



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 Grünwald  
M.Sc. David Krönert  
M.Sc. Tanja Eiler

#### **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 Einleitung</b>	<b>6</b>
1.1 Hintergrund und Motivation . . . . .	6
1.2 ELISE Projektbeschreibung . . . . .	7
1.3 Gliederung dieser Dokumentation . . . . .	8
1.4 Anhang . . . . .	9
<b>2 Organisation</b>	<b>10</b>
2.1 Verantwortungsbereiche . . . . .	10
2.2 Gruppentreffen . . . . .	11
<b>3 Grundlagen</b>	<b>12</b>
3.1 Definition von Emotionen . . . . .	12
3.1.1 Kategoriale Klassifikation . . . . .	14
3.1.2 Circumplex Modell . . . . .	15
3.2 Virtual Reality (VR) . . . . .	16
3.3 Hardware: HTC VIVE . . . . .	18
3.4 Software: Unreal engine 4 . . . . .	19
3.5 Sensoren und biophysiolgische Signale zur Emotionserkennung . . . . .	20
3.5.1 Körpertemperatur-Sensor . . . . .	20
3.5.2 Blood Volume Pulse-Sensor (BVP) . . . . .	21
3.5.3 Messen der Sauerstoffsättigung (SpO <sub>2</sub> ) . . . . .	21
3.5.4 Galvanic Skin Response (GSR) . . . . .	21
3.5.5 Elektroenzephalografie (EEG) . . . . .	21
3.5.6 Elektrookulografie (EOG) . . . . .	21
3.5.7 Analog/Digital-Wandler . . . . .	21
3.6 Kommunikation . . . . .	21
3.7 Grundlagen der Mustererkennung . . . . .	22
3.8 Emotion Recognition Chain . . . . .	24
3.8.1 Datenerfassung . . . . .	24
3.8.2 Vorverarbeitung . . . . .	24
3.8.3 Segmentation . . . . .	25
3.8.4 Merkmalsextraktion . . . . .	25
3.8.5 Klassifikation . . . . .	25
<b>4 State-of-the-Art Analyse</b>	<b>29</b>
<b>I Erster Prototype</b>	<b>30</b>

<b>5 Systementwurf und Konzept</b>	<b>30</b>
5.1 Anforderungen . . . . .	30
5.2 Konzept . . . . .	30
5.3 Hardwareauswahl . . . . .	30
5.3.1 Auswahlkriterien . . . . .	30
5.3.2 Festlegung der genutzten Hardware . . . . .	30
5.4 Hardwarearchitektur . . . . .	30
5.4.1 GSR-Sensor . . . . .	30
5.4.2 Temperatur-Sensor . . . . .	30
5.4.3 Pulsoximeter . . . . .	30
5.4.4 EEG . . . . .	30
5.4.5 EOG . . . . .	30
5.4.6 Datenübertragung . . . . .	30
5.5 Programmierung . . . . .	30
5.6 Aufnahme der übertragenen Daten . . . . .	30
<b>6 Software</b>	<b>31</b>
<b>7 Emotionsinduktion</b>	<b>32</b>
7.1 Ablauf . . . . .	32
7.2 Fragebogen . . . . .	32
7.3 Szenarien . . . . .	34
7.3.1 Glück . . . . .	34
7.3.2 Langeweile . . . . .	36
7.3.3 Frustration . . . . .	36
<b>8 Messreihe</b>	<b>38</b>
<b>9 Mustererkennung</b>	<b>39</b>
9.1 Datenerfassung . . . . .	39
9.2 Vorverarbeitung . . . . .	39
9.3 Segmentation . . . . .	39
9.4 Merkmalsextraktion . . . . .	40
9.4.1 Handgefertigte Merkmale . . . . .	40
9.4.2 Codebook Approach . . . . .	43
9.4.3 Deep Neural Networks (DNN) . . . . .	47
9.4.4 Merkmals Auswahl . . . . .	50
9.5 Klassifikation . . . . .	52
<b>10 Ergebnisse</b>	<b>53</b>
10.1 Ergebnisse der hand-gefertigten Merkmale . . . . .	53
10.2 Ergebnisse des Codebook Approach . . . . .	54

10.3 Ergebnisse der Deep Neural Networks . . . . .	54
10.4 Analyse der Ergebnisse . . . . .	55
<b>II Zweiter Prototype</b>	<b>59</b>
<b>11 Systementwurf und Konzept</b>	<b>59</b>
11.1 Anforderungen . . . . .	59
11.2 Konzept . . . . .	59
11.3 Hardwareauswahl . . . . .	59
11.3.1 Auswahlkriterien . . . . .	59
11.3.2 Festlegung der genutzten Hardware . . . . .	59
11.4 Hardwarearchitektur . . . . .	59
11.4.1 GSR-Sensor . . . . .	59
11.4.2 Temperatur-Senosr . . . . .	59
11.4.3 Pulsoximeter . . . . .	59
11.4.4 EEG . . . . .	59
11.4.5 EOG . . . . .	59
11.4.6 Datenübertragung . . . . .	59
11.5 Programmierung . . . . .	59
11.6 Aufnahme der übertragenen Daten . . . . .	59
<b>12 Software</b>	<b>60</b>
<b>III Dritter Prototype</b>	<b>61</b>
<b>13 Systementwurf und Konzept</b>	<b>61</b>
13.1 Anforderungen . . . . .	61
13.2 Konzept . . . . .	61
13.3 Hardwareauswahl . . . . .	61
13.3.1 Auswahlkriterien . . . . .	61
13.3.2 Festlegung der genutzten Hardware . . . . .	61
13.4 Hardwarearchitektur . . . . .	61
13.4.1 GSR-Sensor . . . . .	61
13.4.2 Temperatur-Senosr . . . . .	61
13.4.3 Pulsoximeter . . . . .	61
13.4.4 EEG . . . . .	61
13.4.5 EOG . . . . .	61
13.4.6 Datenübertragung . . . . .	61

13.5 Programmierung . . . . .	61
13.6 Aufnahme der übertragenen Daten . . . . .	61
<b>14 Software</b>	<b>62</b>
<b>15 Emotionsinduktion</b>	<b>63</b>
15.1 Ablauf . . . . .	63
15.2 Fragebogen . . . . .	64
15.3 Szenarien . . . . .	66
15.3.1 Glück . . . . .	66
15.3.2 Langeweile . . . . .	73
15.3.3 Frustration . . . . .	75
<b>16 Messreihe</b>	<b>77</b>
<b>17 Mustererkennung</b>	<b>78</b>
<b>18 Ergebnisse</b>	<b>78</b>
<b>19 Alternative Lösungen</b>	<b>79</b>
19.1 Kalibrierung . . . . .	79
19.2 Plan B . . . . .	79
<b>20 Zusammenfassung und Ausblick</b>	<b>80</b>
20.1 Zusammenfassung . . . . .	80
20.2 Ausblick . . . . .	80
<b>Abbildungsverzeichnis</b>	<b>82</b>
<b>Tabellenverzeichnis</b>	<b>84</b>
<b>Abkürzungen</b>	<b>85</b>
<b>Anhang</b>	<b>89</b>

# 1 Einleitung

Verantwortlich: Minas  
- One/two missing sentences

## 1.1 Hintergrund und Motivation

RfP

Die Gesellschaft befindet sich seit mehreren Jahren in einem beschleunigten Wandel der das Leben verändert. Dieser Wandel wurde durch die Automation und die Digitalisierung, welche sich beide ergänzen verstärkt und wird auch als vierte industrielle Revolution bezeichnet, die zu Veränderungen im Alltag als auch in der Wirtschaft geführt hat. Durch die Digitalisierung mussten viele Branchen wie die Musik, Einzelhandel und die Logistik & Versand Industrie umstrukturieren oder wurden wie zum Beispiel die Schreibmaschinenindustrie vollständig beseitigt. Somit erscheinen täglich technologische Innovationen, die im Internet, Fernsehen oder in Zeitschriften veröffentlicht werden. Ein Bereich sind die affektiven Technologien. Dieser Bereich hat sich als interdisziplinäres Forschungsfeld etabliert und untersucht die Interaktion zwischen Mensch und Maschine, wobei Emotionen im Mittelpunkt stehen und lässt sich in zwei Systeme unterteilen. Zu einem die emotionssensitiven Technologien, womit Maschinen verstehen was Menschen fühlen und zum anderen die "Emotional Robotic"- Technologien, die einen Roboter menschenähnlicher erscheinen lassen.

Um die Forschung im Bereich emotionssensitiven Technologien voranzutreiben wurde ein Forschungsprojekt mit den Namen "ELISE: Entwicklung von interaktiven und emotionssensitiven Lernsystemen zur Kompetenzerhaltung im Geschäftsprozessmanagement" ins Leben gerufen (siehe Kapitel 1.2).

Der Lehrstuhl Medizinische Informatik und Mikrosystementwurf entwickelt im Rahmen des Gesamtprojektes ein Sensorsystem, welches die Vital-, Elektroenzephalografie-, Elektrookulografie- und galavanische Hautreaktionwerte aufzeichnet. Diese werden dann vom Lehrstuhl für Mustererkennung ausgewertet. Die Lerninhalte der Hauptanwendung des ELISE-Projekts werden daraufhin an Emotionen und Gemütslagen der Lernenden wie Glück, Langeweile, Frustration auf Basis von biomedizinischer Daten angepasst, um so den individuellen Erfolg des Lernenden zu erhöhen.

Diese Projektarbeit befasst sich mit dem Entwurf eines kompakten mikrocontrollergestützten Systems zur Emotionserkennung in einer Virtual-Reality-Umgebung.

Die Projektarbeit baut auf eine vorher am Lehrstuhl geschriebene Master Thesis auf[1] und ist eine Zusammenarbeit mit dem Lehrstuhl für Mustererkennung. Sie beinhaltet den Aufbau und die Programmierung des Mikrocontrollers, welches zur Kommunikation der verschiedenen biomedizinischen Sensoren dient, die Entwicklung der VR-Umgebung, die Schnittstellen zwischen Hardware und Software und die Speicherung der Rohdaten (Vital-, Elektroenzephalografie-, Elektrookulografie- und galavanische Hautreaktionwerte). Diese Rohdaten wurden anhand von Messreihen an 88 Probanden gewonnen und wurden dem Lehrstuhl für Mustererkennung zur Verarbeitung übergeben. Die Ergebnisse der Verarbeitung werden auch aufgeführt.

## 1.2 ELISE Projektbeschreibung

RfP

ELISE ist ein Verbundprojekt, welches die Entwicklung eines interaktiven und emotionssensitiven Lernsystems zur Kompetenzentwicklung im Bereich der Geschäftsprozessmanagement plant. Hierfür kamen fünf Partner des Forschungskollegs (FoKoS) der Universität Siegen zusammen – der Lehrstuhl für Wirtschaftsinformatik & Center for Responsible Innovation & Design, die Forschungsgruppe Research Group for Pattern Recognition, der Lehrstuhl Medizinische Informatik und Mikrosystementwurf der Universität Siegen, der Spieleentwickler Limbic Entertainment GmbH und der Softwarehersteller Software AG. Zusammen befassen sie sich mit einem interaktiven und emotionssensitiven Lernsystems in Form eines Spiels, das in einer virtuellen Umgebung erfolgt. Zudem befasst sich das Projekt mit der Auswirkung von solcher Systeme hinsichtlich ethischer und gesellschaftlicher Aspekte auf die Akzeptanz potenzieller Nutzerinnen und Nutzer. Abbildung 1 zeigt das vorhaben, welches durch das Projekt Elise verwirklicht werden soll.

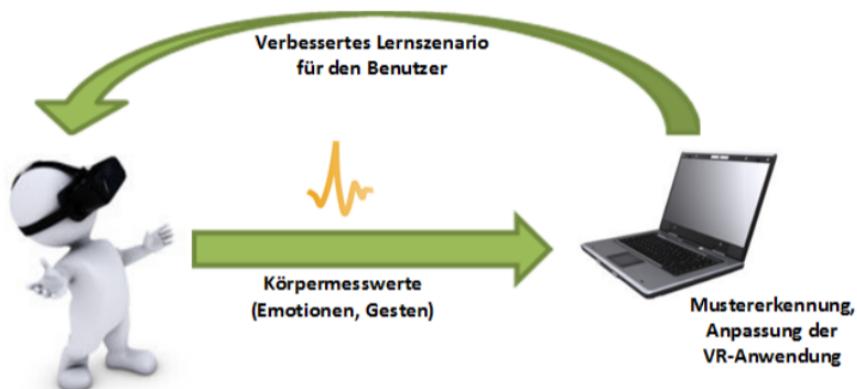


Abbildung 1: Grobe Übersicht des Gesamtprojekts[1].

Der Lehrstuhl Medizinische Informatik und Mikrosystementwurf entwickelt im Rahmen des Gesamtprojektes ein Sensorsystem, welches die Vital-, Elektroenzephalografie-, Elektrookulografie- und galavanische Hautreaktionwerte aufzeichnet. Diese werden dann vom Lehrstuhl für Mustererkennung ausgewertet.

Die Lerninhalte der Hauptanwendung des ELISE-Projekts werden daraufhin an Emotionen und Gemütslagen der Lernenden wie Glück, Langeweile, Frustration auf Basis von biomedizinischer Daten angepasst, um so den individuellen Erfolg des Lernenden zu erhöhen.

## 1.3 Gliederung dieser Dokumentation

Verantwortlich: Minas

Die Dokumentation ist in mehreren Teil gegliedert: eine Einführung, die Entwicklung des ersten Prototypen, die Entwicklung des zweiten Prototypen, die Entwicklung des dritten/finalen Prototypen und ein Schlussteil.

Einführung:

Die Dokumentation fängt mit einer Einleitung an, die den Hintergrund und die Motivation, die Elise Projektbeschreibung, die Gliederung dieser Dokumentation und den mit der Dokumentation übergebenen Anhang erläutert (geschrieben von Minas Michail). Der nächste Punkt "Organisation" beinhaltet zum einen die Verantwortungsbereiche der aufgeteilten Aufgaben innerhalb der Gruppe und zum anderen wird erläutert wie und wann die Gruppentreffen stattgefunden haben (geschrieben von Artur Piet). Daraufhin werden die Grundlagen übermittelt, um späteres geschehen in der Dokumentation besser nach voll ziehen zu können. Der Punkt Grundlagen enthält die Definition von Emotionen (geschrieben von Arnaud Eric Toham Waffo), die Grundlagen der verwendeten Hardware und Software (geschrieben von Arnaud Eric Toham Waffo), die Grundlagen der verwendeten Sensoren und biophysiologischen Signale (geschrieben von Kevin Orth), die Grundlagen der Kommunikation zwischen den gewonnenen Sensordaten und dem Board (geschrieben von Kevin Orth & Jonas Pöhler), die Grundlagen der Mustererkennung (geschrieben von Artur Piet) und die Grundlagen zur Emotionserkennung (geschrieben von Artur Piet). Nach den Vermittlungen der Grundlagen

Entwicklung des ersten Prototypen:

Die Entwicklung des ersten Prototypen umfasst den Systementwurf und das Konzept der Hardware (geschrieben von Kevin Orth), die entwickelte Software zur Kommunikation (geschrieben von Kevin Orth & Jonas Pöhler), die Emotionsinduktion (geschrieben von Meryem Dural, Boris Kamdem & Minas Michail), die Messreihe (geschrieben von Kevin Orth & Artur Piet), die Mustererkennung (geschrieben von Artur Piet) und die gewonnenen Ergebnisse (geschrieben von Artur Piet).

Entwicklung des zweiten Prototypen:

Die Entwicklung des zweiten Prototypen umfasst den Systementwurf und das Konzept der Hardware und die modifizierte Software zur Kommunikation (beides geschrieben von Kevin Orth & Jonas Pöhler). Parallel zur Fertigstellung des zweiten Prototypen, lief die Entwicklung der Emotionsinduktion für den dritten Prototypen.

Entwicklung des dritten Prototypen:

Die Entwicklung des dritten Prototypen umfasst den Systementwurf und das Konzept der Hardware (geschrieben von Kevin Orth), die entwickelte Software zur Kommunikation (geschrieben von Kevin Orth & Jonas Pöhler), die Emotionsinduktion (geschrieben von Meryem Dural, Boris Kamdem & Minas Michail), die Messreihe (geschrieben von Kevin Orth & Artur Piet), die Musterkennung (geschrieben von Artur Piet) und die gewonnenen Ergebnisse (geschrieben von Artur Piet).

Schlussteil:

Der Schlussteil beinhaltet das Kapitel Alternative Lösungen aufgeteilt in Kalibrierung (geschrieben von Jonas Pöhler) und Plan B (geschrieben von Meryem Dural) und eine Zusammenfassung (geschrieben von Meryem Dural) sowie einen Ausblick (geschrieben von X).

## **1.4 Anhang**

## 2 Organisation

Verantwortlich: Artur  
- RfP

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 anderen 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.).

