

Durchgeführt im Rahmen des

## **BMBF-Projekt: ELISE**

# **Dokumentation der Projektarbeit**

Entwurf eines kompakten mikrocontrollergestützten Systems zur  
Emotionserkennung in einer Virtual-Reality-Umgebung

**WiSe 2017/2018 und SoSe 2018**

### **Projektbetreuer:**

#### **Medizinische Informatik und Mikrosystementwurf**

Prof. Dr. rer. nat. Rainer Brück

Dr.-Ing. Armin Grünewald

M.Sc. David Krönert

#### **Forschungsgruppe für Mustererkennung**

Prof. Dr.-Ing. Marcin Grzegorzek

M.Sc. Frédéric Li

### **Projektteilnehmer:**

Artur Piet (Sprecher der Projektgruppe)

Jonas Pöhler (Stellv. Sprecher der Projektgruppe)

Arnaud Eric Toham Waffo

Boris Kamdem

Kevin Orth

Meryem Dural

Minas Michail

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>8</b>
1.1	Hintergrund und Motivation . . . . .	8
1.2	ELISE Projektbeschreibung . . . . .	8
1.3	Gliederung dieser Dokumentation . . . . .	8
1.4	Anhang . . . . .	8
<b>2</b>	<b>Organisation</b>	<b>9</b>
2.1	Verantwortungsbereiche . . . . .	10
2.2	Gruppentreffen . . . . .	10
<b>3</b>	<b>Grundlagen</b>	<b>11</b>
3.1	Definition von Emotionen . . . . .	11
3.2	Virtual Reality (VR) . . . . .	11
3.3	Sensoren und biophysiological Signale zur Emotionserkennung . .	11
3.3.1	Körpertemperatur-Sensor . . . . .	11
3.3.2	Blood Volume Pulse-Sensor (BVP) . . . . .	11
3.3.3	Messen der Sauerstoffsättigung (SpO2) . . . . .	11
3.3.4	Galvanic Skin Response (GSR) . . . . .	11
3.3.5	Elektroenzephalografie (EEG) . . . . .	11
3.3.6	Elektrookulografie (EOG) . . . . .	11
3.3.7	Analog/Digital-Wandler . . . . .	11
3.4	Kommunikation . . . . .	11
3.5	Grundlagen der Mustererkennung . . . . .	11
3.6	Emotion Recognition Chain . . . . .	13
3.6.1	Datenerfassung . . . . .	13
3.6.2	Vorverarbeitung . . . . .	13
3.6.3	Segmentation . . . . .	14
3.6.4	Merkmalsextraktion . . . . .	15
3.6.5	Klassifikation . . . . .	15
<b>4</b>	<b>State-of-the-Art Analyse</b>	<b>20</b>
<b>I</b>	<b>Erster Prototype</b>	<b>21</b>
<hr/>		
<b>5</b>	<b>Systementwurf und Konzept</b>	<b>21</b>
5.1	Anforderungen . . . . .	21
5.2	Konzept . . . . .	21

5.3	Hardwareauswahl . . . . .	21
5.3.1	Auswahlkriterien . . . . .	21
5.3.2	Festlegung der genutzten Hardware . . . . .	21
5.4	Hardwarearchitektur . . . . .	21
5.4.1	GSR-Sensor . . . . .	21
5.4.2	Temperatur-Senosr . . . . .	21
5.4.3	Pulsoximeter . . . . .	21
5.4.4	EEG . . . . .	21
5.4.5	EOG . . . . .	21
5.4.6	Datenübertragung . . . . .	21
5.5	Programmierung . . . . .	21
5.6	Aufnahme der übertragenen Daten . . . . .	21
<b>6</b>	<b>Realisierung</b>	<b>22</b>
<b>7</b>	<b>Emotionsinduktion</b>	<b>23</b>
7.1	Ablauf . . . . .	23
7.2	Fragebogen . . . . .	23
7.3	Szenarien . . . . .	23
7.3.1	Glück . . . . .	23
7.3.2	Langeweile . . . . .	23
7.3.3	Frustration . . . . .	23
<b>8</b>	<b>Messreihe</b>	<b>24</b>
<b>9</b>	<b>Mustererkennung</b>	<b>25</b>
9.1	Merkmalsextraktion für Emotionserkennung . . . . .	25
9.1.1	Hand-gefertigte Merkmale . . . . .	25
9.1.2	Codebook Approach . . . . .	25
<b>10</b>	<b>Ergebnisse</b>	<b>26</b>
10.1	Ergebnisse der hand-gefertigten Merkmale . . . . .	26
10.2	Ergebnisse des Codebook Approach . . . . .	26
10.3	Analyse der Ergebnisse . . . . .	26
<b>II</b>	<b>Zweiter Prototype</b>	<b>27</b>
<b>11</b>	<b>Systementwurf und Konzept</b>	<b>27</b>
11.1	Anforderungen . . . . .	27
11.2	Konzept . . . . .	27

11.3	Hardwareauswahl . . . . .	27
11.3.1	Auswahlkriterien . . . . .	27
11.3.2	Festlegung der genutzten Hardware . . . . .	27
11.4	Hardwarearchitektur . . . . .	27
11.4.1	GSR-Sensor . . . . .	27
11.4.2	Temperatur-Senosr . . . . .	27
11.4.3	Pulsoximeter . . . . .	27
11.4.4	EEG . . . . .	27
11.4.5	EOG . . . . .	27
11.4.6	Datenübertragung . . . . .	27
11.5	Programmierung . . . . .	27
11.6	Aufnahme der übertragenen Daten . . . . .	27
<b>12</b>	<b>Realisierung</b>	<b>28</b>
<b>13</b>	<b>Emotionsinduktion</b>	<b>29</b>
13.1	Ablauf . . . . .	29
13.2	Fragebogen . . . . .	29
13.3	Szenarien . . . . .	29
13.3.1	Glück . . . . .	29
13.3.2	Langeweile . . . . .	29
13.3.3	Frustration . . . . .	29
<b>14</b>	<b>Messreihe</b>	<b>30</b>
<b>15</b>	<b>Mustererkennung</b>	<b>31</b>
15.1	Merkmalsextraktion für Emotionserkennung . . . . .	31
15.1.1	Hand-gefertigte Merkmale . . . . .	31
15.1.2	Codebook Approach . . . . .	31
<b>16</b>	<b>Ergebnisse</b>	<b>32</b>
16.1	Ergebnisse der hand-gefertigten Merkmale . . . . .	32
16.2	Ergebnisse des Codebook Approach . . . . .	32
16.3	Analyse der Ergebnisse . . . . .	32
<b>III</b>	<b>Dritter Prototype</b>	<b>33</b>
<hr/>		
<b>17</b>	<b>Systementwurf und Konzept</b>	<b>33</b>
17.1	Anforderungen . . . . .	33
17.2	Konzept . . . . .	33

