Recognition of Emotions Based on Sensor Data

KSWS/Project Smart Computing - MS4 Concept ("Pflichtenheft")


```
Group G01:
Ole Björn Adelmann    (ole.adelmann@uni-rostock.de),
Mhd Esmail Kanaan     (mhd.kanaan@uni-rostock.de),
Dennis Kelm           (dennis.kelm@uni-rostock.de)
Jonny Schlutter       (jonny.schlutter@uni-rostock.de),
Gia Huy Hans Tran     (gia.tran@uni-rostock.de)
```

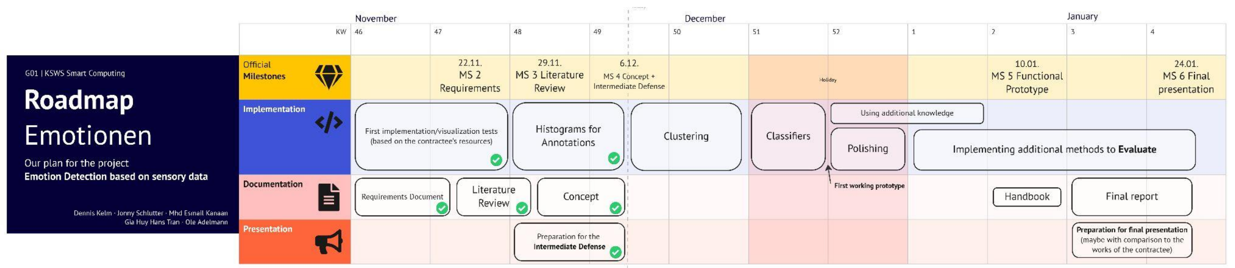
Table of Contents

Purpose and Scope

Our team is tasked with the development of a software that is able to identify the emotions of a person, based on previously recorded motor sensor data gathered from their smartphone or smartwatch.

It is developed as part of the course *KSWS/Project: Smart Computing* in the winter semester 2022/23 at the University of Rostock. Our supervisors are Dr. Sebastian Bader and Maximilian Popko from the *Institute for Visual & Analytic Computing* at the *Faculty of Informatics and Electrical Engineering* of the University of Rostock.
The paper by our supervisors [1] and the Extrasensory Dataset [2] serve as the basis for our project. The main goal is to connect the methods that were used by these papers. We carried out a reproducible, systematic literature review on emotion recognition related papers, and other published work that has been done using the ExtraSensory Dataset.

Roadmap



See this page or access the website of the roadmap[1] for the newest version of this image. The following text provides more context.

Implementation

As we are extending the work by our contractor, we will start our implementation with first tests and adjust some parameters of it. Next, we will generate histograms based on the given emotions by the participants. After the official milestone MS 4, we will begin to have a look at clustering. As we will do more work on documentation and presentation at the beginning of the project period, we want to implement at least the basics of these two parts of our pipeline. In the holidays (and when possible, the weeks before), we will focus on classification. It is important that we develop a modularized system, so we can work parallel. Using knowledge to enhance our predictions is also a component of our project, so we will implement that at the beginning of the year. Furthermore, we will take some time to make our results better looking and, hopefully, error-free, so that our

[1] See https://miro.com/app/board/uXjVPCn1X-w=/?share_link_id=75534587031

functional prototype will not be a prototype anymore. Our project will end with the evaluation phase.
For more details, read the functional requirements.

Documentation

After we finish this document, we will do a literature review to find the best approaches to our given problem. And with MS 4, we have developed a concept paper with more detailed descriptions on our modules and what we have found out in our literature review (i.e. which methods to use). Documentation is the main part of the first weeks of the project, after that, we will focus only on the implementation. At the end, we will provide a handbook on how to use our finished scientific product. This will be very short as this is a research project.