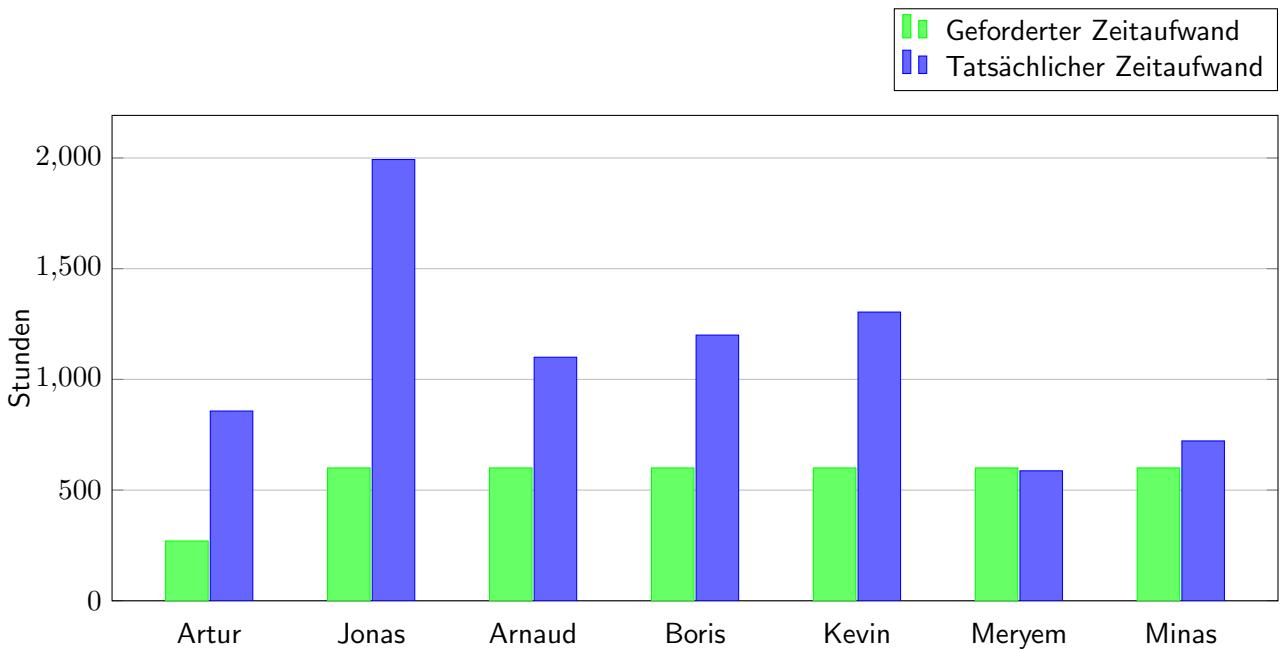
Die Laufzeit der Projektgruppe wurde auf etwa 1 Jahr gesetzt, wobei das Kick-Off Meeting am 23.10.2017 stattfand. Der geforderte individuelle Zeitaufwand aller Gruppenmitglieder entspricht der jeweils im Modulhandbuch des Studienganges definierten Leistungspunkte für die Projektarbeit, also 600 Stunden 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 tatsächlichen Zeitaufwand.

### 2.1 Verantwortungsbereiche

RfP

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, Berichtserstellung und Sprecher der PG



- Jonas Pöhler: Hardware, Webanbindung, VR Szenarien und Stellv. Sprecher der PG
- Arnaud Eric Toham Waffo: Plan B
- Boris Kamdem: Langeweile-Szenario sowie Fragebogen in VR und Plan B
- Kevin Orth: Komplette Hardware
- Meryem Dural: Frustations-Szenario in VR und Plan B
- Minas Michail: WarmUp- und Glücks-Szenario, Zusammenführung / Verknüpfung der VR-Szenarien in einer Unreal-Datei, Plan B

## 2.2 Gruppentreffen

### RfP

Die wöchentliche Gruppentreffen finden 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 erklärt kurz woran letzte Woche gearbeitet wurde, was die aktuellen Herausforderungen und Probleme sind und was genau als nächstes geplant ist. Ziel ist es den Austausch und die Kommunikation unter den Teammitgliedern und den Projektleitern zu fördern, damit alle Teilnehmer auf dem selben Wissenstand sind, da einzelnen Aufgaben durchaus großen Einfluß auf Tätigkeiten von anderen Mitgliedern haben können.

# 3 Grundlagen

Verantwortlich: Arnaud  
- RfP

Mit diesem Kapitel werden Einblicke in die notwendige Grundlagen zur Realisierung unserer Arbeit gegeben. Zuerst werden wir uns mit der Thematik Emotion (Definition und Klassifikation) beschäftigen. Im Weiteren werden "Virtual Reality" (beziehungsweise HTC Vive VR-Brille) genauer erklärt. Ebenfalls kommt eine Erläuterung der Sensoren und Biophysiolgische Signale, welche für unsere Arbeit von Bedeutung sind. Näher wird auf die Kommunikation zwischen verschiedenen Sensoren eingegangen. Dazu kommt auch eine Erklärung wie die Datenerfassung(Messreihe) stattfinden sollte. Danach befassen wir uns mit der Grundlagen der Mustererkennung. Ein Berichtsteil über die "Emotion Recognition chain" schließt das Kapitel "Grundlagen" ab.

## 3.1 Definition von Emotionen

Verantwortlich: Arnaud  
- RfP

Der Begriff "Emotion" wird zwar weltweit (in fast allen Sprachen) und von alle Menschen verwendet (die soziale oder intellektuelle Ebene spielt keine Rolle), ist aber relativ schwer zu definieren. Dieses Paradox wurde bereits im folgenden Zitat explizit erwähnt: "Everybody knows what an emotion is, until asked to give a definition." [2] von Fehr und Russell, zwei amerikanischer Emotionsforscher (Psychologe). Noch überraschender ist die Schwankung in der Definition dieses Begriffs "Emotion" im Laufe der Zeit: Allein die englischsprachige Experimentalpsychologie[3] liefert zwischen 1872 und 1980 mehr als 92 verschiedene Definitionen. Wir verstehen daher, dass es schwierig wäre, zu versuchen, diese Definition zu formulieren. Wie können wir also die Schwierigkeit, eine Definition für einen solchen gemeinsamen Begriff zu finden, erklären? Man sollte nicht vergessen werden, dass es sich um einen eher abstrakten Begriff handelt und daher ist die Emotion sehr subjektiv. Neben diesem abstrakten Aspekt ist auch anzumerken, dass sich der Begriff der Emotion auch auf viele Bereiche bezieht, die sich ebenso voneinander unterscheiden wie sie variieren: z.B. Literatur, Philosophie, Psychologie usw. Neben diesem multidisziplinären Charakter, der die Pluralität der Vorstellungen und Ansätze in jedem Definitionsversuch erklären könnte, ist es auch notwendig, die Variationen von Sprachen, Perioden und sogar Kulturen hinzuzufügen.

Es wird uns allein mit dieser Arbeit (und das ist auch nicht von uns erwartet) nicht möglich sein, alle Fragen im Zusammenhang mit der Definition des Begriffs "Emotion" zu betrachten, aber wir werden einige konkrete Beispiele vorstellen, um

die Komplexität zu veranschaulichen, die sich aus der Suche nach einer Definition von Emotion ergeben kann. Die antiken Philosophen[4] waren die ersten, die sich mit Emotionen und ihren Einflüssen auf den Alltag beschäftigen haben. Tatsächlich nahmen Stoiker wie Zeno und Plato bereits 370 n. Chr. Emotionen als eine Krankheit der Seele wahr, die für sie ein Hindernis für denjenigen war, der denken wollte. Platon geht mit seiner Allegorie von der Höhle tiefer, indem er alles was emotional ist mit alle vernünftig (verständlich) kontrastiert: das heiß Emotion und Vernunft gehen nicht zusammen. Descartes und Aristoteles vervollständigen Platons Beobachtungen, indem sie in die Analyse von Emotionen eine Dualität (positiv und negativ) der Perzeption einbringen. Aristoteles glaubt, dass alles, was das Leben auf positive Weise beeinflusst, durch positive Emotionen bedingt ist, und Descartes glaubt, dass Emotionen für unser Überleben unerlässlich sind und dass die Schwäche eines Menschen eng mit der Fähigkeit der Seele verbunden ist, seine Emotionen zu kontrollieren. Charles Darwin in seinem präsentierte weitere ebenso faszinierende neue Elemente über Emotionen, ohne den bisherigen Beobachtungen zu widersprechen[5]. Er verallgemeinert die Emotionen für alle Kulturen und fand sogar Ähnlichkeit mit Tieren. Der Darwin präsentiert Emotionen als Körpersignale (oder Reaktionen) auf äußere Handlungen(Externe Ereignisse), begleitet von spezifischen körperlichen Äußerungen wie: Gesichtsausdrücke, Gesten und oft Geräusche, die alle spezifisch je nach Emotionen sind.

So können wir weiterhin andere berühmte wissenschaftliche Namen wie William James[6], Walter Cannon[7], Stanley Scharter[8] usw. nennen, die zu unterschiedlichen Zeiten und an unterschiedlichen Orten das Thema untersucht haben, mit ebenso relevanten wie unterschiedlichen Schlussfolgerungen, aber die Beobachtung bleibt die gleiche, die Definition bleibt unbeständig. Neben den Schwierigkeiten, die sich aus den oben genannten unterschiedlichen Wahrnehmungen ergeben, trägt die tägliche missbräuchliche Verwendung bestimmter Begriffe wie Gefühl, Affekt, Stimmung, Gefühl (um nur einige zu nennen) als Synonym für Emotionen dazu bei mehr Verwirrung in das Verständnis von Emotionen. Diese Begriffe beziehen sich zwar auf Gemütszustände, sind aber jedoch nicht Synonym von Emotion[3]. Trotz des fehlenden Konsenses über die Frage der Definition dieses Begriffs gibt es jedoch Elemente, die in allen Definitionsversuchen wiederkehren, nämlich:

- das Vorhandensein eines Auslösers;
- den psychischen Zustand des Subjekts;
- einen bestimmten Körpersausdruck;
- eine physiologische Reaktion (Herzfrequenz, Atmung, Schwitzen ...);
- eine bestimmte Qualität, Intensität und Dauer.

Basierend auf den obigen Elementen können wir daher zu dem Schluss kommen,

dass wir zwar nicht genau definieren können, was eine Emotion ist, können wir aber jedoch jede davon mit bestimmten spezifischen Elementen in Verbindung setzen: einen Stimulus (der der Auslöser der Emotion ist), eine physiologische und körperliche Reaktion. Die Existenz mehrerer verschiedener Emotionen lässt sich gerade wegen der Spezifität dieser Elemente vermuten. Im nächsten Teil werden wir versuchen, die verschiedenen Emotionen zu klassifizieren, mit einem besonderen Fokus auf diejenigen, die wir zu induzieren versuchen.

Wie bei der Definition von Emotionen gibt es keine einheitliche Klassifizierung von Emotionen. Und wie bei den versuchten Definitionen gab es im Laufe der Zeit mehrere vorgeschlagene Klassifikationen, die sich auch im Laufe der Zeit entwickelt haben. Diese Mehrheit von Klassifikationen ist vermutlich eine Folge des Mangels an Konsens über die Definition selbst. Es gibt jedoch zwei Arten von Klassifikationen, die vor allem unterscheidet und durchgesetzt haben: eine dimensionale Klassifikation[4] und eine kategoriale Klassifikation[9].

### 3.1.1 Kategoriale Klassifikation

Angetrieben von Charles Darwin, (der auch der erste war, der grundlegende Emotionen identifizierte) diese Vorgehensweise ist die von Emotionsforscher mit einer evolutionären Ansatz. Diese Klassifikation basiert auf der Existenz von einer Reihe von sogenannten Grundemotionen oder primären Emotionen. Diese Emotionen wären in einer begrenzten Anzahl und ihre Entwicklung wurde stark von der Evolution beeinflusst. Wie viele Dinge in Bezug auf die Emotionen in allgemein, variiert die Anzahl von Grundemotionen von einem Emotionsforscher zu einem anderen (siehe Tabelle 1).

Forscher	Basisemotionen
Gray (1982)	Furcht, Freude, Ärger
Panksepp (1982)	Furcht, Erwartung, Ärger, Panik
Tomkins (1984)	Furcht, Freude, Ärger, Verzweiflung, Ekel, Überraschung, Interesse, Scham, Zufriedenheit
Plutchik (1980)	Furcht, Freude, Ärger, Traurigkeit, Ekel, Überraschung, Akzeptanz, Erwartung
Arnold (1960)	Furcht, Liebe, Ärger, Traurigkeit, Hass, Hoffnung, Begehrungen, Mut, Niedergeschlagenheit, Verzweiflung, Widerwille
Oatley & Johnson-Laird (1987)	Furcht, Glück, Ärger, Traurigkeit, Ekel
Ekman (1982)	Furcht, Freude, Ärger, Traurigkeit, Ekel, Überraschung
Izard (1987)	Furcht, Freude, Ärger, Traurigkeit, Ekel, Überraschung, Interesse, Verachtung

Tabelle 1: Vergleich einige Basisemotions-Theorien[9].

Obwohl sie sich nicht über die Anzahl der Grundemotionen einig sind, sind sich die Forscher einig über die Existenz komplexerer Emotionen, die die Kombination mehrerer primärer Emotionen wären. Basierend auf diesem Ansatz wurden mehrere Theorien entwickelt, um Emotionen darzustellen, von denen eine der

bekannte das Plutchik-Modell ist. Plutchiks Modell verwendet ein radförmiges Diagramm (siehe Abbildung 2) und verschiedene Farben, um jede Emotion darzustellen.

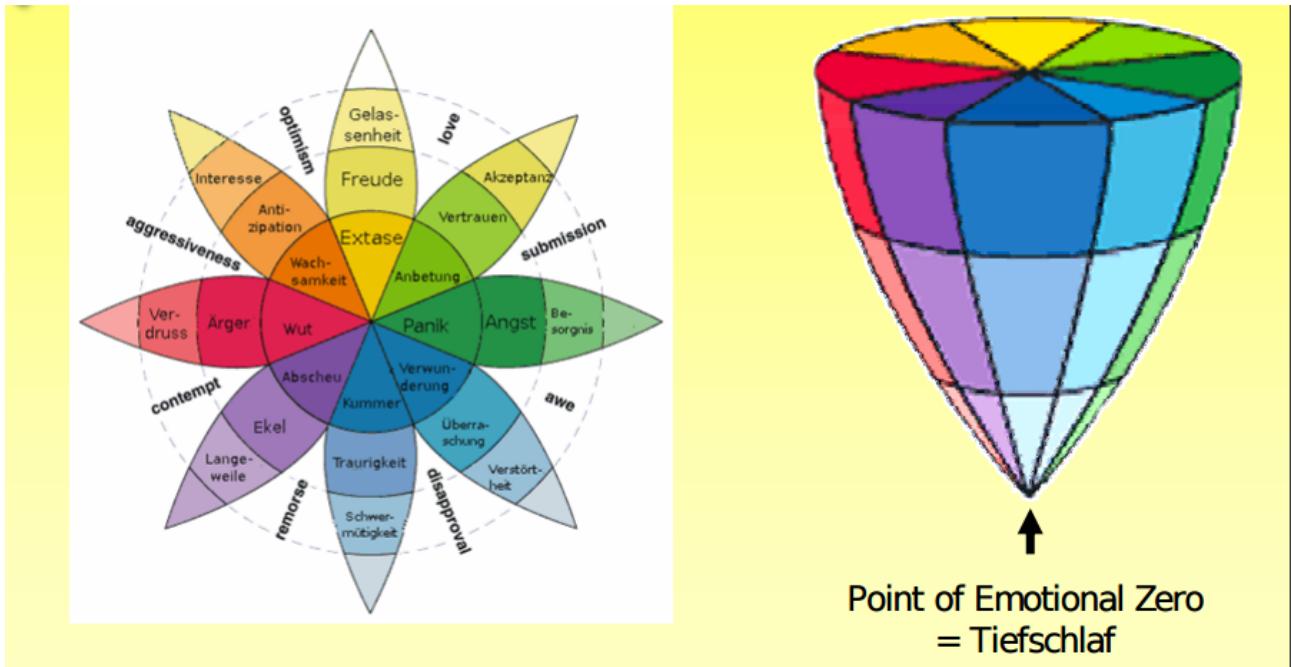


Abbildung 2: Einordnung von Emotion nach Plutchik.

Diese Theorie fundiert sich auf das acht primäre Emotionen mit einer klare trennung (bzw unterschied) zwischen primären, sekundären und tertiären Emotionen. Die verbindet jede primäre Emotion mit spezifischen Motivationssystem und Verhaltendstendenzen zur Bewältigung grundlegender adaptiver Probleme (siehe Tabelle 2).