17.3	Hardwareauswahl . . . . .	33
17.3.1	Auswahlkriterien . . . . .	33
17.3.2	Festlegung der genutzten Hardware . . . . .	33
17.4	Hardwarearchitektur . . . . .	33
17.4.1	GSR-Sensor . . . . .	33
17.4.2	Temperatur-Senosr . . . . .	33
17.4.3	Pulsoximeter . . . . .	33
17.4.4	EEG . . . . .	33
17.4.5	EOG . . . . .	33
17.4.6	Datenübertragung . . . . .	33
17.5	Programmierung . . . . .	33
17.6	Aufnahme der übertragenen Daten . . . . .	33
<b>18</b>	<b>Realisierung</b>	<b>34</b>
<b>19</b>	<b>Emotionsinduktion</b>	<b>35</b>
19.1	Ablauf . . . . .	35
19.2	Fragebogen . . . . .	35
19.3	Szenarien . . . . .	35
19.3.1	Glück . . . . .	35
19.3.2	Langeweile . . . . .	35
19.3.3	Frustration . . . . .	35
<b>20</b>	<b>Messreihe</b>	<b>36</b>
<b>21</b>	<b>Mustererkennung</b>	<b>37</b>
21.1	Merkmalsextraktion für Emotionserkennung . . . . .	37
21.1.1	Hand-gefertigte Merkmale . . . . .	37
21.1.2	Codebook Approach . . . . .	37
<b>22</b>	<b>Ergebnisse</b>	<b>38</b>
22.1	Ergebnisse der hand-gefertigten Merkmale . . . . .	38
22.2	Ergebnisse des Codebook Approach . . . . .	38
22.3	Analyse der Ergebnisse . . . . .	38
<b>IV</b>	<b>Vierter Prototype</b>	<b>39</b>
<b>23</b>	<b>Systementwurf und Konzept</b>	<b>39</b>
23.1	Anforderungen . . . . .	39
23.2	Konzept . . . . .	39

23.3	Hardwareauswahl . . . . .	39
23.3.1	Auswahlkriterien . . . . .	39
23.3.2	Festlegung der genutzten Hardware . . . . .	39
23.4	Hardwarearchitektur . . . . .	39
23.4.1	GSR-Sensor . . . . .	39
23.4.2	Temperatur-Senosr . . . . .	39
23.4.3	Pulsoximeter . . . . .	39
23.4.4	EEG . . . . .	39
23.4.5	EOG . . . . .	39
23.4.6	Datenübertragung . . . . .	39
23.5	Programmierung . . . . .	39
23.6	Aufnahme der übertragenen Daten . . . . .	39
<b>24</b>	<b>Realisierung</b>	<b>40</b>
<b>25</b>	<b>Emotionsinduktion</b>	<b>41</b>
25.1	Ablauf . . . . .	41
25.2	Fragebogen . . . . .	41
25.3	Szenarien . . . . .	41
25.3.1	Glück . . . . .	41
25.3.2	Langeweile . . . . .	41
25.3.3	Frustration . . . . .	41
<b>26</b>	<b>Messreihe</b>	<b>42</b>
<b>27</b>	<b>Mustererkennung</b>	<b>43</b>
27.1	Merkmalsextraktion für Emotionserkennung . . . . .	43
27.1.1	Hand-gefertigte Merkmale . . . . .	43
27.1.2	Codebook Approach . . . . .	43
<b>28</b>	<b>Ergebnisse</b>	<b>44</b>
28.1	Ergebnisse der hand-gefertigten Merkmale . . . . .	44
28.2	Ergebnisse des Codebook Approach . . . . .	44
28.3	Analyse der Ergebnisse . . . . .	44
<b>29</b>	<b>Zusammenfassung und Ausblick</b>	<b>45</b>
29.1	Zusammenfassung . . . . .	45
29.2	Fazit . . . . .	45
29.3	Ausblick . . . . .	45
	<b>Abbildungsverzeichnis</b>	<b>46</b>

<b>Tabellenverzeichnis</b>	<b>47</b>
<b>Abkürzungen</b>	<b>48</b>
<b>Anhang</b>	<b>49</b>

# 1 Einleitung

Verantwortlich: Minas

- Next Step: Bitte selbst bei github hochladen oder Artur zuschicken

## 1.1 Hintergrund und Motivation

## 1.2 ELISE Projektbeschreibung

## 1.3 Gliederung dieser Dokumentation

## 1.4 Anhang



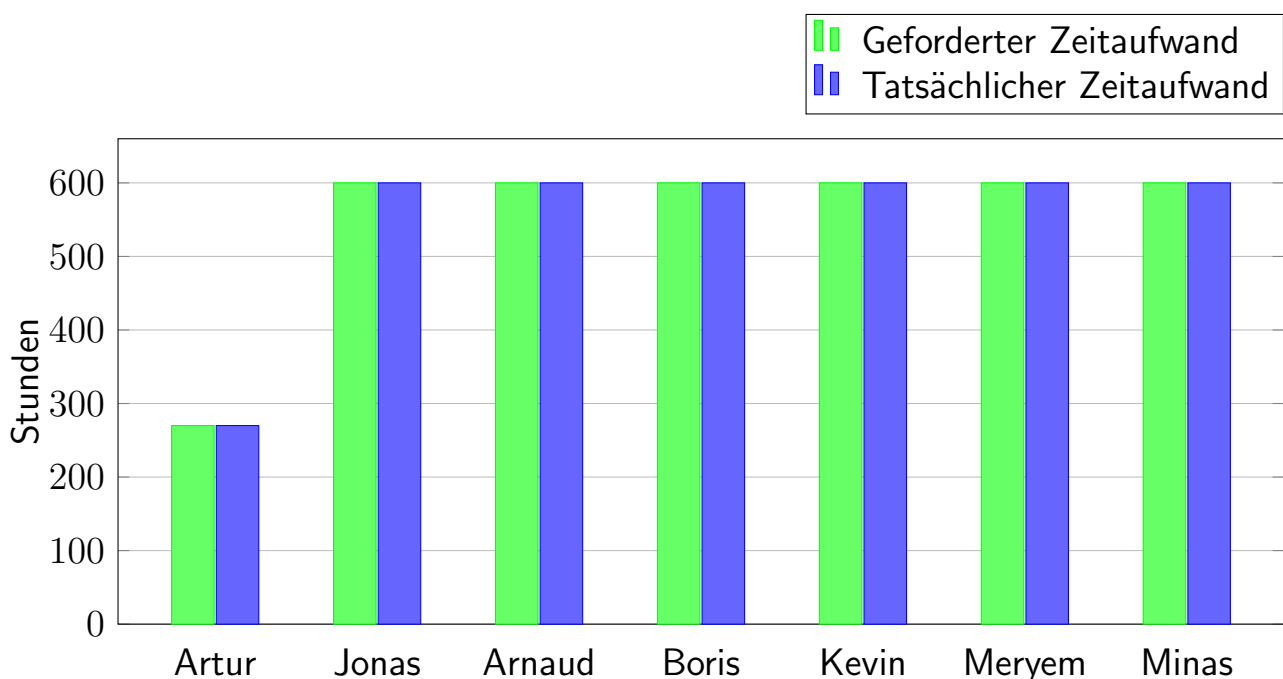
## 2 Organisation

Verantwortlich: Artur

Diese Projektgruppe des ELISE Projektes wird dem Lehrstuhl Medizinische Informatik & Mikrosystementwurf und dem Lehrstuhl für Mustererkennung zugeordnet.