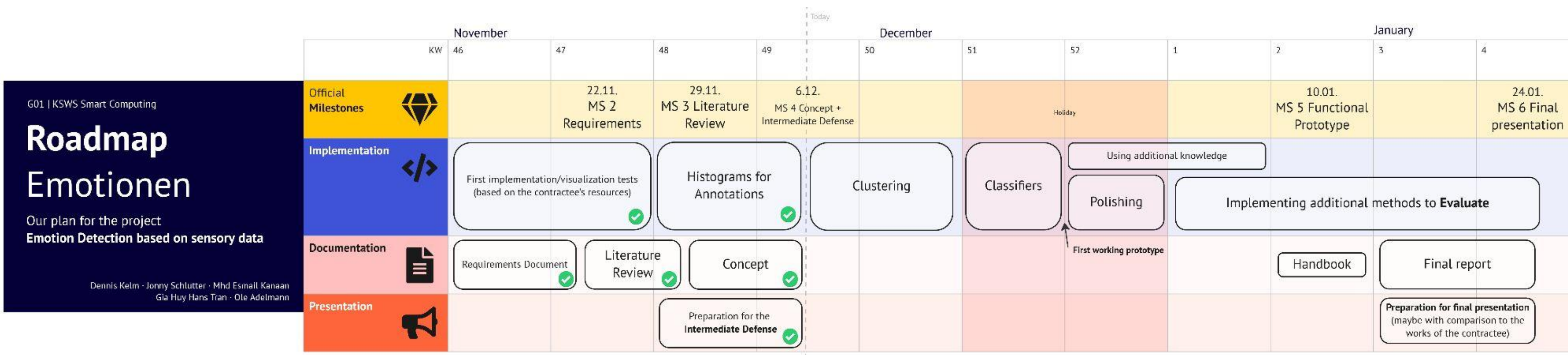
Presentation

Like documentation, presentation is only relevant at the beginning (subtopics and intermediate defense), and at the end with the final presentation to the contractors. As our supervisor Sebastian Bader pointed out in a meeting, a scientific poster could be possibly required, so in that case we will make that. Our visualizations would fit well to that.

What happens if?

- Last week of 2022 is for work that has not been done before, if we need more time e.g. for a module
- Our “Scrum” project management technique allows us to assign issues dynamically, so when somebody cannot work for e.g., a week (because of illness, …), all other members can do his work. So we are not dependent on one specific team member.
- When we are finished with one module earlier than planned, we have more time to polish it or work on other modules
- We focus on performance only to such an extent that we do not run into timing issues in the evaluation phase, as we have to run the whole program with different parameters etc.

Roadmap - big image



Functional Requirements

Our pipeline consists of different functions, more information about the used methods to implement them can be found in the next chapter. We focus on clearly separating code and data.

We will use the source [6] to reproduce its results and to enhance it with additional knowledge, i.e. the fact that every person behaves similarly on the same hours.

#	Code	Data	Description	Planned methods
010	-	Raw mood data Raw motion data	Already available as separated data	-
020	-	Motion features	Motion features already extracted for the ExtraSensory Dataset	Additionally, calculate the *hour of the day/data point of the day* (for better use with additional knowledge)
030	Cleaning mood data	Clean mood data	Clean out null and irrelevant data, e.g. subjects that did not contribute mood data	Only use data by the 18 subjects that provided more than 1000 samples and have less than 90% missing data (like it was done in [6])
040	Generating mood labels[2]	Mood labels at the timestamps	Generate a list of actual emotions at the timestamps (do not list every emotion that has not been transmitted)	We will use the PDA model like in [6]
050	Visualization of mood labels	-	-	If possible with our libraries, a "boxed plot" will be generated, plotting labels for each participant for each day with every hour of the day[3] (if not, we will use some other visualization)
060	Generating	Divisions (e.g.		Divide the data in 30 minute