Subjektiv	Verhalten	Funktion
Angst, Entsetzen	Rückzug, Flucht	Schutz
ärger, Wut	Angriff, Beißen	Zerstörung
Freude, Ekstase	Paarung, Besitz	ergreifen Fortpflanzung
Traurigkeit, Trauer	Weinen, Bitte um Hilfe	Reintegration
Akzeptanz, Anbetung, Vertrauen	Paarbildung, Pflege	Zusammengehörigkeit, Bindung
Ekel, Abscheu	Sich übergeben	Ablehnung, Zurückweisung
Erwartung, Antizipation	Untersuchen	Exploration, Erkundung
überraschung	Innehalten, Einfrieren	Orientierung

Tabelle 2: Einige Basisemotionen jeweils mit Verhalten und Funktion[9].

### 3.1.2 Circumplex Modell

Dieser von Wundt[9] initiierte Ansatz geht von dem Gedanken aus, dass die emotionale Erfahrung in einem mehrdimensionalen Raum dargestellt werden kann. Dieser Zerlegung der emotionale Erfahrung sollte es ermöglichen, eine genaue Analogie zwischen Emotion und Körpераusdruck (Gesichtsausdrücke) herzustellen. Dieser

Ansatz wurde daher den Vorteil, dass sie die Tür zu einer möglichen Quantifizierung der emotionalen Erfahrung öffnet. Die Idee von Wundt (siehe Abbildung 3) war einer dreidimensionalen Zerlegung(Lust-Unlust, Spannung-Entspannung (Lösung), Beruhigung (Ruhe)-Erregung siehe) der emotionalen Erfahrung.

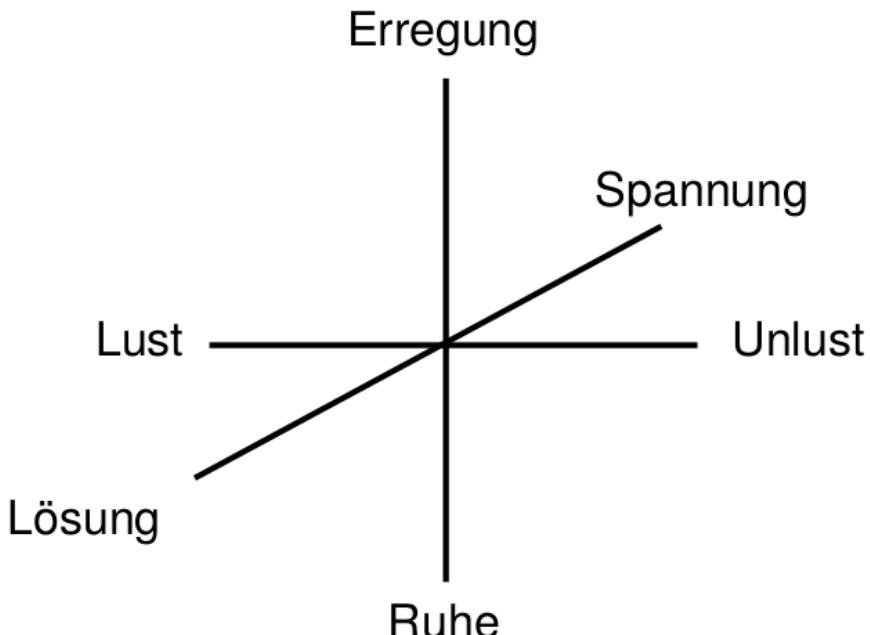


Abbildung 3: Einordnung von emotionale Erfahrungsprozess nach Wundt[9].

Die Frage nach der Anzahl der Dimensionen, die zur Darstellung der emotionalen Erfahrung notwendig sind, wird jedoch zu mehreren Theorien führen. Allerdings schlug Russell[9] eine zweidimensionale Darstellung mit sechs primären Emotionen nach Ekam vor (siehe Tabelle 1). Emotionen werden also dank dieses Modells (siehe Abbildung 4) eine horizontale Komponente haben: Valenz (Freude/Verdruss) und eine vertikale Komponente: Erregung (Aktivierung).

Die Valenz unterscheidet positive Emotionen von negativen Emotionen, und die Erregung informiert über körperliche Erregung, die man durch die Anzahl von physiologischen Reaktion feststellen kann. Jede Emotion lässt sich als Kombination dieser beiden Parameter darstellen was für eine mathematische auswertung sehr hilfreich sein könnte. Dieses Modell hat viel Erfolg gehabt, da er die Darstellung von einer Unendlichkeit von Emotionen erlaubt hat. Trotz der unterschiedlichen Anzahl von Dimensionen von verschiedenen Autoren vorgeschlagen, die beiden Dimensionen von Russell (Valenz und Erregung) sind Faktoren die in fast alle Modelle dieses Ansatzes auftreten.

## 3.2 Virtual Reality (VR)

Verantwortlich: Arnaud  
- RfP

Der Begriff Virtual Reality ist ein Begriff, der sich auf eine Vorrichtung bezieht,



Abbildung 4: Einordnung von emotional Erfahrung nach Russell[9].

mit denen eine Umgebung von einer Maschine digital simuliert wird. Je nach verwendeter Technologie ermöglicht es dem Nutzer, ein virtuelles Universum mit seinen verschiedenen Sinnen zu erleben: am häufigsten Sehen, aber auch Berühren, Hören, Riechen. Dieses Konzept sollte es einer Person ermöglichen, ein Immersives Erlebnis zu erleben und eine sensomotorische Aktivität in einer künstlichen Welt durchzuführen. Dieses virtuelle Universum kann eine Reproduktion der realen oder einer völlig imaginären Welt sein. Virtual Reality ist von der Augmented Reality, die virtuelle Elemente in einer realen Umgebung hinzugefügt, zu unterscheiden. Obwohl das Feld der Vorliebe für Virtual Reality Videospiele sind, kann es auch und sogar schon in mehreren Bereichen gelten, nämlich[10]:

- Architektur, in der die Frage der Modellierung, Visualisierung und Simulation wesentlich ist, kann diese Technologie Modelle in realen Maßstab visualisieren, also eine künstliche Umgebung, die mit der realen Umgebung identisch ist;
- Medizin, wo einige Ärzte diese Technologie mit ihren Patienten einsetzen, um z.B. Angststörungen und Phobien zu behandeln; auch für die Ausbildung einiger Medizinstudenten: Es ist ratsamer, eine Operation für einen chirurgischen Studenten zu simulieren, als ihm ein Lebewesen als Versuchskaninchen anzubieten.

Der Einsatz dieser Technologie erfordert zwei grundlegende Elemente: eine Vorrangung(Hardware) für die Verbreitung von Virtual Reality und eine Entwicklungsumgebung wie unreal engin, unity usw., um unsere virtuelle Welt zu erzeugen. Für unsere Projekt, wurde als Hardware HTC VIVE und als Software Unreal engin ausgewählt.

### 3.3 Hardware: HTC VIVE

RfP

HTC Vive ist eine Vorrichtung (siehe Abbildung 5) bestehend aus einem Head-Mounted Display (siehe Abbildung 6), zwei Controllern (siehe Abbildung 7), Basisstation (siehe Abbildung 8), die alle von HTC in Kooperation mit Valve entwickelt wurden.



Abbildung 5: HTC VIVE mit VR-Brille, Controllern und Base[11].

Head-Mounted Display (siehe Abbildung 6) ist ein auf den Kopf getragenes visuelles Ausgabegerät, gewöhnlich nennen wir es VR-Brille und es wird mit einem PC über einen USB-Anschluss verwendet. Es bietet dem Benutzer ein Sichtfeld von  $110^\circ$  und einer Auflösung von  $2160 \times 1200$  Pixeln pro Auge um die visuelle Immersion durch vollständiges Eintauchen in die virtuelle Realität ermöglichen. Die Bildwiederholrate von 90 Hz ermöglicht eine vollkommen natürliche ruckelfreie Wahrnehmung von Szenen und Bewegungsabläufen. Damit der Benutzer das Relief wahrnehmen kann, verwendet der Helm stereoskopisches Sehen, das darin besteht, unsere beiden Augen zu benutzen, um das Relief wahrzunehmen, denn Sie wissen, dass wir mit beide Augen ein Bild unterschiedlich wahrnehmen, das Bild des rechten Auges wird im Vergleich zu dem des linken Auges verschoben und umgekehrt. Dieser Unterschied liegt darauf, dass unsere Augen 65 mm voneinander entfernt sind und daher nicht an der gleichen Stelle auf unserem Gesicht stehen, was zu einer Verschiebung des wahrgenommenen Bildes führt. Das Headset verfügt über viele Sensoren wie z.B. einen Gyroskop, (um den Benutzer im Weltraum zu lokalisieren), ein Beschleunigungsmesser und ein Laser-Positionsmesser-System.

Um die Position des Benutzers in einem Raum zu ermitteln, bietet das HTC VIVE zwei sogenannte "Steam-VR-Basisstationen" (siehe Abbildung 7) dem sogenannten Lighthouse-System (eine von Valve entwickelte Tracking-Technologie). Die Position wird dabei mit Hilfe von Infrarot-Lasern bestimmt. Jede Basisstation sendet Laserstrahlen aus, die von VR-Brille-Photosensoren und den Controllern erkannt werden.



Abbildung 6: VR-Brille[11].



Abbildung 7: VR-Basisstation[11].

Für eine gute Interaktion mit dem virtuellen Umgebung verfügt der Benutzer über zwei drahtlose Controller (jeweil ein pro Hand). Beide drahtlosen Steuerungen sind mit Gyroskop, Beschleunigungssensor und Laserpositionssensoren. Die Positions-messung der Controller erfolgt wie bei der VR-Brille über Photosensoren.



Abbildung 8: VR-Controller[11].

### 3.4 Software: Unreal engine 4

RfP

Um unsere virtuelle Umgebung zu erzeugen, werden wir als Entwicklungsumgebung Unreal Engine verwenden. Unreal Engine ist eine von Epic Games entwickelte Videospiel-Engine, die sich hauptsächlich an First-Person-Shooter-Spielen orientiert, obwohl sie immer vielfältiger wird. Eine Spiel-Engine ist ein spezielles Framework für Computerspiele, das den Spielverlauf steuert und für die visuelle Darstellung des Spielablaufes verantwortlich ist. In der Regel werden derartige Plattformen auch

als Entwicklungsumgebung genutzt und bringen dafür auch die nötigen Werkzeuge mit. Die Hauptkonkurrenten dieses Motors sind Unity, die von Crytek entwickelte CryENGINE sowie Amazon's Lumberyard (Gabel der CryENGINE). Unreal Engine würde von Anfang für das Projekt ELISE ausgewählt, deswegen müssten wir weiter verwenden. Diese Umgebung wird ständig verbessert, wir verwenden die vierte Version: Unreal Engine 4. Die Fähigkeiten der Unreal Engine 4 sind fantastisch. Es ist schon lange her, dass man dachte, die Entwicklung von Videospielen sei nur für eine Elitegruppe. Von nun an kann jeder Videospiele entwickeln, dank der Entwicklung von sehr zugänglichen Spiele-Engines wie der Unreal Engine 4. Die Unreal Engine 4 ist nicht nur extrem einfach zu bedienen und wird zunehmend von unabhängigen Studios verwendet, sondern ist auch eine der beliebtesten Spiele-Engines der größten Videospielstudios.

Es gibt viele Online-Tutorials, die detailliert erklären, wie Sie mit Unreal Engine beginnen können, so dass wir dieses Problem nicht untersuchen werden. Wir werden jedoch einige sehr wichtige Elemente (Blueprint) vorstellen, die wir während dieses Projekts verwenden werden.

Blueprint ist ein Programmiersystem, das als Visual Scripting System oder "Visual Scripting" auf Englisch bezeichnet wird. Eine der Besonderheiten von Unreal Engine ist der Wunsch, für alle zugänglich zu sein. Der Blueprint muss es denen, die nicht wissen, wie man programmiert (Zeilen Code schreibt), ermöglichen, mit der unreal Engine zu arbeiten. Mit Blueprint verbinden Sie einfach Elemente und lassen sie mit Visual Scripting miteinander interagieren. Dies ermöglicht es Ihnen, das Gleiche wie Programmierer zu tun, ohne eine Zeile Code zu schreiben(siehe Abbildung 9).

Blueprint ist daher ein extrem leistungsfähiges System, mit dem Sie Spielprototypen sehr schnell und einfach erstellen können. Blueprint wurde entwickelt, um von Anfängern, Künstlern oder sogar erfahrenen Programmierern verwendet zu werden. Man unterscheidet LevelBlueprint und BlueprintClasse.

## 3.5 Sensoren und biophysiolgische Signale zur Emotionserkennung

Verantwortlich: Kevin

### 3.5.1 Körpertemperatur-Sensor

RfP

Für das Projekt wurde zu Beginn ein Infrarot-Temperatursensor des Typs MLX90614

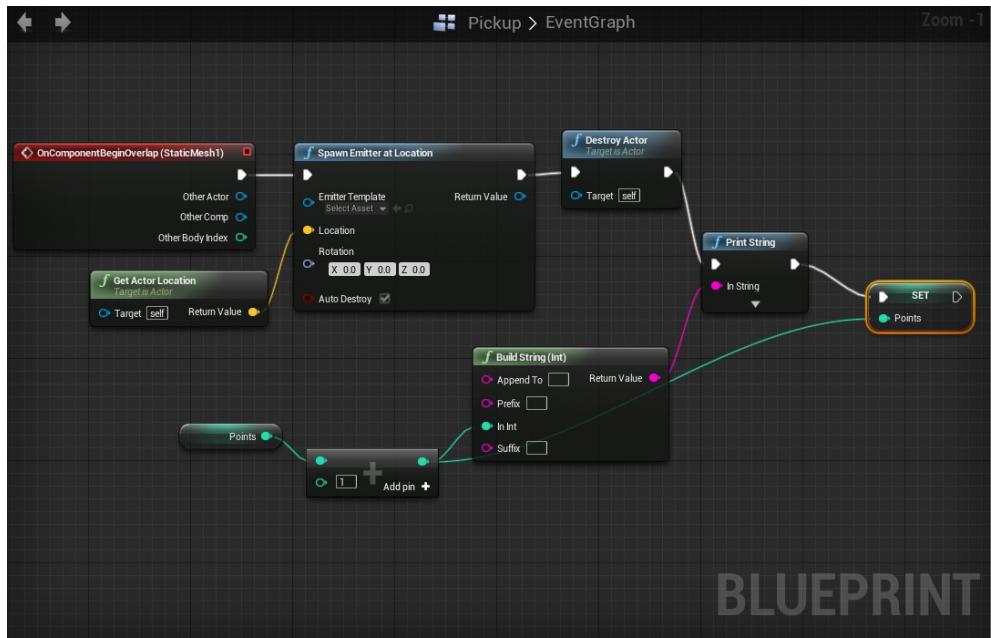


Abbildung 9: Beispiel eines Blueprintcodes.

von der Firma "Melexis-Microelectronic Integrated Systems" verwendet. Der MLX90614 ist ein sensibler digitaler 16-Bit Sensor, dessen Genauigkeit bei  $+/-0,5^{\circ}\text{C}$  liegt. Der Arbeitsbereich ist für Temperaturen des Sensors liegt zwischen  $-40^{\circ}\text{C}$  und  $+125^{\circ}\text{C}$ , und für die Kontaktlose Messung an Objekten, also der mögliche Temperatur Messbereich, zwischen  $-40^{\circ}\text{C}$  und  $+380^{\circ}\text{C}$ . Die Datenübertragung vom Sensor auf das Messboard erfolgt mittels eines I2C-Busses (siehe Abschnitt x.x zu I2C). Die Übertragungsrate wurde hierbei mit 100.000 kbit/s gewählt. Die Verwendung von I2C hat den Vorteil, dass auch andere digitale Sensoren ohne größeren zusätzlichen Verkabelungsaufwand an den Bus angeschlossen werden können. In der Abbildung X ist der Anschluss des Sensor exemplarisch aufgezeigt.

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

### 3.5.3 Messen der Sauerstoffsättigung (SpO2)

### 3.5.4 Galvanic Skin Response (GSR)

### 3.5.5 Elektroenzephalografie (EEG)

### 3.5.6 Elektrookulografie (EOG)

### 3.5.7 Analog/Digital-Wandler

## 3.6 Kommunikation

Verantwortlich: Kevin, Jonas

Zur übertragung der gewonnenen Sensordaten, wie auch zur Kommunikation der einzelnen Teilkomponenten untereinander, wurden verschiedene Technologien und Protokolle evaluiert.

Die Implementierung erfolgte schlussendlich mit UDP Broadcasts und einem rudimentären Kommunikationsprotokoll. Für die Übertragung war es wichtig, dass die Implementierung sowohl auf Seiten der Unreal Engine, als auch beim Mikrocontroller und bei der Kontrollanwendung einfach und mit geringem Arbeitsaufwand möglich ist. Zugleich sollte auf eine einfache Wartbarkeit und die Möglichkeit die Verbindung "live" zu debuggen geachtet werden.