Die Gruppe wird von Dr.-Ing. Armin Grünewald, David Krönert und Frédéric Li geleitet. Innerhalb der Gruppe wurden dazu unabhängig ein Sprecher und ein stellvertretender Sprecher von den Gruppenmitgliedern gewählt. Als Projektgruppensprecher wurde Artur Piet und als Stellvertreter Jonas Pöhler ausgewählt. Diese Stellung ist jedoch nicht die eines Leiters mit Entscheidungs- und Weisungsbefugnissen. Die Sprecher sind also auf die Kooperationsbereitschaft der anderen Gruppenmitglieder angewiesen. Konkrete Aufgaben der Sprecher waren unter anderem das Verteilen von Verantwortungsbereichen auf alle Mitglieder, die Einführung von regelmäßigen Gruppentreffen um den Überblick über alle Fortschritte zu garantieren und die Übernahme möglichst aller organisatorischer Tätigkeiten (z.B. Messreihen organisieren, Beschaffungsprozesse koordinieren, usw.).

Die Laufzeit der Projektgruppe wurde auf etwa 1 Jahr gesetzt, wobei das Kick-Off Meeting am 23.10.2017 stattfand. Der geforderte individuelle Zeitaufwand aller Gruppenmitglieder entspricht der jeweils im Modulhandbuch des Studienganges definierten Leistungspunkte für die Projektarbeit, also 600 Stunden für Jonas Pöhler, Arnaud Eric Toham Waffo, Boris Kamdem, Kevin Orth, Meryem Dural sowie Minas Michail und 270 Stunden für Artur Piet. Es folgt ein Balkendiagramm mit den geforderten Stunden im Vergleich zum tatsächlichem Zeitaufwand.



## 2.1 Verantwortungsbereiche

Innerhalb der Projektgruppe (PG) war ein Ziel, dass jeder der Mitglieder "Experte" für einen Bereich wird. Damit haben wir die Verantwortung relativ gleichmäßig auf alle aufgeteilt. Dazu muss noch erwähnt werden, dass die Teammitglieder nicht nur ausschließlich die Aufgaben des jeweiligen Verantwortungsbereiches erledigt haben. Es wurde sich vielmehr gegenseitig immer unterstützt und viel zusammengearbeitet. Die Verantwortungsbereiche haben sich mit der Zeit folgendermaßen aufgeteilt:

- Artur Piet: Mustererkennung, 3D-Konstruktion und Sprecher der PG
- Jonas Pöhler: Hardware, Webanbindung und Stellv. Sprecher der PG
- Arnaud Eric Toham Waffo: Elektrodenauswahl
- Boris Kamdem: Langeweile-Szenario und Fragebogen in VR
- Kevin Orth: Komplette Hardware
- Meryem Dural: Frustrations-Szenario in VR
- Minas Michail: Glücks-Szenario in VR

## 2.2 Gruppentreffen

Wöchentlichen Gruppentreffen fanden immer Donnerstags ab 14:00 Uhr statt und beginnen in der Regel mit einer Update-Runde. Hierbei kommen alle Gruppenteilnehmer der Reihe nach dran und jeder sagt kurz woran letzte Woche gearbeitet wurde, was die aktuellen Herausforderungen und Probleme sind und was genau als nächstes zu tun geplant wird. Ziel ist es den Austausch und die Kommunikation unter den Teammitgliedern und den Projektleitern zu fördern und damit alle Teilnehmer auf dem selben Wissenstand zu halten, da einzelnen Aufgaben durchaus großen Einfluß auf Tätigkeiten von anderen Mitgliedern haben können.

## 3 Grundlagen

Verantwortlich: Arnaud

### 3.1 Definition von Emotionen

Verantwortlich: Arnaud

### 3.2 Virtual Reality (VR)

Verantwortlich: Arnaud, Boris

### 3.3 Sensoren und biophysiological Signale zur Emotionserkennung

Verantwortlich: Arnaud, Kevin

#### 3.3.1 Körpertemperatur-Sensor

#### 3.3.2 Blood Volume Pulse-Sensor (BVP)

#### 3.3.3 Messen der Sauerstoffsättigung (SpO2)

#### 3.3.4 Galvanic Skin Response (GSR)

#### 3.3.5 Elektroenzephalografie (EEG)

#### 3.3.6 Elektrookulografie (EOG)

#### 3.3.7 Analog/Digital-Wandler

### 3.4 Kommunikation

Verantwortlich: Kevin, Jonas

### 3.5 Grundlagen der Mustererkennung

Verantwortlich: Artur

- Bereit zum Korrekturlesen.

Mustererkennung (enlg. "pattern recognition") ist ein Unterthema des maschinellen Lernens. Das Ziel besteht darin, automatisierte Systeme zu entwerfen, die hoch abstrakte Muster in Daten erkennen können. Konkret heißt dies, dass man Maschinen beibringen möchte komplexer Aufgaben zu lösen, welche vom Menschen nahezu mühelos und natürlich erledigt werden können. Typische Beispiele für die zahlreichen Anwendungsbereiche sind die Objekterkennung, Spracherkennung sowie die

Erkennung und Verfolgung in Bildern. Die Emotionserkennung ist ein Anwendungsbereich der Mustererkennung. Die Hauptidee hinter der Lösung eines Mustererkennungs-Problems ist es, dieses als Klassifikationsproblem zu übersetzen, wobei die zu erkennende Mustern die unterschiedliche Klassen bilden. Die vom Mustererkennungs-System eingegebenen Daten werden dann verarbeitet und der “am nächsten liegenden” Klasse zugeordnet. Beispielsweise können bei der Emotionserkennung die Eingangsdaten Bilder oder physiologische Signale sein, die in verschiedene Klassen eingeteilt werden, welche jeweils einer Emotion entsprechen.

Ein wichtiger Teil eines jeden Mustererkennung-Problems ist der Lernansatz, mit welchem die Maschine lernen soll die Muster in den Daten zu erkennen. Traditionell werden zwei Ansätze verwendet:

- Überwachter Lernansatz: Dieser Ansatz kann nur verwendet werden, wenn vor der Verarbeitung der Daten ein Datenbeschriftungsschritt durchgeführt wurde. In diesem Schritt wird jedem Element des Datensatzes ein Etikett (engl. “label”) zugewiesen, das angibt, welcher Klasse der jeweilige Datenpunkt zugeordnet werden kann. Die zusätzlichen Informationen, die die Etiketten liefern, werden als Grundlage verwendet, um sie mit der Vorhersage des Systems zu vergleichen und zu korrigieren, wenn sie nicht gleich sind.
- Unüberwachter Lernansatz: Dieser Ansatz wird verwendet, wenn keine Etiketten für die Daten vorhanden sind. Unüberwachte Lerntechniken zielen darauf ab, der Maschine beizubringen, Muster in den Daten selbst zu finden. Sie werden meist verwendet, um Einblicke in Daten zu erhalten, deren Struktur unbekannt ist.

