Guillermo Reyes guillermo.reyes@dfki.de

Practical task topic 3: Effects of cognitive load on the transfer of control of autonomous vehicles

Distractions while driving can lead to accidents. These distractions can be visual (e.g. taking your eyes off the road), physical (manipulating objects inside the vehicle) and even cognitive (thinking about something else while driving). The degree to which one is distracted will have an impact on how well one can execute the primary task of driving. In this project, you will be focusing on detecting cognitive distractions by estimating the cognitive load (or mental workload) of the driver in real-time. Cognitive load (CL) can be estimated by using subjective or objective measurements. Subjective measurements include questionnaires, such as the NASA-TLX, and are widely used in psychology experiments. However, in order for a system to estimate CL in real-time, objective measurements of the psychophysiological factors of CL must be employed. Multiple types of sensors have been tried in the past in order to estimate CL including skin-conductance sensors, eye-trackers, heart-rate monitor, etc. In this project you will be using an electroencephalogram (EEG) to measure the driver's brain activity while the driver is distracted. By knowing when the driver is distracted, an autonomous system could, e.g., offer to take over the control when it can handle the situation, or estimate when the driver has the cognitive capacities to take the control back from the vehicle.

Step 1: Data Acquisition (7 points)

In order to create a system capable of estimating in real-time when the driver is experiencing a high CL, the first step is to acquire the data. For this purpose, your first task is to design and conduct a small experiment (among yourselves or also with other classmates) to gather enough EEG data to train your prediction model. The experiment should be conducted using a driving simulator. For this, you can use DFKI's OpenDS[1], Microsoft's AirSim[2] or any other driving simulator you want. Every simulator has its own advantages and disadvantages, but the choice is ultimately yours. In order to record the EEG data, you will be using a gtec Unicorn EEG[3]. With this, you will be recording the brain activity of the driver while navigating the virtual environment of the simulator. At the same time, the driver should be distracted by a secondary task to different degrees. One common choice for the secondary task is the auditive n-back task, but you could also include other, more common, secondary tasks for comparison. For this task, you should provide a design description of your experiment with the conditions, variables, hypotheses, etc.

Step 2: Model Training (5 points + 2 extra)

Now that you have acquired the EEG data, the next step is to train the prediction model. Concretely speaking, your task is to train a classification model capable of distinguishing between

[1] You will be added to the MS Team
[2] https://github.com/microsoft/AirSim
[3] https://www.unicorn-bi.com/

high CL and low CL. Additional points if your model distinguishes three classes (high-, mid-, and low-CL). The model you train can be either general or personalized. You can use any libraries you want and are encouraged to try different algorithms. The choice of programming language is free, but Python is strongly suggested. For this task, you should provide a report on your results and discuss them. You should address your hypotheses, and compare the algorithms and conditions that you chose. Feel free to include code snippets, tables, plots, etc.

Step 3: Demonstration (3 points + 1 extra)

By this point you have trained an algorithm capable of classifying the CL level of the driver. The final step of your task is to deploy your algorithm to a real-time system and create a small demonstration. Use the same driving simulator you used for step 1 and modify the environment for the final demonstration. Let the driver put on the EEG used to capture the data and drive in the virtual environment. Then, start loading the driver cognitively with the same or other secondary activities you chose for step 1. Finally, create a draft interface that notifies the driver of their CL and allows them to engage or disengage from autonomous driving. The autonomous vehicle might suggest the driver to relinquish control when they are under high CL. Alternatively, the autonomous vehicle might suggest the driver to take over the control when it detects a driving situation it cannot handle autonomously, while first ensuring that the driver is cognitively capable of taking over. You can decide on which direction you want to handle the transfer of control. The interface should be functional but is not expected to be polished. For this task, you should provide a video or a live demonstration of the system along with the code used for all three subtasks.