Die ersten Tests erfolgten mit einer einfachen Ausgabe der Sensordaten über eine USB-Serial Verbindung. Da jedoch der Sensor abgesetzt und drahtlos Daten senden sollte, wurde dann beim ESP32 UDP Broadcasts als Kommunikationsweg gewählt. Neben der einfachen Implementierung auf allen Teilkomponenten ist hierbei auch die Möglichkeit einer 1:n Kommunikation gegeben. In Zukunft sollte darüber nachgedacht werden, Broadcasts durch Multicasts zu ersetzen, da dadurch in einem n:m Szenario eine einfachere Separierung der Datenströme der einzelnen Sensoren erfolgen kann.

Jeder Teilnehmer wird durch eine einzigartige ID identifiziert, die in jedem Paket mitgesendet wird.

#### (BILD EINFÜGEN UNTERSCHIEDLICHE DATENPAKETE)

Der Sensor sendet hierbei nicht für jeden Datenwert ein Paket, sondern aggregiert immer 75 Datenwerte in ein Paket. Dabei werden die einzelnen Kanäle in 4 Pakete aufgeteilt. Paket 1 enthält die Sensordaten für Temperatur, GSR und EEG1. Paket 2 für. Paket 3 für. Paket 4 zuletzt dann die Daten für IR-Raw. Diese Aggregation und Aufteilung wurde durch Instabilitäten in Sensor Netzwerkstack erforderlich. Trotzdem ist durch diese Lösung weiterhin eine Datenrate von 250 Samples/Sekunde möglich. Diese liegt weit über dem theoretischen Minimum, was für eine verlustfreie Rekonstruktion des Signals nötig wäre, so dass auch weiterhin die Kommunikation robust gegenüber Störeinflüssen ist.

### 3.7 Grundlagen der Mustererkennung

Verantwortlich: Artur  
- RfP

Mustererkennung (engl. "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 Mustererkennung-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 Etikette besteht). Aus diesem Grund wächst das Interesse an unüberwachter 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 Unüberwachter Techniken angewendet werden [12].

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, Frustation und Langeweile.

Um den Emotionsklassifizierer aufzubauen, wird ein standardmäßiger, überwachter Lernansatz namens Emotion Recognition Chain verwendet, der im folgendem Kapitel beschrieben wird.

## 3.8 Emotion Recognition Chain

Verantwortlich: Artur  
- RfP

Die Emotion Recognition Chain (ERC) besteht aus fünf Hauptschritten: Datenerfassung, Vorverarbeitung, Segmentierung, Merkmalsextraktion und Klassifizierung (vgl. Abb. 10). In den folgenden Unterkapiteln wird für jeden Schritt eine allgemeine Erklärung gegeben.

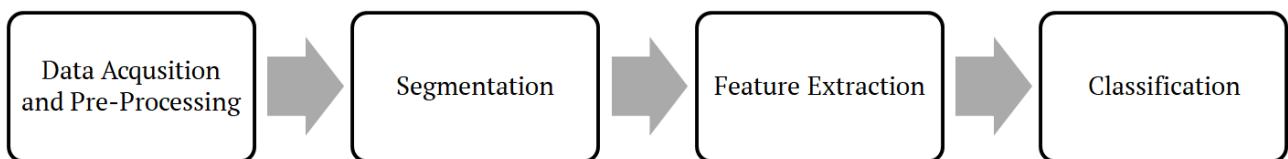


Abbildung 10: Emotion Recognition Chain: Zeitreihen-Datensätze werden von tragbaren Sensoren aufgenommen (Datenerfassung) und vorverarbeitet (Vorverarbeitung). Die Daten werden dann in Segmente unterteilt (Segmentierung), aus denen Merkmale extrahiert werden (Merkmalsextraktion). Mit den gewonnenen Merkmalen wird schließlich ein Klassifikator trainiert und anschließend dessen Ergebnisse bewertet (Klassifikation).

### 3.8.1 Datenerfassung

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.

### 3.8.2 Vorverarbeitung

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

### **3.8.3 Segmentation**

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.

### **3.8.4 Merkmalsextraktion**

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.

### **3.8.5 Klassifikation**

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.

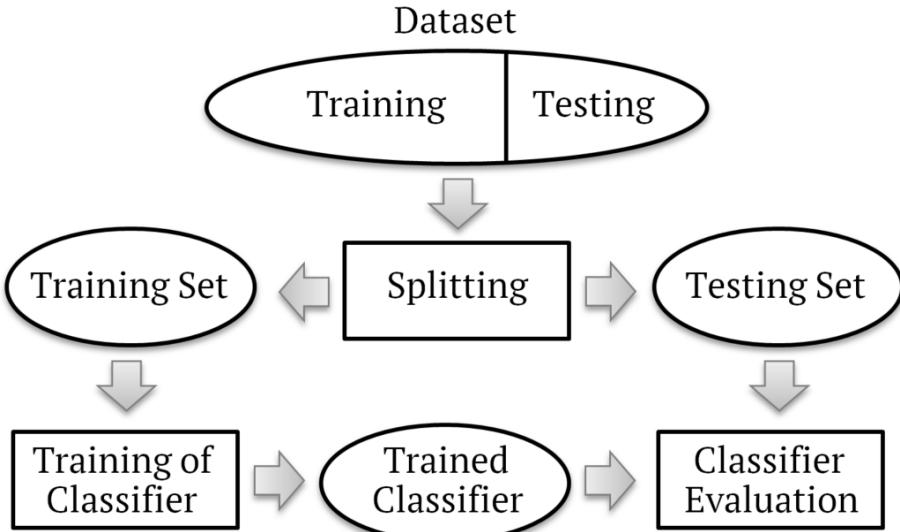


Abbildung 11: Aufteilen eines Datensets in ein Trainings- und ein Testset. Das Trainingsset dient zum Erstellen und Trainieren eines Klassifikators. Die Performance wird mithilfe des Testsets ermittelt und bewertet.

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 Klassenettiket 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 trennenden Hyperebene sind. Cortes und Vapnik zeigten in [13], dass die Gleichung der optimalen Hyperebene nur von diesen spezifischen Vektoren abhängig ist. Abbildung 3.8.5 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.

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

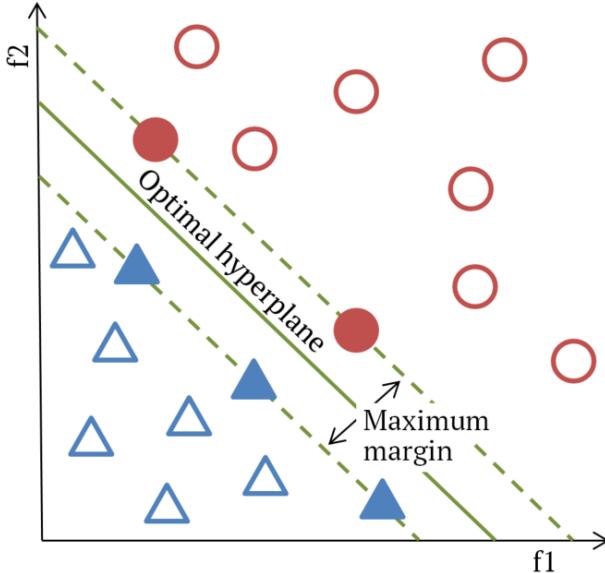


Abbildung 12: Beispiel eines SVM-Klassifikators im zweidimensionalem Merkmalsraum: Die Datenpunkte beider Klassen (dargestellt durch nicht-ausgefüllte blaue Dreiecke und rote Kreise) sind durch eine Hyperebene getrennt, welche die Margin maximiert. Stützvektoren werden als ausgefüllte Dreiecke und Kreise dargestellt.

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 [13] 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 erübriggt 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:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|) \quad (1)$$

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

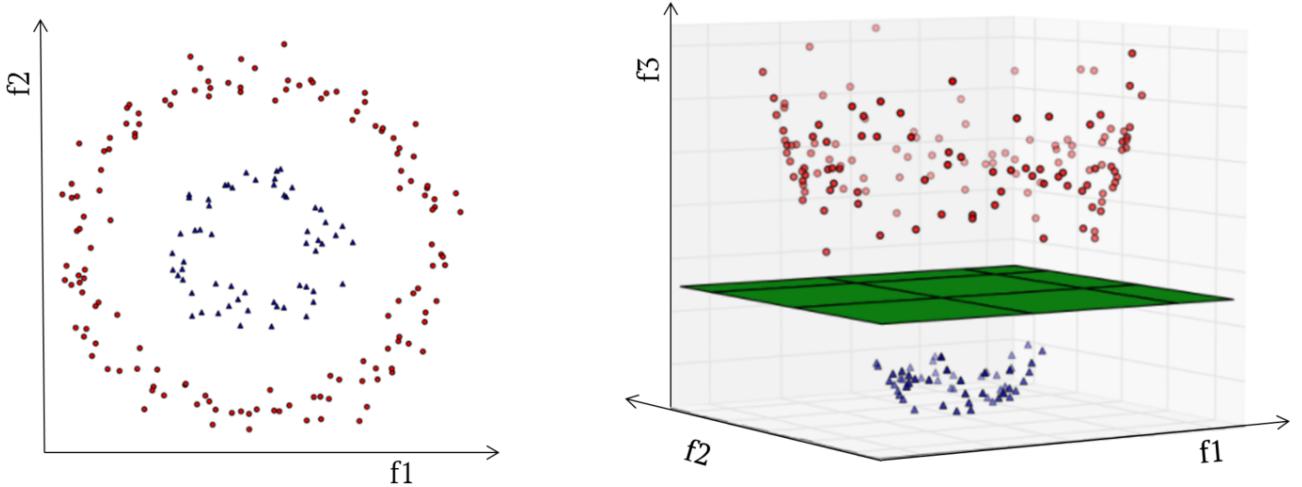


Abbildung 13: Beispiel für die Verwendung des Kernel-Tricks: Nicht-linear trennbar Daten in einem 2D-Merkalsraum (links) können linear trennbar gemacht werden, wenn sie auf 3D projiziert werden (rechts) [14].

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 [13] 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.

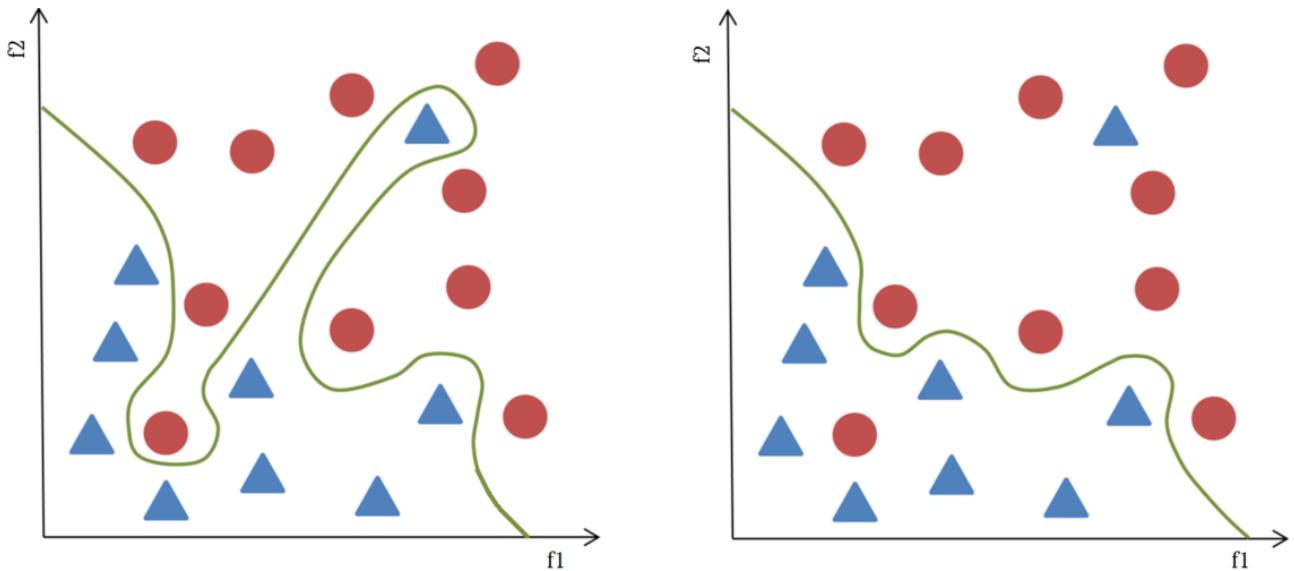


Abbildung 14: Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalsraum: Der hard-margin SVM trennt die beiden Klassen perfekt im Gegensatz zum Soft-Margin SVM, der einige Fehlklassifikationen zulässt.

## 4 State-of-the-Art Analyse

Verantwortlich: Jonas

## Part I

# 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 Software

Verantwortlich: Jonas, Kevin

## 7 Emotionsinduktion

Verantwortlich: Minas  
- RfP

Das Kapitel beinhaltet einen Ablauf, die verwendeten Fragebögen und die genutzten Szenarien (Glück, Langeweile, Frustration) der Emotionsinduktion für den ersten Prototypen. Kapitel Ablauf beschreibt in welcher Reihenfolge die Szenarien und die Fragebögen aufgerufen wurden. In den Kapiteln Fragebogen, Glücks-, Langeweile- und Frustration-Szenario wird, die Entwicklung und Implementierung erläutert. Die Emotionsinduktion des ersten Prototypen wurde nicht in einer VR-Umgebung entwickelt. Zudem muss angemerkt werden, dass diese Emotionsinduktion für eine Zwischenstudie diente. Dadurch wurde der Fokus nicht auf die Implementierung gelegt, welche verschiedene Formate besaß z.B. HTML oder PowerPoint, sondern auf den Inhalt, welches der Proband zu sehen oder zu hören bekam. Die Emotionsinduktion wurde sowohl in Englisch als auch in Deutsch angeboten.

### 7.1 Ablauf

Verantwortlich: Minas  
- RfP

Der Ablauf der Emotionsinduktion lief wie folgt ab:

1. Glück (PowerPoint) → Kapitel 7.3.1
2. Langeweile (Video) → Kapitel 7.3.2
3. Frustration (Flashgame) → Kapitel 7.3.3
4. Langeweile (Spiel im Browser) → Kapitel 7.3.2

Nach jedem Szenario folgte ein sich automatisch öffnender Fragebogen zum Szenario. Der Wechsel zwischen den Szenarien musste manuell statt finden.

### 7.2 Fragebogen

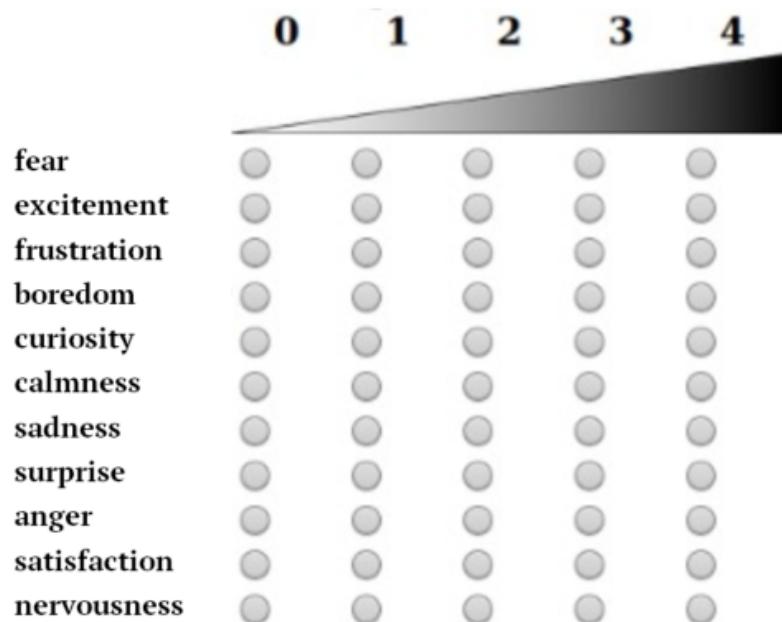
Verantwortlich: Boris  
- RfP

Ein Fragebogen kann als Formular definiert werden, das einen Satz von Fragen enthält, die für einen bestimmten Zweck definiert wurden[15]. Hier dient der Fragebogen als Befragungsinstrument für die Realisierung unseres Projekts. Hierfür wurden für die Ausarbeitung die Hauptkomponenten definiert, die Auskunft über das Problem der Studie und über die Versuchsperson geben.

Für das erste Experiment dieser Studie sieht das Fragebogenmodell folgendes aus: eine Liste von elf Emotionen (Angst, Aufregung, Frustration, Langeweile, Neugier, Ruhe, Traurigkeit, Überraschung, Wut, Befriedigung und Nervosität) mit je vier nummerierten Check-Boxen, die nach der Intensität des Gefühls der betreffenden Emotion angekreuzt werden sollten. Das erste Feld entspricht der niedrigsten Intensität und das vierte Feld der höchsten Intensität. Man hat auch die Möglichkeit, keine Check-Box auf eine Emotion anzukreuzen, wenn man seiner Meinung nach die Emotion gar nicht gespürt hat. Das würde dem Intensität Null entsprechen. Zudem soll man auch für jedes Viertel der Zeit eines Szenario die dominante Emotion wählen.