[2] Our synonym for "annotations"
[3] see "annotations" graphic in the original paper or here

	divisions	divided timeslots)		time slots (test multiple time slot lengths -> see Secondary Research Questions, longer or shorter is possible)
061	Generating histograms	Histograms	Distributions of the mood labels for each division	We will use (and possibly) enhance the `generate_histograms` method by Maximilian in his preliminary work
062	Visualization of the histograms	-	Generating violin plots of the histograms	1. Bar chart like Figure 3a [1] or Figure 1 in [6] 2. Violin chart like Figure 3b [1]
070	Clustering	Clustered data	Cluster the histograms	KMeans[4] (if possible: Latent Dirichlet Allocation[5] additionally, because we give it a lower priority)
071	Clustering visualization	Plots	-	Violin plot (to better see the distribution)
080	Training classifiers	-	Training a classifier for each cluster on motion features to identify emotion (to use additional knowledge, give the classifier the *hour of the day* feature)	High priority: Random Forest[6] and SVM[7] [8] with five-fold, stratified cross-validation, because it has the best overall evaluation score in [6][9]
090	Emotion prediction	Emotion	Predict the emotion by using the correct classifier on a sensor data input	-

[4] See also https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[5] See also https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html

[6] See also https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[7] See also https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html and the user guide https://scikit-learn.org/stable/modules/svm.html

[8] If we have the time to, the algorithms XGBoost and CatBoost could also be investigated, so they are prioritized low

[9] See Figure 5 of this source

100	Evaluation of clustering	Score	Use different parameters (different cluster sizes k) etc.	Silhoette coefficient for score[10] and to find an optimal k[11] ($0 \leq k \leq 10$ is reasonable), F-measure[12] and Fowlkes-Mallows index[13]. Also use the F1 score by the classifier.
101	Visualization of the cluster evaluation	Plot	-	Scatterplot matrix, line chart
110	Evaluation of classifying	Score	Use different algorithms for classifying	Accuracy, Precision, Recall, Specificity, AUROC, F1 like in [6]
111	Visualization of the classifier evaluation	Plot	-	Tables like in [6] and bar charts because there are only two algorithms to compare (with their different scores)

[10] See also https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

[11] See also https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

[12] See also https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html

[13] See also https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fowlkes_mallows_score.html

General methods

The software takes a set of data, mainly consisting of motor sensor data, as input. Using machine learning methods, a model capable of identifying emotions based on sensor data needs to be constructed. In the next sections we describe the dataset that we will be using and provide a general outline on the pipeline of the software.

Used programming tools

We will use the programming language Python[14] with several science-oriented libraries (i.e., scikit-learn[15], pandas[16] and numpy[17]) and with a directory structure that is based on the functional requirements table. Anaconda[18] will be our Python environment, as it contains every needed library. Tableau Desktop[19] will be part of our tools to quickly generate internal visualizations, but for final visualizations we will only use matplotlib[20] and seaborn[21]. To visualize our roadmap, we use Miro[22]. And to organize ourselves, we use the built-in Scrum tool in GitLab, that will also be our repository.

Data Description

We will be training our models using the ExtraSensory [2] dataset. This dataset is made up of accelerometer, gyroscope, and other data gathered from the smartphones and smartwatches of 60 users. The data was recorded in intervals of one minute. Each datapoint is labeled with the activity being done and optionally with the mood felt while the measurements were taken.

sensor	details	dimension	#us	#ex
accelerometer	Tri-axial direction and magnitude of acceleration. 40Hz for ~20sec.	(~800) x 3	60	308,306
gyroscope	Rate of rotation around phone's 3 axes. 40Hz for ~20sec.	(~800) x 3	57	291,883
magnetometer	Tri-axial direction and magnitude of magnetic field. 40Hz for ~20sec.	(~800) x 3	58	282,527
watch accelerometer	Tri-axial acceleration from the watch. 25Hz for ~20sec.	(~500) x 3	56	210,716
watch compass	Watch heading (degrees). nC samples (whenever changes in 1deg).	nC x 1	53	126,781
location	Latitude, longitude, altitude, speed, accuracies. nL samples (whenever changed enough).	nL x 6	58	273,737
location (quick)	Quick location-variability features (no absolute coordinates) calculated on the phone.	1 x 6	58	263,899
audio	22kHz for ~20sec. Then 13 MFCC features from half overlapping 96msec frames.	(~430) x 13	60	302,177
audio magnitude	Max absolute value of recorded audio, before it was normalized.	1	60	308,877
phone state	App status, battery state, WiFi availability, on the phone, time-of-day.	5 discrete	60	308,320
additional	Light, air pressure, humidity, temperature, proximity. If available sampled once in session.	5	---	---