Überwachtes Lernen liefert aktuell weit bessere Ergebnisse, jedoch ist die Beschriftung mit Etiketten der Daten nicht immer einfach oder teilweise sogar gar nicht möglich (z.B. wenn die Datenmenge sehr groß ist oder wenn Unsicherheit über die Vergabe der Etiketete besteht). Aus diesem Grund wächst das Interesse an unüberwachten Lernansätzen. Diese Ansätze sind jedoch schwierig zu benutzen, da sie eine große Menge an Daten voraussetzen. Kompromisse sind mit semi-überwachten Lernansätzen möglich, bei denen die Daten für einen Teil des Datensatzes (aber nicht für den ganzen Datensatz) mit Etiketten beschriftet und damit bekannt sind. In diesem Fall kann eine Mischung aus überwachten und Unüberwachten Techniken angewendet werden [1].

Im Rahmen des ELISE-Projekts werden mit Hilfe von Mustererkennungsverfahren eindimensionale Zeitsignale von physiologischen Sensoren in Echtzeit für die Erkennung von drei Emotionen verarbeitet: Glück, Frustration und Langeweile. Um den Emotionsklassifizierer aufzubauen, wird ein standardmäßiger, überwachter Ler-

ansatz namens Emotion Recognition Chain verwendet, der im folgendem Kapitel beschrieben wird.

## 3.6 Emotion Recognition Chain

Verantwortlich: Artur  
- ERC Bild ändern (5 statt 4 Schritte)

Die Emotion Recognition Chain (ERC) besteht aus fünf Hauptschritten: Datenerfassung, Vorverarbeitung, Segmentierung, Merkmalsextraktion und Klassifizierung (vgl. Abb. 1). In den folgenden Unterkapiteln wird für jeden Schritt eine allgemeine Erklärung gegeben.

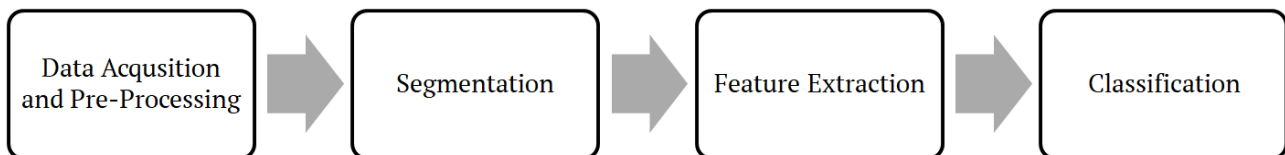


Abbildung 1: Emotion Recognition Chain: Zeitreihen-Datensätze werden von tragbaren Sensoren aufgenommen (Datenerfassung) und vorverarbeitet (Vorverarbeitung). Die Daten werden dann in Segmente unterteilt (Segmentierung), aus denen Merkmale extrahiert werden (Merkmalsextraktion). Mit den gewonnenen Merkmalen wird schließlich ein Klassifikator trainiert und anschließend dessen Ergebnisse bewertet (Klassifikation).

### 3.6.1 Datenerfassung

Verantwortlich: Artur  
@Kevin, bitte Korrekturlesen und mit Deinem Teil abstimmen.

Dieser Schritt der ERC beinhaltet die Auswahl und den Aufbau der Sensoren, die Messreihendurchführung (um Daten zu erhalten) und Etikett-Beschriftungstechniken. Ziel ist es relevante und möglichst fehlerfreie Daten von Versuchspersonen für die verschiedenen emotionalen Zustände zu gewinnen. Der Datenerfassungsschritt ist besonders wichtig, da er der erste in der ERC ist und die Ergebnisse aller folgenden Schritte direkt von der Qualität des Datensatzes abhängen.

### 3.6.2 Vorverarbeitung

Verantwortlich: Artur  
- Bereit zum Korrekturlesen.

Das Ziel der Vorverarbeitung ist die "Verbesserung" der Daten für die nachfolgenden Schritte der ERC. In der Regel ist es dadurch besser möglich Muster in Daten erkennen zu können. Vorverarbeitete Daten erreicht man durch Anwendung von z.B. Fil-

terung (Rauschunterdrückung), Normierung oder Reduzierung von unerwünschten oder unbedeutenden Datenteilen.

Fehl am Platz, das kommt erst später.

Normalisierungstechniken wurden auf dem gesamten Datensatz angewendet. Wir haben insbesondere die Standardnormalisierung verwendet, welche den Mittelwert der Daten auf Null setzt und die Einheitsvarianz ergibt [2]. Die Formel für die Standardnormierung lautet:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (1)$$

wobei  $x$  ein Datenpunkt eines Sensorkanales,  $\bar{x}$  ist der Durchschnitt der Gesamtheit für diesen Sensorkanal und  $\sigma$  ist die entsprechende Standardabweichung.

### 3.6.3 Segmentation

Verantwortlich: Artur  
- Bereit zum Korrekturlesen.

Ziel dieses Schrittes ist es, Teile von Daten zu identifizieren, welche wichtige Informationen über die zu erkennenden Emotionen enthalten. Dies geschieht durch Filtern der Daten und Ausschließen von Segmenten, die für das Klassifizierungsproblem nicht relevant sind. Zusätzlich wird die zu verarbeitende Datenmenge reduziert, indem Segmente eines Zeitfensters fester Größe aus den Daten extrahiert werden. Diese Vorgehensweise ist heute in der Praxis besonders wichtig, da sonst hardwarebedingte Einschränkungen die zu verarbeitende Datenmenge begrenzen könnten.

Eventuell fehl am Platz, wenn hier nur Grundlagen und weitere Details erst später.  
- Abklären.

In dieser Projektarbeit wurde ein Schiebefensteransatz (engl. “sliding window approach”) verwendet. Ziel der Methode ist die Segmentierung der vorhandenen Daten in kleinere Einheiten, um die Merkmalsextraktion sowie die anschließende Klassifizierung zu vereinfachen oder gar erst zu ermöglichen. Die Länge des Zeitfensters (engl. “time window”) und des Gleitschritts (engl. “sliding stride”) sind zu bestimmende Parameter (und werden auch als “Hyperparameter” bezeichnet), wobei sich das Zeitfenster auf die feste Größe pro extrahiertem Segment und der Gleitschritt auf den Abstand zu dem Beginn des darauf folgenden Zeitfensters bezieht. Es ist zu beachten, dass sich aufeinanderfolgende Zeitfenster überlappen können, sobald der definierte Gleitschritt kleiner als das Zeitfenster ist.