### Experiment series 1 – Questionnaire

Below is a list of emotions. Please indicate how strongly you felt the emotions during the experiment. Please chose a number from a scale of “0” (no emotion) to “4” (strong emotion).



Please select in the following lists for the 4 time frames which of the listed emotions occurred most strongly with you. If no emotion was dominant, please select “[none]”.

Time frames: Begin – 25%

Please select

25% - 50%

Please select

50% - 75%

Please select

75% - 100%

Please select

Abbildung 15: Bild des verwendeten Fragebogens.

## 7.3 Szenarien

Verantwortlich: Meryem  
- RfP

Im Folgenden Kapitel werden drei verschiedene Szenarien dargeboten. Die Emotionen Glück, Langweile und Frustration werden hierfür zunächst erläutert. Für jede dieser Emotionen werden Szenarien vorgestellt, bei welchen diese bei den Probanden ausgelöst bzw angeregt werden sollen.

### 7.3.1 Glück

Verantwortlich: Minas  
- RfP

Das Wort “Glück” wird im deutschsprachigen Gebrauch mit zwei unterschiedlichen Grundbedeutungen verwendet. In anderen Sprachen wird dieser Begriff auseinandergehalten z.B. im englischen als “luck” und “happiness” und im französischen als “la bonne chance” und “le bonheur”. Im deutschen Sprachgebrauch muss zwischen den Glückszufall und die Glücksgabe, welches nicht erzwungen werden kann und zwischen das Glücklichsein, die Glückserfahrung und das Glückserlebnis unterschieden werden[16].

In diesem Szenario war es die Absicht anhand Fotos mit Texten und Musik im Hintergrund eine Glückserfahrung oder auch ein Glückserlebnis bei den Probanden auszulösen. Abbildung 17 zeigt die hierfür verwendeten Bilder und deren Texte.

## Glücks-Szenario



Abbildung 16: Vignetten für das Glücks-Szenario.

Die Bilder wurden in einer PowerPoint-Präsentation abgespielt. Diese wechselten alle 30 Sekunden, während die Audiodatei im Hintergrund ablief. Die Idee stammt von der wissenschaftlichen Publikation “Mood inductions for four specific moods: A procedure employing guided imagery vignettes with music,” von J. D. Mayer, J. P. Allen, und K. Beauregard. In dieser Publikation wurde versucht anhand Vignetten und Hintergrundmusik Glücksemotionen auszulösen. Die verwendete Audiodatei aus der Publikation, welches aus dem Ballettstück “Coppélia ou La Fille aux yeux d’email” stammte, wurde in diesem Szenario übernommen. Die aus der Publikation verwendeten Vignetten wurden insofern modifiziert, das einzelne durch eigene Vignetten ersetzt wurden.

### 7.3.2 Langeweile

Verantwortlich: Boris  
- RfP

Langeweile wird im Allgemeinen als eine Emotion wahrgenommen, die man erlebt, wenn man einen Zustand durchläuft, an dem man kein Interesse hat[17]. Dieses Gefühl charakterisiert Inaktivität, Unproduktivität und kann manchmal zu Melancholie oder Traurigkeit führen. In dieser Studie wird das Erwachen der Langeweile durch eine grüne rotierende Scheibe mit einem schwarzen Strahl verursacht.

Für das erste Prototyp wird zwei Szenarios benutzt um die Langeweile bei Probanden zu erwecken. Als ersten wurde eine 5 minütigen Video über die "Latente Steuer im Jahres Abschluss" von der Tax Universität<sup>1</sup> benutzt um diese Emotion auszulösen. Dabei ging es um eine kurze Einführungsvideo über Auswirkung von Latentesteuern in der Jahresbilanz. Als zweites Szenario würde das "Peg-Turning" Spiel benutzt. Sein Prinzip besteht darin, alle 3 Sekunden eine grüne Scheibe durch ein Mausklick zu drehen.

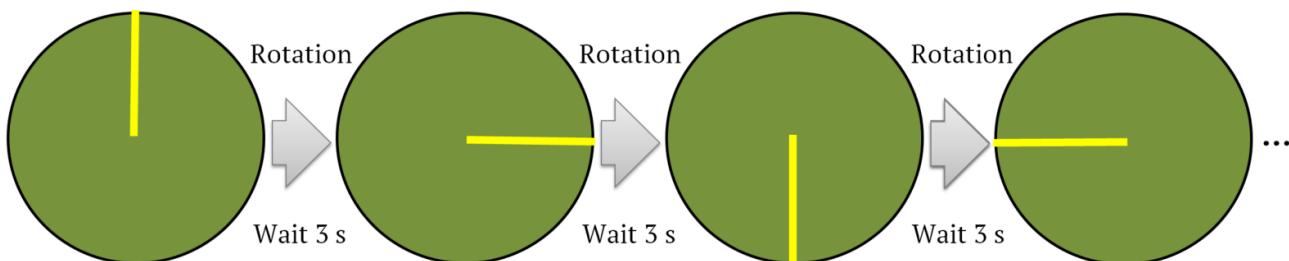


Abbildung 17: Bild des Langeweile-Szenarios.

### 7.3.3 Frustration

Verantwortlich: Meryem  
- RfP

Die Frustration stellt einen negativen Zustand des Menschen dar, welcher mehrere Indikatoren haben kann. Dieser Zustand kann sowohl eine Gefühlslage als auch eine Folge vorhergehender Emotionen sein. Im Allgemeinen setzt die Frustration ein, wenn Misserfolgserlebnisse sowie Versagungs- und Enttäuschungserlebnisse vorliegen. Ursachen dafür können in der Person selber und/ oder in Umwelteinflüssen liegen. Frustration wird ausgelöst, wenn die Person eine wirkliche Benachteiligung erleidet oder sie als diese wahrnimmt. Eine empfundene Ungerechtigkeit resultierend aus Erwartungen der Umwelt an die Person sowie Erwartungen, die an sich selbst gerichtet sind, können den Menschen frustrieren.

<sup>1</sup>Link zu dem Video: <https://www.youtube.com/watch?v=nhcG8zC7G2o>

Für das Frustrationsexperiment der Studie soll auf das Auslösen der Misserfolgerlebnisse sowie eine empfundene Ungerechtigkeit zurückgegriffen werden. Der Proband wird aufgefordert das Spiel "Frustabit" zu spielen. Ziel des Spiels ist es, den Cursor, mit Hilfe der Computermause, durch das Labyrinth zu navigieren, ohne den Rand bzw die Wand zu berühren. Hinzu kommt die Aufgabenstellung, acht Level innerhalb von fünf Minuten zu lösen, um eine Prämie von 10 EUR zu gewinnen. Zur Manipulation hat der Versuchsleiter eine kabellose Maus, von der der Proband nichts weiß. Mit Hilfe dieser Maus verhindert der Versuchsleiter, dass der Proband die Aufgabenstellung erfüllen und die Prämie gewinnen kann, indem der Versuchsleiter den Cursor nach belieben an die Wand des Labyrinths steuert.

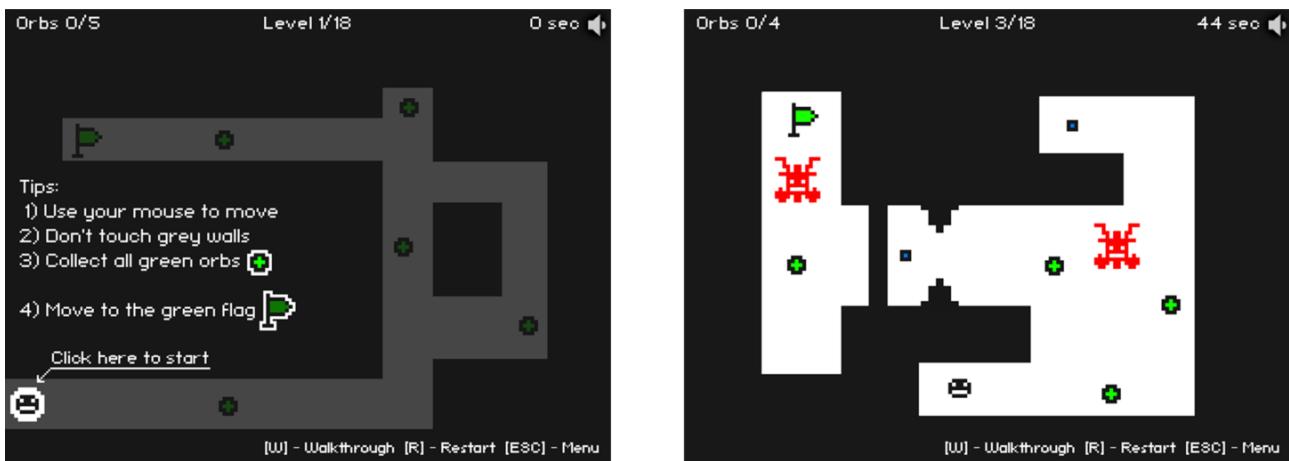


Abbildung 18: Bild des Frustabit Spieles für das Frustration-Szenario.

## 8 Messreihe

Verantwortlich: Kevin, Artur

# 9 Mustererkennung

Verantwortlich: Artur  
- RfP

In diesem Kapitel werden alle Schritte entlang der ERC (vgl. Kapitel 3.8) konkretisiert und es wird detailreich beschrieben, wie wir vorgegangen sind.

## 9.1 Datenerfassung

In Kapitel 3.8.1 wurde das verwendete Datenset bereits detailliert beschrieben, sodass hier darauf verzichtet wird.

## 9.2 Vorverarbeitung

Wie bereits in Kapitel 3.8.2 beschrieben, ist das Ziel der Vorverarbeitung die "Verbesserung" der Daten für die nachfolgenden Schritte der ERC. Im Rahmen des ELISE Projektes wurden Normalisierungstechniken auf dem gesamten Datensatz angewendet. Wir haben insbesondere die Standardnormalisierung verwendet, welche den Mittelwert der Daten auf Null setzt und die Einheitsvarianz ergibt [18]. Die Formel für die Standardnormierung lautet:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (2)$$

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

## 9.3 Segmentation

Wie bereits in Kapitel 3.8.3 beschrieben, ist das Ziel der Segmentation Teile von Daten zu identifizieren, welche wichtige Informationen über die zu erkennenden Emotionen enthalten. 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 entsprechendem Fenster enthaltenen Zeitstempel basiert. Es wird davon ausgegangen, dass jedes Zeitfenster nur von einer Emotion belegt ist.

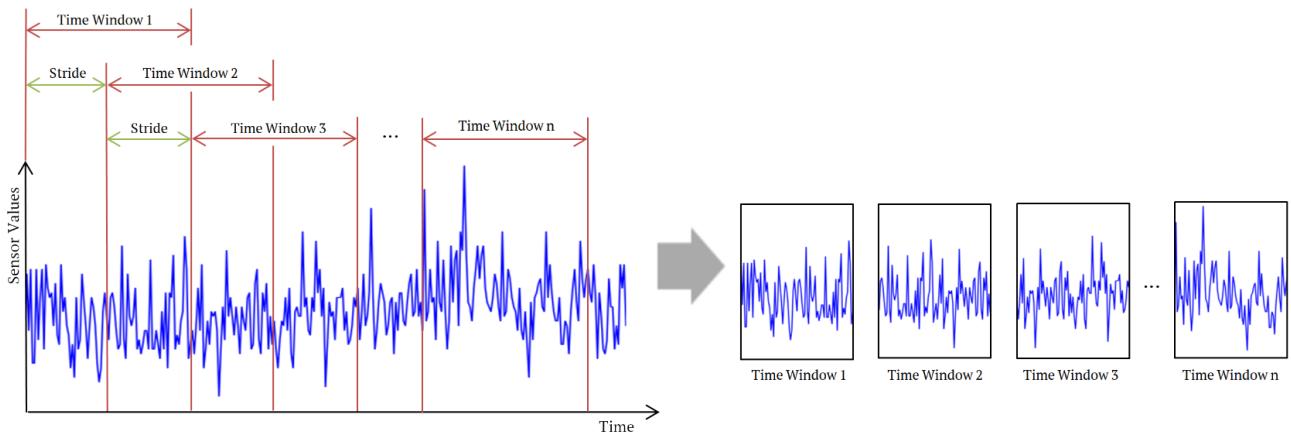


Abbildung 19: Schiebefenster-Segmentierung: Die Daten werden durch ein Zeitfenster fester Größe in kleinere Segmente aufgeteilt. Das Fenster wird mit einem festen Gleitschritt geschoben, um den aufeinanderfolgend Daten-Zeitfenster zu erhalten.

## 9.4 Merkmalsextraktion

RfP

Wie bereits in Kapitel 3.8.4 beschrieben, ist das Ziel der Merkmalsextraktion Charakteristiken und Merkmale in den Daten zu finden, die für das Klassifizierungsproblem von möglichst hoher Relevanz sind. Im Rahmen des ELISE Projektes haben wir verschiedene Vorgehensweisen angewendet. Im den folgenden Unterkapiteln werden diese vorgestellt.

### 9.4.1 Handgefertigte Merkmale

Der handgefertigten Merkmal Ansatz (enlg. "hand-crafted features approach") besteht in der Berechnung relativ einfacher Merkmale von denen vermutet wird, dass sie für das Klassifizierungsproblem der Eingangssignale relevant sein können. Diese Vorgehensweise hat den Vorteil des einfachen Aufbaus als auch der relativ geringen benötigten Rechenleistung, wobei potentiell gute Klassifizierungsergebnisse erwarten werden.

Obwohl frühere Forschungsarbeiten schon handgefertigte Merkmale zur Emotionserkennung unter mithilfe physiologischer Signale getestet haben (vgl. [19]), wurde dieser Ansatz noch nie für die Erkennung dieser spezifischen Emotionen unter Verwendung dieser Kombination von Sensoren getestet. Zusätzlich haben wir zuerst handgefertigte Merkmale getestet, um ein Basisergebnis zu liefern, mit der die Ergebnissen der anderen Ansätze vergleichen werden können. Handgefertigte Merkmale sind in der Regel entweder einfache statistische Werte, Fourier-basierte oder selbstentwickelte Merkmale sein, die aufgrund von Vorkenntnissen der Daten verwendet werden. Diese Arbeit wurden statistische, Fourier-basierte und selbsten-twikelte Merkmale getestet.

## Statistische Merkmale

Die Tabelle 3 fasst die elf verschiedenen und in der Studie verwendeten statistischen Merkmale zusammen [20]. Wir bezeichnen  $\mathbf{x} = (x_1, x_2, \dots, x_T)$  als Vektor, der die in einem Datenzeitfenster der Länge  $T$  enthaltenen Sensorwerte für einen Sensorkanal darstellt.

Merkmalname	Definition
Mittelwert	$mean(\mathbf{x}) = \frac{1}{T} \sum_{k=1}^T (x_k)$
Standard-Abweichung	$\sigma(\mathbf{x}) = \sqrt{\frac{1}{T} \sum_{k=1}^T (x_k - \mu)^2}$
Maximum	$max(\mathbf{x}) = \max(x_1, x_2, \dots, x_T)$
Minimum	$min(\mathbf{x}) = \min(x_1, x_2, \dots, x_T)$
Amplitude	$A(\mathbf{x}) = max(\mathbf{x}) - min(\mathbf{x})$
25/50/75% Perzentil	Wert einer Menge, unter dem 25/50/75% der Werte aus der Menge fallen.
Interquartiler Bereich	Differenz zwischen dem 75. und 25. Perzentil.
Schräge	$\gamma_1(\mathbf{x}) = E \left[ \left( \frac{X-\mu}{\sigma} \right)^3 \right] = \frac{\mu_3}{\sigma^3} = \frac{E[(X-\mu)^3]}{(E[(X-\mu)^2])^{3/2}} = \frac{\kappa_3}{\kappa_2^{3/2}}$
Kurtosis	$Kurt[\mathbf{x}] = E \left[ \left( \frac{X-\mu}{\sigma} \right)^4 \right] = \frac{\mu_4}{\sigma^4} = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^2}$

Tabelle 3: Statistische Merkmale, die im Rahmen des ELISE-Projektes verwendet wurden.

## Fourier-basierte Merkmale

Die in diesem Kapitel präsentierten Ergebnisse wurden in einer Bachelorarbeit [21] ermittelt, die auch ihm Rahmen des ELISE-Projektes geschrieben wurde.

