponent test, you are allowed to record a supplementary video presentation (up to 10 mins to be uploaded to Luminus) for your ML/dashboard components respectively. We will use the videos to assist our evaluation.

II. The evaluation server will not give out the ground truth dance move in replies to the evaluation client when it is running. During the evaluation, the server reveals

the correct dance move via visual prompt that you will be able to watch on the screen.

III. As explained during the lecture, real-time raw data is usually difficult to interpret. Be creative about what features to be visualized in the dashboard. Communicate with your ML person about what features to extract.