Die Daten werden auf Zeitstempel-Ebene mit Etiketten beschriftet, basierend auf den von der jeweiligen Versuchsperson ausgefüllten Fragebögen. Jedem Zeitfenster

wird ein Etikett zugeordnet, welches das dominante (d.h. am meisten vorhandene) Etikett der im entsprechenden Fenster enthaltenen Zeitstempel basiert. Es wird davon ausgegangen, dass jedes Zeitfenster nur von einer Emotion belegt ist.

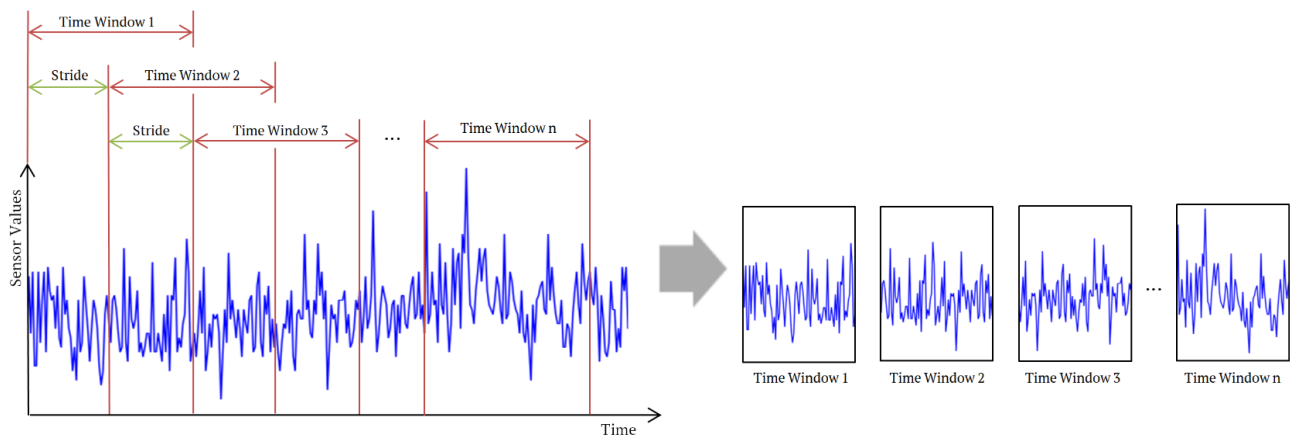


Abbildung 2: Schiebefenster-Segmentierung: Die Daten werden durch ein Zeitfenster fester Größe in kleinere Segmente aufgeteilt. Das Fenster wird mit einem festen Gleichschritt geschoben, um den aufeinanderfolgend Daten-Zeitfenster zu erhalten.

### 3.6.4 Merkmalsextraktion

Verantwortlich: Artur  
- Bereit zum Korrekturlesen.

Hier werden Charakteristiken und Merkmale in den Daten gesucht, die für das Klassifizierungsproblem von möglichst hoher Relevanz sind. Alle nach dem Segmentierungsschritt extrahierte Daten-Zeitfenster kann durch einen Merkmalsvektor (engl. "feature vector") dargestellt werden. Mit Hilfe von Merkmalsvektoren kann ein Klassifikator dann einfacher trainiert werden als nur mit den Rohdaten. Unser Fokus in der Mustererkennung lag vor allem auf der Merkmalsextraktion, da unsere Erfahrungen und frühere Forschungsarbeiten gezeigt haben, dass die Wahl der Merkmale sehr wichtig für die endgültigen Klassifizierungsergebnisse sind. Darüber hinaus wurden noch keine state-of-the-art Merkmale für die Emotionserkennung mit dieser spezifischen Assoziation von eindimensionalen physiologischen Signalen gefunden.

### 3.6.5 Klassifikation

Verantwortlich: Artur  
- LOSOCV im späteren Kapitel erklären.  
- Bereit zum Korrekturlesen.

Ziel des Klassifizierungsschritts ist es, ein Klassifizierungsmodell zu trainieren, das in der Lage ist, Objekte in den Daten (dargestellt durch ihren Merkmalsvektor) in

die entsprechende Klasse zuzuordnen.

Der Datensatz der Merkmalsvektoren, der im vorherigen Schritt des ERC erhalten wurde, wird in einen Trainingsset (engl. "training set") und einen Testset (engl. "testing set") unterteilt, so dass alle Klassen in beiden Sets vorhanden sind. Mit dem Trainingsdaten wird ein Klassifikator erstellt und trainiert. Der so erhaltene Klassifikator wird dann anhand der Daten des Testsets ausgewertet. Es ist wichtig, dass die Trainings- und Testsets unterschiedlich sind (d.h. nicht die gleichen im Daten Trainings- und Testset verwenden), da es sonst in einer Überanpassung (engl. "overfitting") des Klassifikator resultieren kann. Eine Überpassung tritt auf, wenn ein Klassifikator zufällige Schwankungen oder Rauschen in den Trainingsdaten "zu gut" lernt und dann bei neuen, unbekannten Daten deutlich schlechter abschneidet. Der Grund hierfür ist, dass diese gelernten Schwankungen oder Rauschen in den Trainingsdaten keinerlei Relevanz für das eigentliche Klassifizierungsproblem haben.

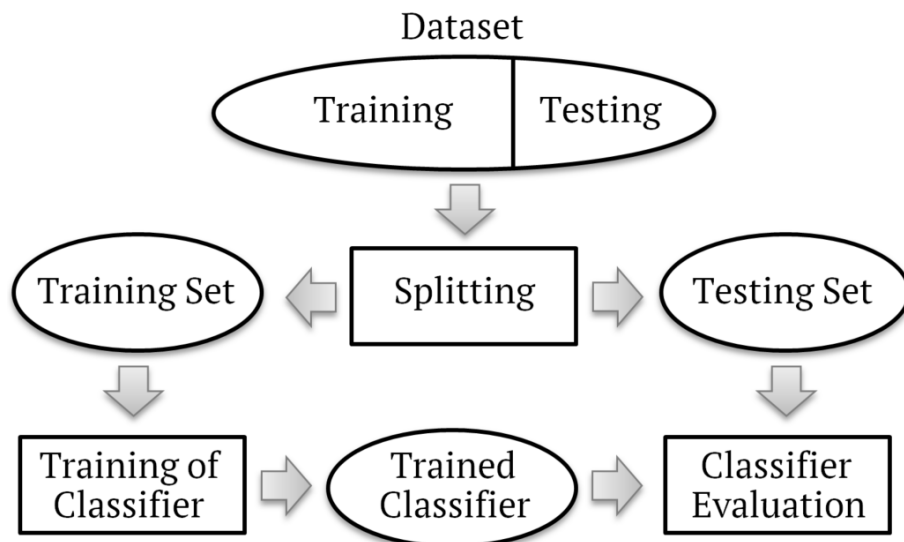


Abbildung 3: Aufteilen eines Datensets in ein Trainings- und ein Testset. Das Trainingsset dient zum Erstellen und Trainieren eines Klassifikators. Die Performance wird mithilfe des Testsets ermittelt und bewertet.