Die Transformation von Zeitsignalen im Frequenzbereich bei Studien zu einem gängigen Werkzeug für die Analyse von Zeitreihendaten geworden. Es basiert auf einem mathematischen Prozess namens Fouriertransform, wo ein Signal in einer unendlichen gewichteten Summe von Sinus- und Cosinusschwingungen unterschiedlicher Frequenz zu zerlegen. Bei einer kontinuierlichen Funktion  $f : t \rightarrow f(t)$  ist der Fouriertransformation  $F(w)$  eines Signals  $f$ :

$$F(w) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i w t} dt \quad (3)$$

In den meisten realen Anwendungen sind die Signale jedoch diskontinuierlich. Eine alternative Fourier-Transformation, die sogenannte diskrete Fourier-Transformation (engl. "discrete fourier transform"), kann dann stattdessen angewendet werden. Ein diskretes Signal  $x = (x_0, x_1, x_2, \dots, x_{N-1})$  von  $N$  Punkten wird durch die folgende Formel gegeben:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \quad (4)$$

wobei die Koeffizienten  $X_k \in \mathbb{C}$  die Fourier-Koeffizienten des Originalsignals sind. Die Fourier-Koeffizienten werden hauptsächlich verwendet, um das Leistungsspektrum (engl. "power spectrum")  $p(x)$  des Signals  $x$  zu berechnen, das Auskunft über den Beitrag jeder Frequenzkomponente zum Signal gibt. Die Komponenten  $p_k(x)$  des Leistungsspektrums für  $k \in \{0, 1, \dots, N - 1\}$  werden berechnet durch:

$$p_k(x) = \Re(X_k^2) + \Im(X_k^2) \quad (5)$$

wobei  $p_k(x)$  mit der Energie des Signals  $x$  verglichen werden kann, das der  $k$ -ten Frequenz zugeordnet ist.

Um frequenzbezogene Merkmale zu extrahieren, wird das Leistungsspektrum der Signale in vier Frequenzbänder gleicher Länge unterteilt. Der Mittelwert, die Standard-Abweichung, das Maximum und das Minimum wurde dann auf jedem Frequenzband für jeden Sensorkanal extrahiert, was zu einem Prozess der Merkmalsextraktion aus normalisierten Datenrahmen führte, wie in Abbildung 20 dargestellt.

Zusätzlich wurde für jeden Sensorkanal des gesamten Spektrums die Standard-Abweichung extrahiert. Dies führt zu insgesamt  $9 * (1 + 4 * 4) = 153$  Merkmalen im Frequenzbereich pro Zeitfenster.

## Selbstentwickelte Merkmale

Es wurden zwei eigene Merkmale definiert [20]. Nulldurchgang (engl. "zero

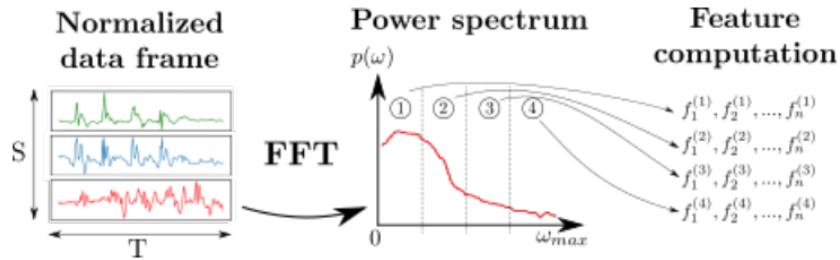


Abbildung 20: Merkmalsextraktion aus frequenzbezogener Domain.

crossing") und Anzahl der Spitzen (engl. "number of peaks"). Im Folgendem werden diese beiden Merkmale detailliert beschrieben.

Das Nulldurchgang-Merkmal zählt die Häufigkeit, mit der das Signal eines Sensorkanals in einem Zeitfenster die Nulllinie überschreitet. Alle Sensorsignale wurden durch Normierung verarbeitet (vgl. Kapitel 9.2) und damit wurden alle Mittelwerte auf Null zentriert. Um zu vermeiden, dass Rauschen entlang der Nulllinie in dem Merkmal gezählt wird, wird nur ein Nulldurchgang in einer bestimmten Zeitspanne gezählt.

Das Spitzenzähler-Merkmal bestimmt die Anzahl von lokalen Hochpunkten im Zeitsignal. Alle lokalen Maximen sind durch einen Onset (Startpunkt), eine Spalte und einen Offset (Endpunkt) gekennzeichnet (vgl. Abbildung 21). Jedes Vorkommen einer Onset/Offset-Paarung wird hierbei als Spalte gezählt. Onsets, Spitzen und Offsets werden durch die folgenden Operationen identifiziert (vgl. [22]):

- Ein Onset wird bestimmt, wenn der Wert des Signals an diesem Punkt nicht negativ ist und die Differenz zwischen ihm und dem nächsten größer als ein vordefinierter Schwellenwert (engl. "threshold") ist.
- Ein Offset wird bestimmt, wenn der Wert des Signals kleiner als der Wert des zuletzt gesetzten Onsets ist.
- Das lokale Maximum zwischen einem Onset und Offset wird als Spalte bezeichnet.

Jedes handgefertigte Merkmal wird auf einem Zeitfenster von Daten für jeden Sensorkanal unabhängig voneinander angewendet. Jedes Zeitfenster ist daher 117 Merkmalen zugeordnet (9 Sensorkanäle multipliziert mit 13 Merkmalen).

#### 9.4.2 Codebook Approach

Die bisherige Methodik mit handgefertigten Merkmalen ist klassisch für den überwachten Lernansatz. Es existieren aber auch einige Nachteile. Das Haupt-

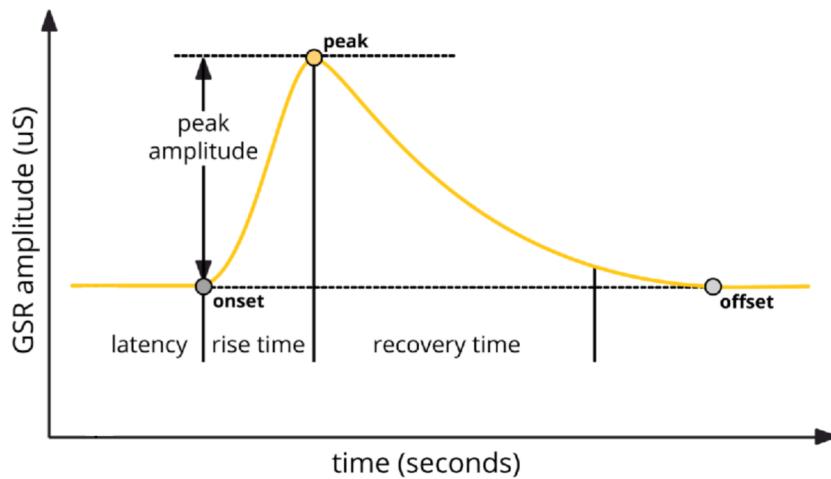


Abbildung 21: Spitzenzähler-Merkmal: Onset (Startpunkt), Spitze und Offset (Endpunkt). Jedes Paar von Onset/Offset erhöht die Anzahl der Spitzen um eins.

problem besteht darin, dass nicht sichergestellt werden kann, dass die gewählten Merkmale die besten Klassifizierungsergebnisse erzielen. Damit besteht immer die Gefahr, dass möglicherweise andere Merkmale bessere Ergebnisse liefern würden, diese handgefertigten Merkmale aber die nicht gefunden wurden. Dieses Risiko besteht insbesondere bei der physiologischen Signalverarbeitung zur Emotionserkennung, wo die Struktur der Daten noch recht unbekannt und allgemein komplex ist. Eine weitere Schwierigkeit besteht darin, relevante selbstentwickelte Features ohne Expertenwissen über die Daten zu finden. Darüber hinaus wurden noch keine gut funktionierenden State-of-the-Art handgefertigten Merkmale identifiziert. Aus diesen Gründen ist es interessant halbautomatische und unüberwachter Ansätze der Merkmalsextraktion zu verwenden und zu testen.

K. Shirahama et al. [23] schlugen eine unüberwachte Merkmalsextraktionsmethode namens Codebook Approach (CA) vor, um Merkmale aus 1D-Zeitreihensignalen zu erzeugen. Der CA hat den Vorteil, dass formbasierte Merkmale gefunden werden können, die für das Problem der Emotionserkennung relevant sind, aber weder offensichtlich noch leicht als Mensch zu interpretieren sind. Der CA besteht aus drei Schritten, die in den folgenden Abschnitten erläutert werden: Codebuchkonstruktion (engl. "codebook construction"), Codewortzuordnung (engl. "codeword assignment") und der anschließenden Klassifizierung.

## Codebuchkonstruktion

Ziel dieses Schrittes ist es, Teilsequenzen (sogenannte "Codewörter") zu bestimmen, die für die 1D-Eingangssensorik charakteristisch sind. Dies wird erreicht, indem Zeitfenster aus dem ursprünglichen Datensatz für jeden Sensorkanal unabhängig voneinander nach dem im Kapitel 9.3 definierten Segmentierungsansatz extrahiert werden. Aus jedem so erhaltenen Zeitfenster der Größe  $T$  werden

kleinere Segmente der Größe  $\alpha$  unterteilt. Ein Clustering-Algorithmus wird dann auf die Menge der Segmente  $\alpha$  angewendet, um Clusterzentren zu finden. Nach der Konvergenz werden die Clusterzentren als Codewörter betrachtet und zum Aufbau einer Sammlung von Codewörtern mit dem Namen "Codebuch" verwendet, wie in Abbildung 22 aus [23] dargestellt. Die Anzahl der Codewörter (d.h. die Größe des Codebuchs oder die Anzahl der Cluster) ist ein Hyperparameter des Verfahrens. Im Rahmen dieser Arbeit wurde ein k-means Clustering-Algorithmus verwendet, um die Codewörter auf den ELISE-Daten zu erhalten.

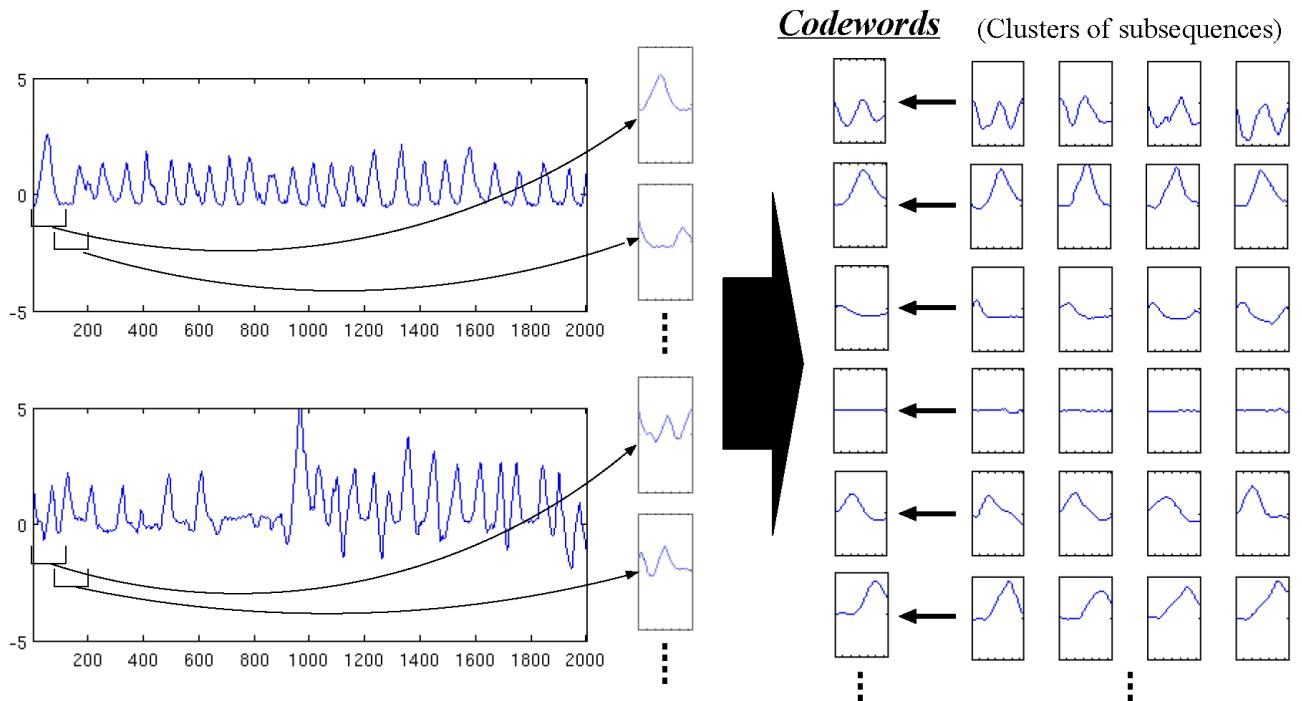


Abbildung 22: Codebuchkonstruktion: Zeitfenster von Daten werden zunächst extrahiert. Ein Clustering-Algorithmus wird dann auf das alle Zeitfenster angewendet, um Clusterzentren (und damit Codewörter) zu finden, die zum Aufbau des Codebuchs verwendet werden.

## Codewortzuordnung

Nach der Konstruktion der Codewörter wird für jedes Zeitfenster  $T$  ein histogrammbasierter Merkmalsvektor erstellt (vgl. Abbildung 23, entnommen aus [23]). Der zu klassifizierende Datensatz wird zunächst in Zeitfenster der Größe  $T$  segmentiert, aus denen Segmente der Größe  $\alpha$  nach dem gleichen Verfahren wie beim Aufbau des Codebuchs extrahiert werden. Jedes Segment  $\alpha$  wird dann mit den Codewörtern verglichen, so dass das "ähnlichste" Codewort gefunden werden kann. Ein K-Bin-Histogramm (mit  $K$  Anzahl der Codewörter) mit Informationen über die Anzahl der Male enthält, die jedes Codewort als am ähnlichsten zu den Segmenten  $\alpha$  im Zeitfenster  $T$  betrachtet wurde, wird dann erstellt und als Merkmalsvektor verwendet, um das zu klassifizierende Zeitfenster  $T$  darzustellen. Das Maß für die Ähnlichkeit von Codewörtern und Datensegmenten  $\alpha$  basiert auf der euklidischen

Entfernung.

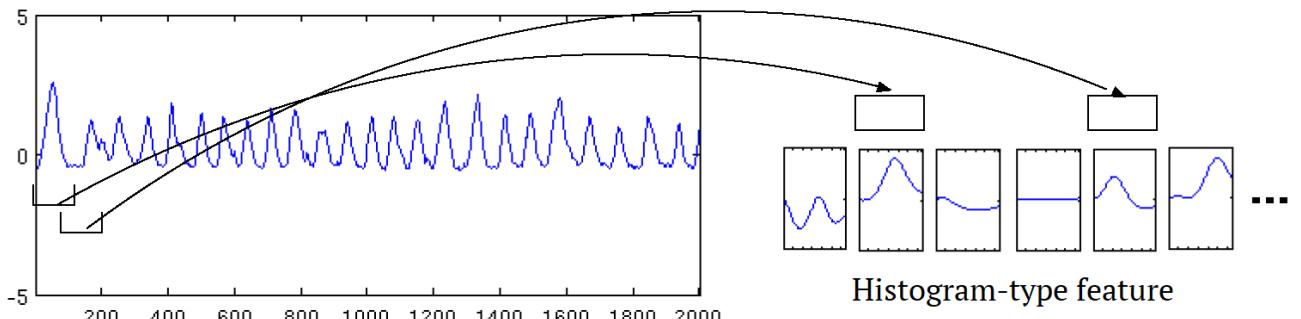


Abbildung 23: Codewortzuweisung: Jedes Datensegment wird mit den Codewörtern verglichen und ein Histogramm mit Informationen darüber, wie oft jedes Codewort als "am ähnlichsten" betrachtet wurde, wird erstellt.

Der zuvor beschriebene Ansatz wird als "Hard-Zuordnung" (engl. "hard assignment") bezeichnet, da Datensegmente einem einzigen Codewort zuzuordnen werden. Ein Nachteil dieser Vorgehensweise ist die mangelnde Flexibilität im Umgang mit potenzieller Unsicherheit bei der Codewortzuweisung, die z.B. auftreten kann, wenn ein Zeitfenster von Daten zwei oder mehr Codewörtern sehr ähnlich ist, da es nur einem zugeordnet werden kann. Ein alternativer Ansatz dieses Problems zu umgehen ist die so genannte "Soft-Zuordnung" (engl. "soft assignment"). Hierbei werden auch alle Codewörter des Codebuchs Datensegmenten zuzuordnen, wobei die Ähnlichkeit jeweils als ein Bin im Histogramms dargestellt wird. Sehr ähnlich entspricht hierbei einem hohen Wert und nicht ähnlich einem kleinen Wert (anstatt 0 oder 1 wie in der Variante der Hard-Zuordnung). Eine Kerneldichtefunktion (vgl. [24]) wird verwendet, um die Histogramm-Bins zu berechnen. Es wurde die folgende Funktion verwendet:

$$f(\alpha, c_k, p) = \frac{1}{\gamma} \frac{g(\alpha, c_k, p)}{\sum_{i=1}^K g(\alpha, c_i, p)} \quad (6)$$

wobei  $f(\alpha, c_k, p)$  die Ähnlichkeit des Segments  $\alpha$  bezeichnet, das sich auf das Codewort  $c_k$  bezieht.  $p$  ist ein Glättungsparameter (ein großes  $p$  bewirkt eine starke Glättung),  $\gamma$  ist die Anzahl der Segmente im Zeitfenster  $T$  und:

$$g(\alpha, c_k, p) = \frac{1}{\sqrt{2\pi p^2}} \exp\left(-\frac{d(\alpha, c_k)^2}{2p^2}\right). \quad (7)$$

Um numerischen Unterlauf zu vermeiden, wird die Gleichung (7) zunächst mit dem Log-Sum-Exp-Trick (vgl. [25]) berechnet.