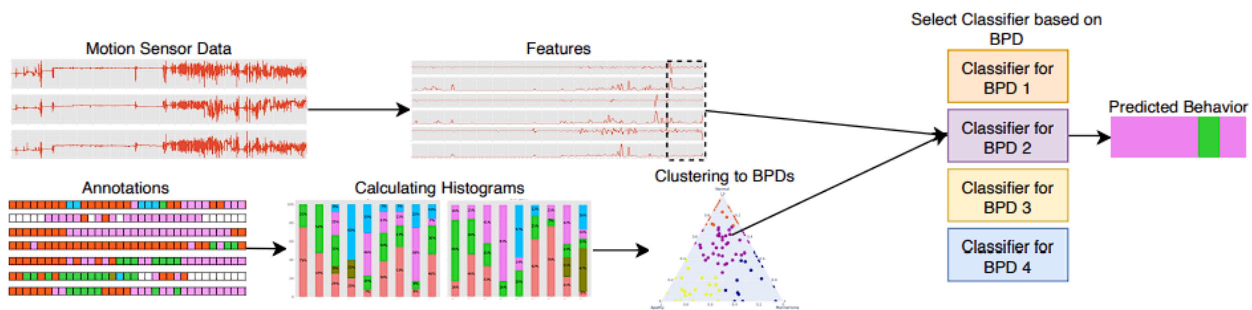
[14] See also https://www.python.org/
[15] See also https://scikit-learn.org/stable/index.html
[16] See also https://pandas.pydata.org/
[17] See also https://numpy.org/
[18] See also https://www.anaconda.com/products/distribution
[19] See also https://www.tableau.com/
[20] See also https://matplotlib.org/stable/index.html
[21] See also https://seaborn.pydata.org/
[22] See also https://miro.com/

Detailed list of all available sensors [2]

We will need to inspect the data and process it to fit our needs. The data is already very tidy, e.g. the sensor data already has its features extracted. However, data on moods are only given very sparsely. Most participants did not note down any emotions, as they were optional during data collection. We will need to find ways to make the most of what is available. Imputing missing labels could be possible, but could also introduce too much bias to be viable.

Pipeline

We are basing our approach on [1]. Which means we will use clustering to identify emotional patterns with respect to time and other factors, and build a classifier for each cluster to predict emotion based on motion-data.



Basic Overview [1]

To get our clusters, we only need to look at the mood data. So first, we separate sensor data and mood labeled data. Then we can calculate the distributions of emotions for a given set of time frames (called Histograms). Lastly, and most importantly, we will carry out an evaluation of our clustering and classifying approaches, by varying parameters and methods.

Calculating Histograms

Our Base Data is annotated just every minute with any number of emotions.
This data is grouped into time frames of a set length. This length could be set to any value, but 30 minutes is suitable. Using a length of 30 minutes, each day is separated into 48 timeslots of 30 minutes length, starting from 00:00:00 to 00:30:00, then 00:30:00 to 01:00:00, and so on. For each time frame, the percentage wise distribution of emotions is then calculated.

Alternatively, we can also utilize other available data while creating our histograms (i.e., we use additional knowledge). We could improve upon the approach used in [1] by taking into account the noise level of the environment, or the current battery percentage, as they can have a direct impact on a person's behavior.

Next, we try to cluster these distributions into a certain number of groups.