Für die Klassifizierung wurde der state-of-the-art Klassifikator Support-Vector-Machine (SVM) verwendet. SVM ist ein überwachtes Lernansatz, der für binäre Klassifikation oder Regressionszwecke genutzt werden kann. Das Ziel des SVM ist es eine Hyperebene (engl. "hyperplane") zu finden, die zwei Klassen im Raum der Merkmale trennt und gleichzeitig den Abstand (engl. "margin") zwischen der Hyperebene und den nächsten Datenpunkten jeder Klasse maximiert. Für jeden Datenpunkt (dargestellt durch den entsprechenden Vektor von Merkmalen) den wir klassifizieren möchten, wird ein Klassenetiket vergeben, je nachdem, zu welcher Seite der Hyperebene es gehört. Die Methode hat ihren Namen von den "Stützvektoren" (engl. "support vectors"), welche die nächstgelegenen Vektoren beider Klassen zur tren-



nenden Hyperebene sind. Cortes und Vapnik zeigten in [3], dass die Gleichung der optimalen Hyperebene nur von diesen spezifischen Vektoren abhängig ist. Abbildung 3.6.5 zeigt eine optimale Hyperebene in 2D, die beide Klassen perfekt teilt. Datenpunkte werden durch nicht-ausgefüllte blaue Dreiecke bzw. rote Kreise für beide Klassen dargestellt, während Unterstützungsvektoren durch ausgefüllte Punkte und Kreise hervorgehoben werden.

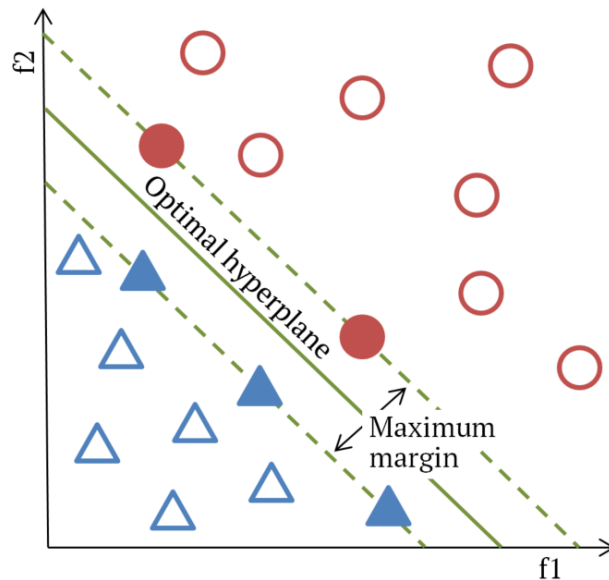


Abbildung 4: Beispiel eines SVM-Klassifikators im zweidimensionalen Merkmalsraum: Die Datenpunkte beider Klassen (dargestellt durch nicht-ausgefüllte blaue Dreiecke und rote Kreise) sind durch eine Hyperebene getrennt, welche die Margin maximiert. Stützvektoren werden als ausgefüllte Dreiecke und Kreise dargestellt.

Es ist anzumerken, dass SVM nur mit zwei Klassen funktioniert, aber sie können zu  $Z$  Klassen verallgemeinert werden. Es gibt zwei Ansätze zur Verallgemeinerung auf  $Z$  Klassen:

- 1 vs 1 besteht darin, Klassifikatoren für jedes Klassenpaar zu trainieren. Auf diese Weise werden  $Z(Z - 1)/2$  Klassifikatoren trainiert, d.h. einen für jedes Klassenpaar.
- 1 vs all besteht darin,  $Z - 1$  Klassen als eine Klasse zu betrachten und die letzte als zweite Klasse anzunehmen, mit der der Klassifikator trainiert wird. Dies wird für jede Klasse wiederholt, was zu  $Z$  Klassifikatoren führt, d.h. einen für jede Klasse.

In der Praxis sind die Daten durch normales SVM fast nie linear trennbar. Einer der Hauptgründe für die Beliebtheit von SVM ist jedoch die Möglichkeit es so genannten Kernel-Tricks. Es basiert auf der Annahme, dass nicht-linear trennbare Daten linear trennbar werden können, wenn sie in einen Raum höherer Dimension projiziert werden. In [3] zeigten Cortes und Vapnik, dass die SVM-Klassifikationsentscheidungsfunktion

als gewichtete Summe von Skalarprodukten zwischen Stützvektoren und dem Vektor der zu klassifizierenden Merkmale ausgedrückt werden kann. Der Kernel-Trick nutzt dies aus, indem er eine Kernelfunktion einführt, die ein skalares Produkt im hochdimensionalen Zielraum repräsentiert. Diese Kernelfunktion erübrigt die eigentliche Zuordnung zwischen dem ursprünglichen und dem hochdimensionalen Feature-Raum. Außerdem ist die Verwendung des Kernel-Tricks oft rechengünstiger als andere Alternativen. Die beiden beliebtesten Kernel sind der lineare und der Radial Basis Function (RBF) Kernel, definiert als:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|) \quad (2)$$

wobei  $\mathbf{x}$  und  $\mathbf{x}'$  Vektoren des Merkmalsraums bezeichnen und  $\gamma$  der Parameter ist, der die "Ausbreitung" (engl. "Spread") des Kernels definiert.

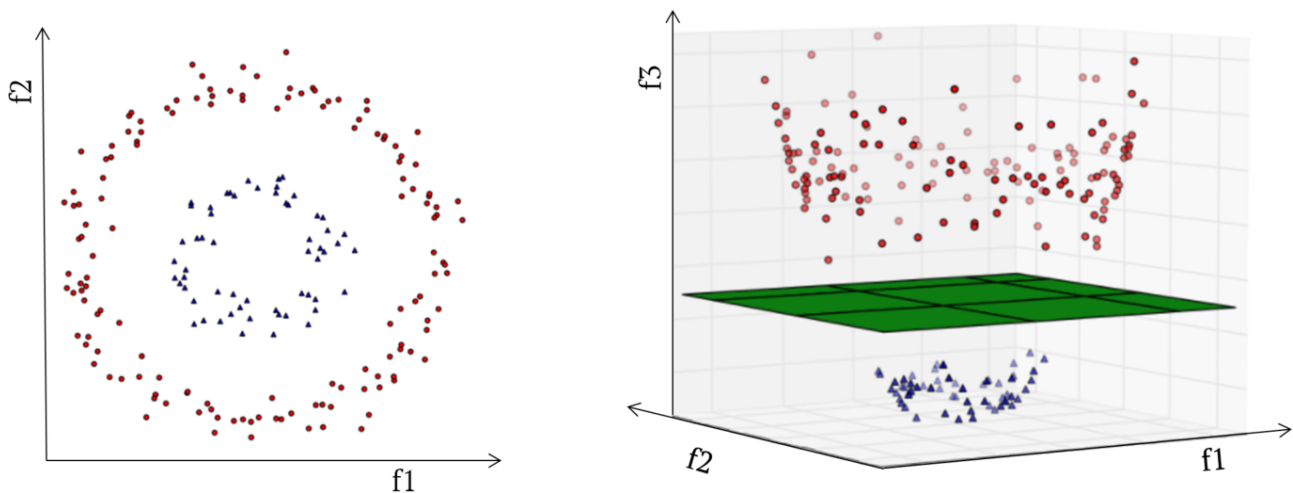


Abbildung 5: Beispiel für die Verwendung des Kernel-Tricks: Nicht-linear trennbare Daten in einem 2D-Merkmalraum (links) können linear trennbar gemacht werden, wenn sie auf 3D projiziert werden (rechts) [4].