## Fusion mehrerer Sensoren

Oftmals werden mehrere Sensoren verwendet, die gleichzeitig mehrere verschiedene Signale derselben Emotion erzeugen. Die Fusion dieser Signale ist wichtig, da sie die Genauigkeit der Emotionserkennung verbessern kann. Es gibt zwei verschiedene Ansätze (vgl. [26]): die frühe Fusion (engl. "early fusion") und die späte Fusion (engl. "late fusion"):

- Frühe Fusion: In der Dimension  $K \times S$  wird nur ein Klassifizierer benötigt, wobei  $K$  die Anzahl der Codewörter und  $S$  die Anzahl der Sensorkanäle ist. Der Klassifikator wird anhand der Verkettung von Codebuchmerkmalen trainiert und ausgewertet, die auf jedem Sensorkanal unabhängig voneinander berechnet wurden.
- Späte Fusion: Erfordert mindestens  $S$ -Klassifikatoren (einen für jeden Sensorkanal). Ein Klassifizierer wird unabhängig für jeden Sensorkanal unter Verwendung der für den betrachteten Sensor erhaltenen Codebuchmerkmale trainiert. Die Vorhersagen der  $S$ -Klassifikatoren werden dann fusioniert, um die Klassenbezeichnung des zu klassifizierenden Zeitfensters zu schätzen (z.B. mit einem zusätzlichen Klassifizierer).

In dem ELISE Projekt verwenden wir den späten Fusionsansatz, weil er rechnerisch günstiger ist und von K. Shirahama (vgl. [23]) empfohlen wird.

### 9.4.3 Deep Neural Networks (DNN)

Im Rahmen des ELISE Projektes, wurden verschiedene Deep Neural Networks (DNN) getestet[27]. Die Standardarchitektur Multilayer Perceptron (MLP), Long Short-Term Memory Network (LSTM), ein CNN/LSTM Hybrid, eigene MLP-Architekturen sowie multimodale MLP (vgl. Abbildung 26 und multimodale CNN wurden bewertet. Alle Ansätze erhalten Datenfenster der Größe  $T \times S$  als Input, mit  $T = 420$  aufgenommene Datenpunkte pro Sekunde jedes Sensors und  $S = 9$  Sensorkanäle. Jedes Modell hat einen Softmax-Output-Layer mit vier Neuronen, jeweils einen pro Klasse. Da die Softmax-Funktion monoton ist und die Summe aller Ergebnisse immer eins ergibt, ist jedes einzelne Ergebnis auch gleichzeitig die Wahrscheinlichkeit, dass der aktuelle Input zu einer Klasse gehört. Alle Modelle wurden mit einem ADADELTA-Optimierer[28] trainiert, wobei die entsprechenden Hyperparameter (z.B. Anzahl der Layer, Neuronen pro Layer, ...) manuell ausgewählt wurden.

Der verwendete MLP hat drei zusätzliche versteckte Layer mit 1.000 Neuronen und ReLU-Aktivierung[29]. Alle Layer sind verbunden, d.h. der Output aller Neuronen

eines Layers werden als Input eines jeden einzelnen Neutronen auf den nächsten Layer.

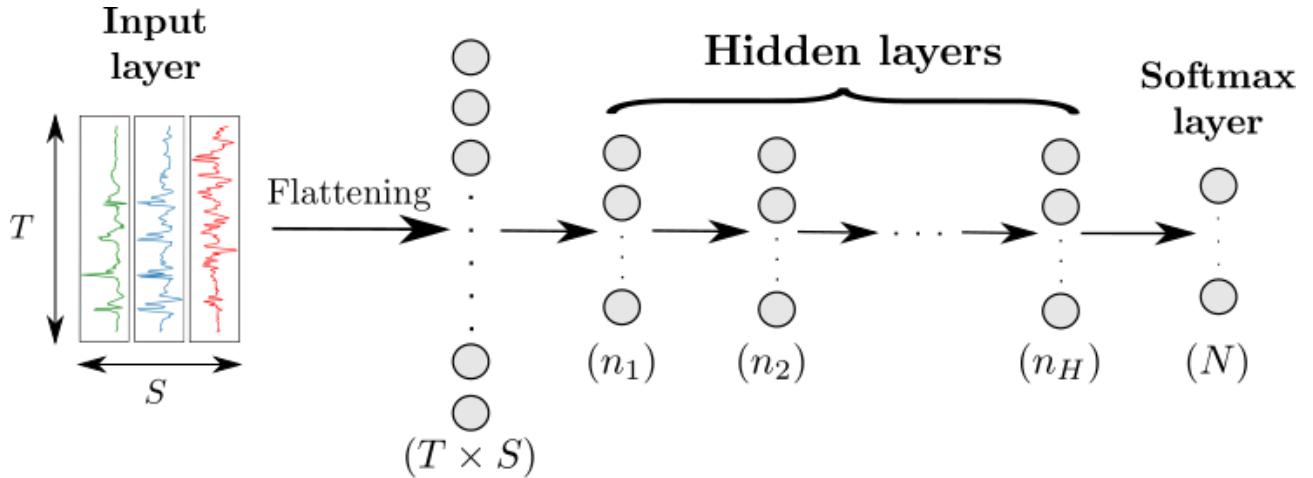


Abbildung 24: Architektur eines MLP Network.

Die verwendete CNN besteht aus drei faltungsbedingten Layern mit zehn Filter-Kernels und jeweils einem ReLU. Die Kernel-Größe ist  $41 \times 1$  für den ersten Layer,  $21 \times 1$  für den zweiten und  $11 \times 1$  für den dritten Layer. Da die zweite Dimension immer eins ist, funktioniert jeder Kernel immer nur an einem Sensorkanal. Alle faltungsbedingten Layer werden gefolgt von einem Maxpooling-Layer der Größe  $2 \times 1$ , welcher den Output um den Faktor zwei downsamplet.

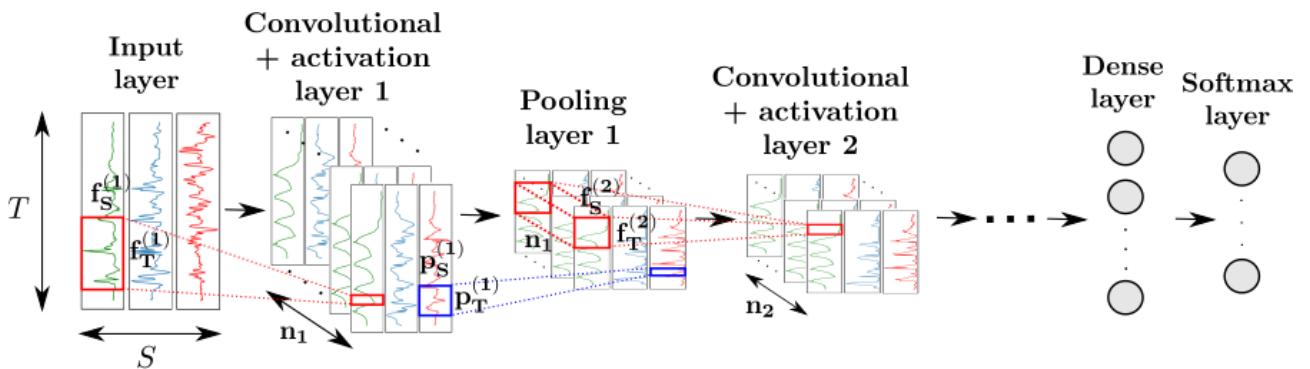


Abbildung 25: Architektur eines CNN Network.

Für MLP und CNN wurde zusätzlich ein Multimodal-Network getestet, welche auch bekannt als mMLP bzw. mCNN sind. Anstatt das ganze Zeitfenster als Input zu benutzen, wird hier jeweils ein Neural-Network mit drei versteckten Layern für jeden Sensorkanal verwendet (siehe Abbildung [?]). Die Ergebnisse wurden zusammengeknüpft und als Input für ein zusätzlichen verbundenen Layer genutzt.

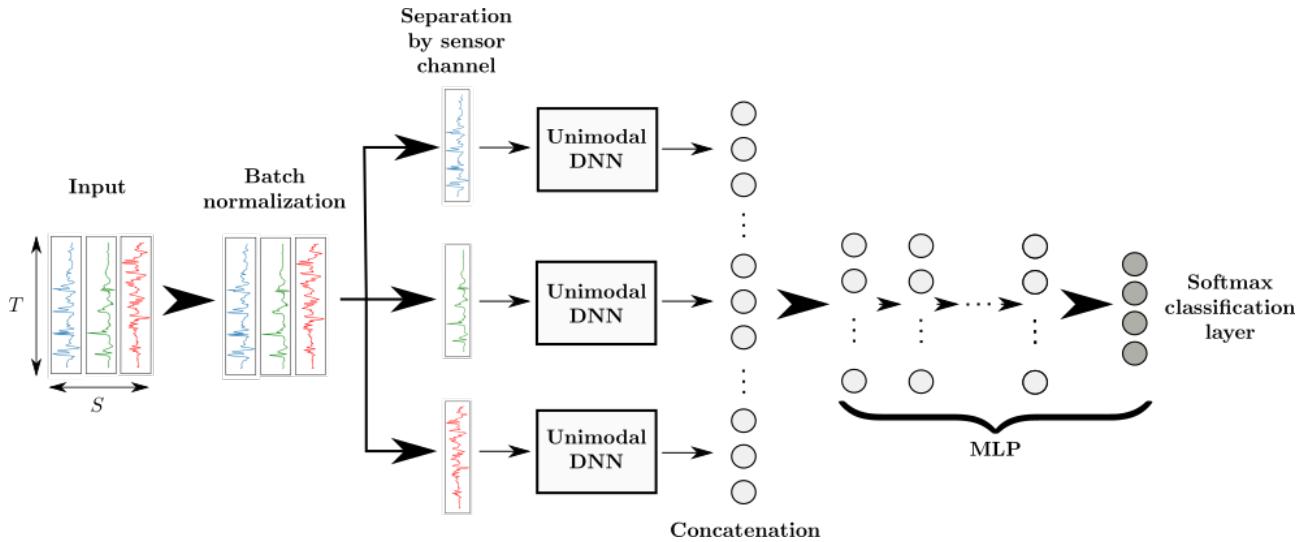


Abbildung 26: Eigene Architektur eines Multimodal-MLP-Network.

LSTMs[30] sind eine spezielle Form von wiederkehrenden Neural-Networks. Sie sind in der Lage, Informationen zu speichern, um Muster über die Zeit zu erkennen. Nicht wiederkehrende Neural-Networks lernen im Gegensatz dazu nur von den aktuellen Input-Daten. Das Verhalten über die Zeit gesehen wird von den sogenannten "Gates" kontrolliert, welche den Informationsfluß innerhalb eines Networks beeinflussen können. Das "Input-Gate" sakralisiert den Einfluss des neuen Daten-Inputs, während das "Forget Date" den Einfluss der gespeicherten Informationen über den vorherigen Zustand des Neurons steuert. Am Ende wird der neue Output von dem "Output-Gate" modifiziert.

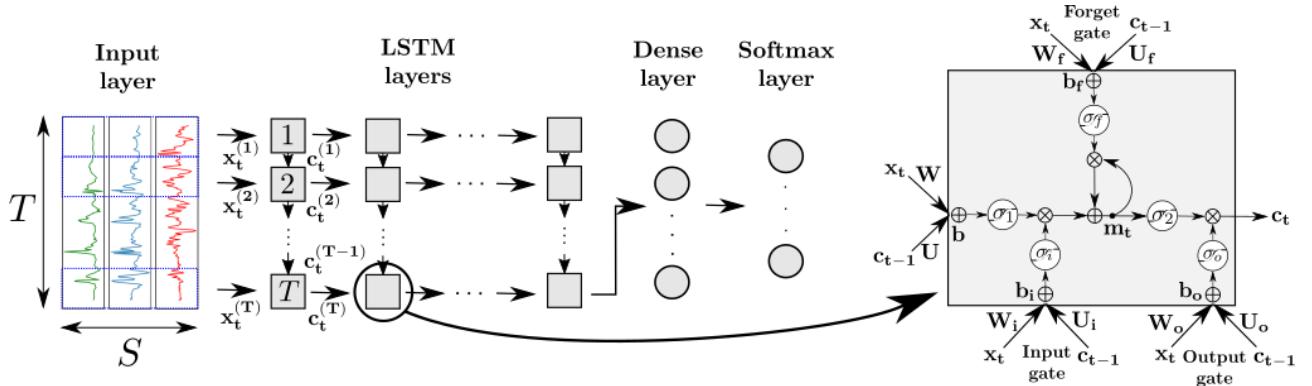


Abbildung 27: Architektur eines LSTM-Networks.

Das CNN/LSTM-Hybrid-Modell stapelt einfach die beiden Networks. Die ersten Layer sind die der vorhin geschriebenen CNN, welche dann von den Layern der LSTM-Layer gefolgt werden.

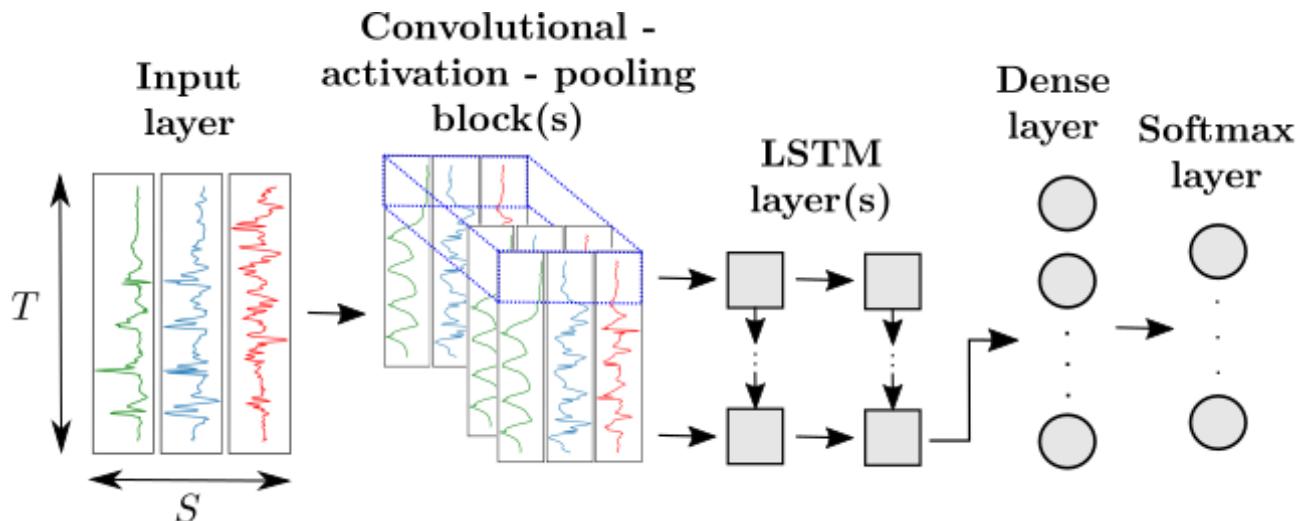


Abbildung 28: Architektur eines CNN/LSTM-Hybriden.

#### 9.4.4 Merkmals Auswahl

Um bessere Kombinationen von handgefertigten Merkmalen zu finden, kann ein Bottom-Up-Merkmal-Auswahlalgorithmus verwendet werden, der auf der Prüfung von Gruppen von Merkmalen mit zunehmender Größe basiert. Zunächst wird das Merkmal mit der höchsten Klassifizierungsperformance unter allen verfügbaren Merkmalen ausgewählt. Anschließend wird die Performance von Gruppen von zwei Merkmalen, die sich aus dem ausgewählten Merkmal und der Reihe nach jedem anderem Merkmal zusammensetzen, werden dann getestet und das beste Paar wird ausgewählt. Dieser Prozess wird so lange wiederholt wiederholt, bis alle relevanten Merkmale verwendet wurden. Am Ende wird die beste Kombination von Merkmalen ausgewählt. Es sei darauf hingewiesen, dass es sich um einen heuristischen Algorithmus handelt, so dass es möglich ist, dass unter Umständen die absolut beste Kombination nicht zwingend gefunden wird.

