

Interaction Technology and Techniques

Assignment 8: Activity Recognition

Summer semester 2017

Submission due: Thursday, 13. July 2017, 23:55

Hand in in groups of max. two.

Your task is to distinguish between different gestures using the accelerometer in the Wiimote.

8.1: Activity Recognition

Implement a Python application that allows the user to record a set of different gestures or activities (standing, sitting, running, shaking the Wiimote, throwing it, etc.) and distinguish between them using data from the Wiimote's accelerometer.

Specifically, implement the following as a PyQtGraph flowchart:

- a graphical user interface where the user can add new gestures and record examples for these gestures using the Wiimote
- a feature-extraction filter that extracts useful information from the raw values (e.g., FFT, stddev, derivatives)
- a machine-learning classifier (e.g., a SVM) that is trained on the example gestures whenever a new example gesture is recorded
- a mode where the user can execute one of the trained gestures with the Wiimote and the system recognizes it.
- optional but helpful: allow the user to remove and retrain gestures.

The individual nodes of the flowchart should be reusable and modular, and should have sensible interfaces (i.e., input and output terminals pass appropriate values, such as lists or integers; no use of application-global variables). Experiment with different sets of gestures and find a few that can be distinguished well.

Example implementation: implement an FftNode that reads in information from a BufferNode and outputs a frequency spectrogram. Furthermore, implement an SvmNode that can be switched between training mode and prediction mode and "inactive" via buttons in the configuration pane (see WiimoteNode and BufferNode for examples). In training mode it continually reads in a sample (i.e. a feature vector consisting of multiple values, such as a list of frequency components) and trains a SVM classifier with this data (and previous data). The category for this sample can be defined by a text field in the control pane. In prediction mode the SvmNode should read in a sample and output the predicted category as a string. Implement a DisplayTextNode that displays the currently recognized/predicted category on the screen.

Hand in the following file:

activity_recognizer.py: a Python file implementing your solution.

Hand in further helper files (incl. wiimote.py, etc.) as needed.

Points

- **1** The python script has been submitted, is not empty, and does not print out error messages.
- **1** The script is well-structured and follows the Python style guide (PEP 8).
- **1** The script is well documented.
- **3** The script correctly implements the features above.

- 1 The flowchart nodes have sensible interfaces.
- 2 The script accurately detects 3 different gestures.
- 2 Training and prediction work in real time
- 2 The user interface for training and prediction is user-friendly and visually pleasant

8.2: Read up on Machine Learning

Download and read either the paper “Avoiding Pitfalls When Using Machine Learning in HCI Studies” (Kostakos et al., 2017) or the paper “Machine Learning: The High-Interest Credit Card of Technical Debt” (Sculley et al., 2014). Both are available in GRIPS. Read appropriate further literature as necessary.

Concisely answer the following questions in your own words :

- What is the title of the paper?
- What are some common use cases for machine learning in practical applications or research prototypes?
- Which problems of machine learning do the authors of the paper identify? Explain one of them in detail.
- What are the credentials of the authors with regard to machine learning? Have they published research on machine learning (or using machine-learning techniques) previously?

Hand in the following file:

machine-learning.txt: a plain-text file containing your answers

Points

- 1 Good answer to first question
- 2 Good answer to second question
- 2 Good answer to third question
- 2 Good answer to fourth question

Submission

Submit via GRIPS until the deadline

All files should use UTF-8 encoding and Unix line breaks. Python files should use spaces instead of tabs.

Have Fun!