Normales SVM verwendet sogenannte hard-margins, die versuchen, eine optimale Hyperebene zu schaffen, die keine Fehlklassifizierungen zulässt. Es ist jedoch oft besser, einige Klassifizierungsfehler zuzulassen, um eine Überanpassung zu verhindern und eine generalisierte Hyperebene zu erhalten. Diese allgemeinere Hyperebene liefert deutlich bessere und zuverlässigere Ergebnisse, wenn sie auf neue und unbekannte Datensätze angewendet wird. Hard-margin SVM kann zu einem Modell führen, das für die Trainingsdaten perfekt funktioniert, aber bei anderen Datensätzen sehr schlecht, weil es seine Trainingsdaten "zu gut" gelernt hat. Aus diesem Grund haben Cortes und Vapnik in [3] eine Variante des Standard-SVM-Klassifikators namens soft-margin SVM (oder auch C-SVM genannt) eingeführt, die eine Fehlklassifizierung von Beispielen beim Erstellen der trennenden Hyperebene toleriert. Der soft-margin Parameter  $C$  wird genutzt um die Anzahl der Fehlklassifikationen festzulegen. Je größer der Wert von  $C$ , desto weniger Fehleinstufungen

sind zulässig. Umgekehrt erlauben kleine Werte von  $C$  mehr Fehlklassifizierungen, um die Verallgemeinerungsfähigkeit des Klassifikators zu verbessern.

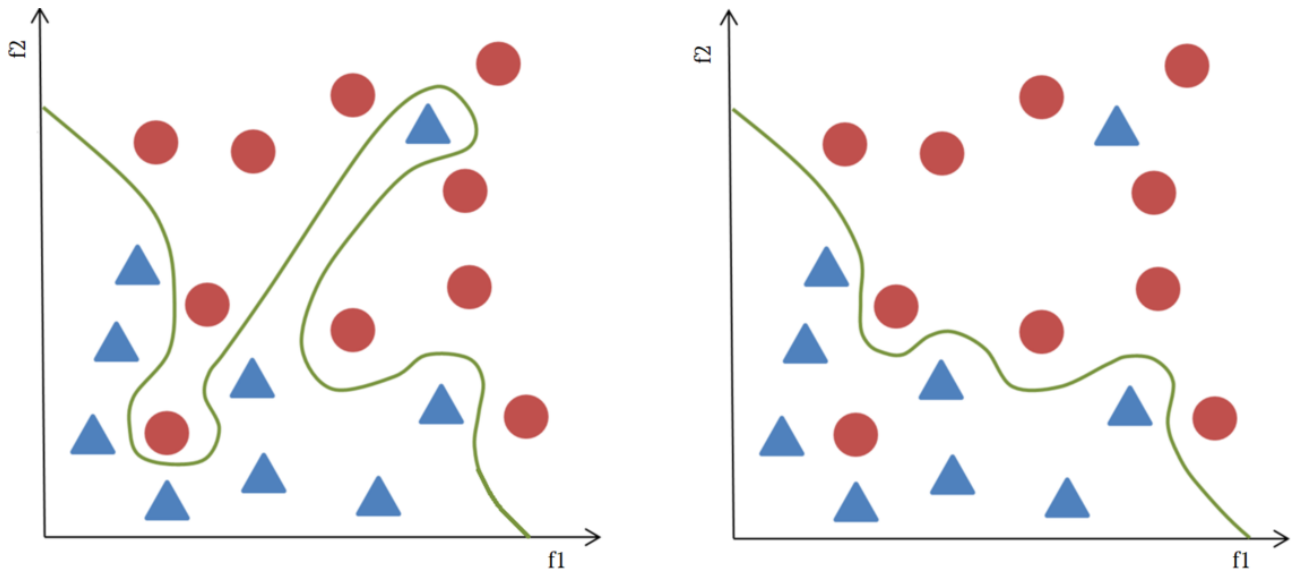


Abbildung 6: Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalraum: Der hard-margin SVM trennt die beiden Klassen perfekt im Gegensatz zum Soft-Margin SVM, der einige Fehlklassifikationen zulässt.