Clustering

“Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups.” [5]

We will use clustering to identify the main emotional patterns over every recorded timeframe. Then, we can use this information to improve our classification of emotions. Instead of training a single classifier to directly map emotions to sensor data, we will train one classifier for each cluster we identify. For this, we will need to identify the appropriate amount of clusters.
In [1], the KMeans algorithm was used. We can try out other algorithms to find out if an improvement can be made this way.

Classification

“**Classification** is the process of recognizing, understanding, and grouping ideas and objects into preset categories or “sub-populations”. Using pre-categorized training datasets, machine learning programs use a variety of algorithms to classify future datasets into categories.” [3]

Using the groups identified during clustering as the basis, we train a classifier for each group, mapping sensor data to emotion. In [1], Naive Bayes, Support Vector Machine and Majority algorithms were used for the classifier. We can try out other algorithms to find out if an improvement can be made this way. So given the sensor data and a clustered group as input, the model will select a certain classifier which is the best for the given cluster. This classifier will then predict our output, the emotion.

Evaluation

We want to evaluate the performance of our models by using varying parameters, different methods or dividing up our data differently. To evaluate the end results of our classification step we will use accuracy, precision, F1-Score and ROC-AUC.
Moreover, we will look at different, appropriate visualizations, in order to possibly find patterns (e.g., higher cluster sizes have higher scores). [4]

Research questions

We will evaluate these research questions:

Primary Comparison of Methods:

- Which **clustering algorithm** KMeans or LDA has the best overall evaluation result, when it is applied to the ExtraSensory dataset? (this question will be answered if LDA is also used)
- Which **amount of clusters** (k) has the best Silhoette coefficient score, F-measure and Fowlkes-Mallows index, when it is applied to the ExtraSensory dataset?
- Which **classification algorithm**[23] SVM or Random Forest has the best Accuracy, Precision, Recall, Specificity, AUROC and F1 score , when it is applied to the ExtraSensory dataset?

Optional Secondary Research Questions:

- How is performance impacted if we **generate our histograms**, i.e. our datapoints for clustering, not solely by time, but also **taking into account other data** like the battery percentage of the phone?
- How many of our initial datapoints of the ExtraSensory Dataset should be condensed into one histogram, i.e. one datapoint for clustering, for optimal results?
- Can we use Two-Step-Classification to first choose a cluster and then choose an emotion? (see Popko Future work last paragraph) -> by how much does the performance drop if we use two step Classification instead of assuming the right cluster is already known?

[23] XGBoost or CatBoost are prioritized low, but could also be investigated (see Functional Requirements)

Sources

[1] M. Popko, S. Bader, S. Lüdtke, and T. Kirste, "Discovering behavioral predispositions in data to improve human activity recognition," *arXiv [cs.LG]*, Jul. 2022. Available: https://doi.org/10.48550/arXiv.2207.08816

[2] Vaizman, Y., Ellis, K., and Lanckriet, G. "Recognizing Detailed Human Context In-the-Wild from Smartphones and Smartwatches". *IEEE Pervasive Computing*, vol. 16, no. 4, October-December 2017, pp. 62-74. doi:10.1109/MPRV.2017.3971131. Also visit http://extrasensory.ucsd.edu/.

[3] "5 types of classification algorithms in machine learning," *MonkeyLearn Blog*, 26-Aug-2020. [Online]. Available: https://monkeylearn.com/blog/classification-algorithms/. [Accessed: 22.11.2022].

[4] Source: Lecture by M. Becker, "Part 2 - Data Science in Practise", *University of Rostock*, 16.11.2022.

[5] "Cluster analysis", *Wikipedia*. Available: https://en.wikipedia.org/wiki/Cluster_analysis. [Accessed: 22.11.2022]

[6] Sultana M, Al-Jefri M, Lee J
Using Machine Learning and Smartphone and Smartwatch Data to Detect Emotional States and Transitions: Exploratory Study
JMIR Mhealth Uhealth 2020;8(9):e17818
URL: https://mhealth.jmir.org/2020/9/e17818
DOI: 10.2196/17818