## 4 State-of-the-Art Analyse

Verantwortlich: Jonas

- Next Step: Gliederung erstellen

## Erster Prototype

# 5 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

## 5.1 Anforderungen

## 5.2 Konzept

## 5.3 Hardwareauswahl

### 5.3.1 Auswahlkriterien

### 5.3.2 Festlegung der genutzten Hardware

## 5.4 Hardwarearchitektur

### 5.4.1 GSR-Sensor

### 5.4.2 Temperatur-Sensor

### 5.4.3 Pulsoximeter

### 5.4.4 EEG

### 5.4.5 EOG

### 5.4.6 Datenübertragung

## 5.5 Programmierung

## 5.6 Aufnahme der übertragenen Daten

## 6 Realisierung

Verantwortlich: Jonas, Kevin  
- Next Step: Gliederung erstellen

## 7 Emotionsinduktion

Verantwortlich: Minas

### 7.1 Ablauf

Verantwortlich: Minas

### 7.2 Fragebogen

Verantwortlich: Boris

### 7.3 Szenarien

Verantwortlich: Meryem

#### 7.3.1 Glück

Verantwortlich: Minas

#### 7.3.2 Langeweile

Verantwortlich: Boris

#### 7.3.3 Frustration

Verantwortlich: Meryem

## 8 Messreihe

Verantwortlich: Kevin, Artur



# 9 Mustererkennung

Verantwortlich: Artur

## 9.1 Merkmalsextraktion für Emotionserkennung

### 9.1.1 Hand-gefertigte Merkmale

### 9.1.2 Codebook Approach

# 10 Ergebnisse

Verantwortlich: Artur

## 10.1 Ergebnisse der hand-gefertigten Merkmale

## 10.2 Ergebnisse des Codebook Approach

## 10.3 Analyse der Ergebnisse

## Zweiter Prototype

# 11 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

## 11.1 Anforderungen

## 11.2 Konzept

## 11.3 Hardwareauswahl

### 11.3.1 Auswahlkriterien

### 11.3.2 Festlegung der genutzten Hardware

## 11.4 Hardwarearchitektur

### 11.4.1 GSR-Sensor

### 11.4.2 Temperatur-Senosr

### 11.4.3 Pulsoximeter

### 11.4.4 EEG

### 11.4.5 EOG

### 11.4.6 Datenübertragung

## 11.5 Programmierung

## 11.6 Aufnahme der übertragenen Daten

# 12 Realisierung

Verantwortlich: Jonas, Kevin  
- Next Step: Gliederung erstellen

# 13 Emotionsinduktion

Verantwortlich: Minas

## 13.1 Ablauf

Verantwortlich: Minas

## 13.2 Fragebogen

Verantwortlich: Boris

## 13.3 Szenarien

Verantwortlich: Meryem

### 13.3.1 Glück

Verantwortlich: Minas

### 13.3.2 Langeweile

Verantwortlich: Boris

### 13.3.3 Frustration

Verantwortlich: Meryem

# 14 Messreihe

Verantwortlich: Kevin, Artur

# 15 Mustererkennung

Verantwortlich: Artur

## 15.1 Merkmalsextraktion für Emotionserkennung

### 15.1.1 Hand-gefertigte Merkmale

### 15.1.2 Codebook Approach

# 16 Ergebnisse

Verantwortlich: Artur

## 16.1 Ergebnisse der hand-gefertigten Merkmale

## 16.2 Ergebnisse des Codebook Approach

## 16.3 Analyse der Ergebnisse



## **Dritter Prototype**

# **17 Systementwurf und Konzept**

Verantwortlich: Kevin, Jonas

## **17.1 Anforderungen**

## **17.2 Konzept**

## **17.3 Hardwareauswahl**

### **17.3.1 Auswahlkriterien**

### **17.3.2 Festlegung der genutzten Hardware**

## **17.4 Hardwarearchitektur**

### **17.4.1 GSR-Sensor**

### **17.4.2 Temperatur-Sensor**

### **17.4.3 Pulsoximeter**

### **17.4.4 EEG**

### **17.4.5 EOG**

### **17.4.6 Datenübertragung**

## **17.5 Programmierung**

## **17.6 Aufnahme der übertragenen Daten**

# 18 Realisierung

Verantwortlich: Jonas, Kevin  
- Next Step: Gliederung erstellen

# 19 Emotionsinduktion

Verantwortlich: Minas

## 19.1 Ablauf

Verantwortlich: Minas

## 19.2 Fragebogen

Verantwortlich: Boris

## 19.3 Szenarien

Verantwortlich: Meryem

### 19.3.1 Glück

Verantwortlich: Minas

### 19.3.2 Langeweile

Verantwortlich: Boris

### 19.3.3 Frustration

Verantwortlich: Meryem

## 20 Messreihe

Verantwortlich: Kevin, Artur

# 21 Mustererkennung

Verantwortlich: Artur

## 21.1 Merkmalsextraktion für Emotionserkennung

### 21.1.1 Hand-gefertigte Merkmale

### 21.1.2 Codebook Approach

## 22 Ergebnisse

Verantwortlich: Artur

### 22.1 Ergebnisse der hand-gefertigten Merkmale

### 22.2 Ergebnisse des Codebook Approach

### 22.3 Analyse der Ergebnisse

## Vierter Prototype

### 23 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

#### 23.1 Anforderungen

#### 23.2 Konzept

#### 23.3 Hardwareauswahl

##### 23.3.1 Auswahlkriterien

##### 23.3.2 Festlegung der genutzten Hardware

#### 23.4 Hardwarearchitektur

##### 23.4.1 GSR-Sensor

##### 23.4.2 Temperatur-Sensor

##### 23.4.3 Pulsoximeter

##### 23.4.4 EEG

##### 23.4.5 EOG

##### 23.4.6 Datenübertragung

#### 23.5 Programmierung

#### 23.6 Aufnahme der übertragenen Daten

## 24 Realisierung

Verantwortlich: Jonas, Kevin  
- Next Step: Gliederung erstellen



## 25 Emotionsinduktion

Verantwortlich: Minas

### 25.1 Ablauf

Verantwortlich: Minas

### 25.2 Fragebogen

Verantwortlich: Boris

### 25.3 Szenarien

Verantwortlich: Meryem

#### 25.3.1 Glück

Verantwortlich: Minas

#### 25.3.2 Langeweile

Verantwortlich: Boris

#### 25.3.3 Frustration

Verantwortlich: Meryem

## 26 Messreihe

Verantwortlich: Kevin, Artur

# 27 Mustererkennung

Verantwortlich: Artur

## 27.1 Merkmalsextraktion für Emotionserkennung

### 27.1.1 Hand-gefertigte Merkmale

### 27.1.2 Codebook Approach

## 28 Ergebnisse

Verantwortlich: Artur

### 28.1 Ergebnisse der hand-gefertigten Merkmale

### 28.2 Ergebnisse des Codebook Approach

### 28.3 Analyse der Ergebnisse

## 29 Zusammenfassung und Ausblick

Machen wir später.

### 29.1 Zusammenfassung

### 29.2 Fazit

### 29.3 Ausblick

# Abbildungsverzeichnis

1	Emotion Recognition Chain . . . . .	13
2	Schiebefenster-Segmentierung . . . . .	15
3	Aufteilen eines Datensets in ein Trainings- und Testset . . . . .	16
4	Beispiel eines SVM-Klassifikators im zweidimensionalen Merkmal- raum . . . . .	17
5	Beispiel für die Verwendung des Kernel-Tricks . . . . .	18
6	Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalsraum . . . . .	19

# Tabellenverzeichnis

# Abkürzungen

Bitte alle verwendete Abkürzungen nochmals hier aufführen.

<b>ANN</b>	Artificial Neural Networks
<b>BMBF</b>	Bundesministerium für Bildung und Forschung
<b>BVP</b>	Blood Volume Pulse
<b>CA</b>	Codebook Approach
<b>CRID</b>	Center for Responsible Innovation & Design
<b>C-SVM</b>	Soft-margin Support-Vector-Machine
<b>CSV</b>	Comma-Separated-Values
<b>EEG</b>	Electroencephalography
<b>EOG</b>	Electrooculography
<b>ERC</b>	Emotion Recognition Chain
<b>GSR</b>	Galvanic Skin Response
<b>HR</b>	Heart Rate
<b>LOSOVCV</b>	Leave-One-Subject-Out-Cross-Validation
<b>PG</b>	Projektgruppe
<b>PPG-ir</b>	Photoplethysmography Infrared
<b>PPG-red</b>	Photoplethysmography Red
<b>RBF</b>	Radial Basis Function
<b>SpO2</b>	Pulse Oximetry
<b>SVM</b>	Support-Vector-Machine
<b>VR</b>	Virtual Reality



# Anhang

# Referenzen

- [1] X. Zhu. Semi-supervised learning literature survey, July 2008.
- [2] J. Grus. *Data Science from Scratch*. O'Reilly Media, April 2015.
- [3] C. Cortes and V. Vapnik. *Support-Vector Networks*, volume volume 20. Kluwer Academic Publishers, Boston, September 1995.
- [4] E. Kim. Everything you wanted to know about the kernel trick. September 2013.