Der folgende Pseudocode beschreibt unseren Algorithmus zur Merkmalsauswahl:

---

**Algorithm 1:** Merkmalsauswahl-Algorithmus.

---

```

1 Input parameters:
2   -  $C, \gamma$  = C-SVM params
3   -  $candidates = [f_1, f_2, \dots, f_n]$  list of n features to test
4   -  $training\_set$  = set of all features computed on the training data
5   -  $testing\_set$  = set of all features computed on the testing data
6 Output parameters:
7   -  $best\_feature\_combination$  = list of the best features
8   -  $best\_accuracy$  = classification accuracy obtained by the best features
9   -  $feature\_ranking$  = list ranking features in decreasing order of
  relevance
10
11 Begin
12  $candidates\_to\_test = candidates$ 
13  $current\_best\_features = \emptyset$ 
14  $current\_best\_accuracy = -1$ 
15  $all\_time\_best\_features = \emptyset$ 
16  $all\_time\_best\_accuracy = -1$ 
17  $feature\_ranking = \emptyset$ 
18 while  $candidates\_to\_test \neq \emptyset$  do
19   for  $feature_f$  in  $candidates\_to\_test$  do
20      $trained\_svm =$ 
      $train\_svm(C, \gamma, training\_set, current\_best\_features \cup \{f\})$ 
21      $accuracy =$ 
      $evaluate\_svm(trained\_svm, testing\_set, current\_best\_features \cup \{f\})$ 
22     if  $accuracy > current\_best\_accuracy$  then
23        $best\_feature\_of\_iteration = f$ 
24        $current\_best\_accuracy = accuracy$ 
25     if  $accuracy > all\_time\_best\_accuracy$  then
26        $all\_time\_best\_features = current\_best\_features \cup \{f\}$ 
27        $all\_time\_best\_accuracy = accuracy$ 
28    $feature\_ranking = feature\_ranking \cup [best\_feature\_of\_iteration]$ 
29    $current\_best\_features =$ 
      $current\_best\_features \cup [best\_feature\_of\_iteration]$ 
30    $candidates\_to\_test = candidates\_to\_test \setminus [best\_feature\_of\_iteration]$ 
31 return  $all\_time\_best\_features, all\_time\_best\_accuracy, feature\_ranking$ 

```

---

## 9.5 Klassifikation

Verantwortlich: Artur  
- RfP

Wie bereits in Kapitel 3.8.5 beschrieben ist das Ziel der Klassifizierung ein Klassifizierungsmodell zu trainieren, das in der Lage ist, Objekte in den Daten in die entsprechende Klasse zuzuordnen. Die Klassen entsprechen hierbei den Emotionen, die erkannt werden sollen: Glück, Langeweile, Frustation und andere (d.h. alle Emotionen, die nicht Glück, Langeweile oder Frustation entsprechen).

Als erstes wird der Datensatz in ein Trainings- und Testset aufgeteilt. Es gibt keine festgelegten Regeln über die Proportionen der Sets. Im Allgemeinen wird das Trainingsset aber größer als das Testset gewählt. Da die Leistungen des Klassifikators jedoch stark von der gewählten Aufteilung abhängen, ist es wichtig, sicherzustellen, dass dieser Schritt richtig durchgeführt wird. Bei einem Datensatz mit mehreren Probanden empfiehlt sich für die Aufteilung zwischen Trainings- und Testsets die Durchführung einer Leave-One-Subjekt-Out-Cross-Validierung (LOSOCV). Die Idee besteht darin,  $N$  verschiedene Aufteilungen des Datensatzes vorzunehmen, wobei  $N$  die Anzahl der Personen ist, die Daten für den Datensatz bereitgestellt haben. Für jeden dieser Splits wird der Testset aus den Daten eines Probanden aufgebaut, während die Daten der anderen Probanden das Trainingsset bilden. Anschließend wird ein Klassifizierer erstellt und ausgewertet. Dies wird für alle Probanden wiederholt, d.h.  $N$  mal. Die so erhaltenen  $N$ -Bewertungskennzahlen (eine pro Proband) können dann gemittelt werden, um eine Gesamtbewertung des Modells zu erhalten. Es ist wichtig zu beachten, dass der LOSOCV-Ansatz bei einer hohen Anzahl von Probanden sehr rechenintensiv sein kann.

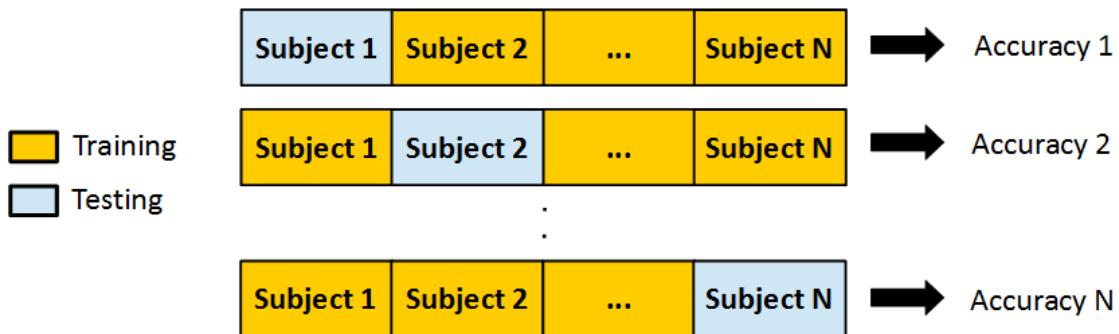


Abbildung 29: Leave-One-Subjekt-Out-Cross-Validation (LOSOCV):  $N$  entspricht der Anzahl der Probanden. Für jeden Split wird ein Testset aus den Daten eines Probanden aufgebaut, während die Daten der anderen Probanden einen Trainingsset bilden. Dieser Vorgang wird für die Daten jedes Probanden durchgeführt.

# 10 Ergebnisse

Verantwortlich: Artur  
- RfP

In den nächsten Unterkapiteln werden unsere Ergebnisse und Analysen für handgefertigte Merkmale und den Codebook Approach vorgestellt.

## 10.1 Ergebnisse der hand-gefertigten Merkmale

Zunächst haben wir die Daten mit einer Normalisierungstechnik vorverarbeitet, die bereits im Kapitel 3.8.1 beschrieben wurde. Entsprechend der ERC haben wir dann den gleitenden Zeitfenster-Segmentierungsansatz (vgl. Kapitel 9.3) mit den Parametern  $T = 120$  und  $\sigma = 30$  verwendet. Um die optimalen Parameter des SVM-Klassifikators (d.h. Soft-Margin  $C$  und Kernelparameter  $\gamma$ ) zu bestimmen, wurde ein Gitter-Suchansatz verwendet. Es besteht darin, einen Satz möglicher Werte für jeden Parameter zu definieren und die Leistungen des Klassifikators für jedes Paar möglicher Werte zu testen (d.h. an jedem “Knoten des Gitters”).

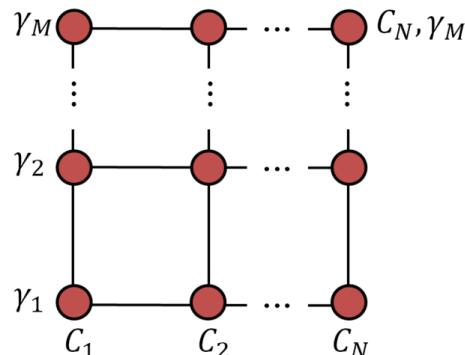


Abbildung 30: Gitter-Suche: Ein definierter Satz möglicher Werte für jeden Parameter wird an jedem “Knoten des Gitters” getestet. Diese Knoten sind in der Abbildung als rote Punkte markiert.

Die folgenden Werte für  $\gamma$  und  $C$  wurden getestet:

$$\begin{aligned}\gamma &\in \{0.002; 0.008; 0.03; 0.15; 0.5; 1; 2\} \\ C &\in \{0.5; 1; 2; 8; 32; 128; 512\}\end{aligned}$$

Da alle Probanden nicht alle Ziel-Emotionen erlebt bzw. angegeben haben, konnte die Leave-One-Subject-Out Kreuzvalidierung nur bedingt für die Bewertung angewendet werden. Wir haben daher eine 5-fache Kreuzvalidierung mit entsprechendem F1-Wert verwendet.

Die Ergebnisse unserer Experimente zeigen, dass ein solcher Aufbau die Erzielung einer relativ hohen Genauigkeit für die Erkennung der vier Klassen ermöglicht. Die

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Durchschnitt
90.18	93.06	91.63	90.61	91.98	91.49

Tabelle 4: Durchschnittlicher F1-Wert in % für die fünf Falten des Datensatzes.

Confusion-Matrizen für den handgefertigten Merkmalsansatz unter Verwendung der SVM-Klassifikation ist in der folgenden Tabelle zu finden.

	Sonstige	Glück	Frustation	Langeweile
Sonstige	<b>95.21</b>	1.98	0.29	2.51
Glück	7.84	<b>90.93</b>	0.39	0.84
Frustation	8.49	2.50	<b>84.14</b>	4.87
Langeweile	5.69	0.70	0.94	<b>92.67</b>

Tabelle 5: Confusion Matrix in % für die fünf Falten des Datensatzes.

## 10.2 Ergebnisse des Codebook Approach

Ähnlich wie bei den handgefertigten Merkmalen haben wir die Daten mit einer Normalisierungstechnik vorverarbeitet und dann die Segmentierung verwendet. Die Zeitfensterparameter sind identisch wie bei der Studie mit den handgefertigten Merkmalen. Für die Klassifizierung haben wir SVM mit Soft-Margins und dem RBF-Kernel benutzt. Um die optimalen Parameter des SVM-Klassifikators zu bestimmen (z.B. Soft-Margin  $C$  und Kernelparameter  $\gamma$ ), wurde hier wieder die Gitter-Suche angewendet.

Die Ergebnisse, die wir mit dem CA mit fester Zuordnung (hard assignment) und  $C = 8$ ,  $\gamma = 0,002$  für jeden Probanden erhielten, waren 52%, 38% und 38%, was einem Durchschnitt von 42,67% entspricht. CA mit Soft-Assignment wurden ebenfalls getestet, lieferte aber schlechtere Ergebnisse als CA mit Hard-Assignment. In diesem speziellen Datensatz schneidet der CA also schlechter ab als die handgefertigten Merkmale.

## 10.3 Ergebnisse der Deep Neural Networks

Die Verwendung von DNNs zur Extraktion von Merkmalen (z.B. Multi-Layer-Perceptron, Convolutional Neural oder Long-Short-Term-Memory Networks) wurden zwar intensiv getestet (siehe [27]), diese Ansätze erreichten aber leider nicht so gute Ergebnisse, wie z.B. die oben vorgestellten Ergebnisse der handgefertigten Merkmale. Die beste Performance erreichten LSTM mit einem F1-Score von 47,99%.

## 10.4 Analyse der Ergebnisse

F1-Score (%)	Probanden-abhängig	Probanden-unabhängig
<b>Handgefertigte Merkmale</b>	91,49%	28,85%
<b>Codebook Approach</b>	42,67%	-
<b>Deep Neural Networks</b>	47,99%	30.91%

Tabelle 6: Vergleich der Ergebnisse der unterschiedlichen Merkmalsextraktionen.

Obwohl die handgefertigten Merkmale hier die besten Ergebnisse erzielt haben, wird davon ausgegangen, dass die Performance in Realität deutlich abnehmen könnte. Der Hauptgrund dafür ist ein sehr Wahrscheinliches Overfitting zu den einzelnen Probanden. Unterstützt wird das durch eine vorläufige qualitative Analyse mithilfe der t-Distributed Stochastic Neighbor Embedding (t-SNE) Methode (siehe Abbildung 31). Oben sieht man die Labels entsprechend den Emotionen und unten entsprechend der Probanden. Es ist klar zu erkennen, dass die Probanden klarere Klassen bilden und damit die handgefertigten Merkmale nicht Testpersonen-unabhängig klassifizieren.

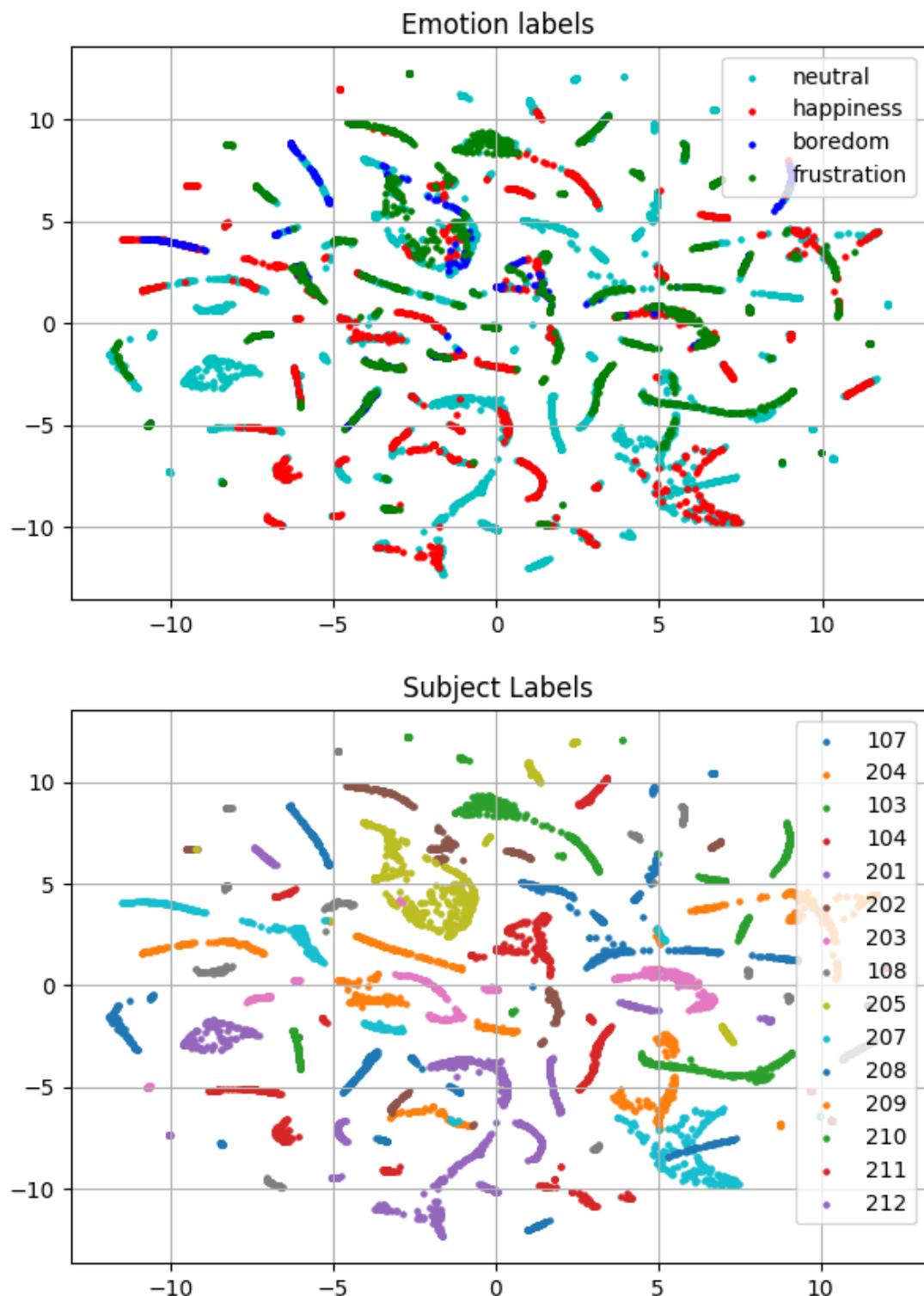


Abbildung 31: t-SNE Plots der handgefertigten Merkmale.

Des Weiteren wurden an einem Probanden-unabhängigen Datensatz ausgetestet, woraufhin die Performance von 91,49% auf nur 28,85% sehr stark sank. Aus diesen Gründen werden die handgefertigten Merkmale wahrscheinlich mehr von den einzelnen Probanden beeinflusst, als von den unterschiedlichen Emotionen an sich.

Entgegen unserer Erwartungen liefert der CA schächtere Performance als die

handgefertigten Merkmale. Der Grund hierfür ist aber sehr wahrscheinlich der selbe wie bei den DNNs, und zwar der relativ kleine Datensatz. Eine kleine qualitative Analyse der in der Studie verwendeten Daten kann die Gründe für diese Beobachtungen begründen. CA-Merkmale (sowohl für weiche als auch für harte Zuweisungen) sind per Definition sehr empfindlich gegenüber Variationen der Formen der Originalsignale: Die Codewörter werden durch Clustering auf Sätzen von Segmenten bestimmt, die aus den Originalsignalen extrahiert wurden, und die histogrammbasierten Merkmale selbst basieren auf direkten Vergleichen zwischen den Codewörtern und dem Segment der zu klassifizierenden Daten. Daher kann jede Quelle von Rauschen oder Unregelmäßigkeiten in den Originaldaten die Effektivität der CA-Funktionen stark beeinträchtigen. Ein Blick auf die in unserer Studie verwendeten Signale ergab zwei Hauptprobleme: falsche Datenwerte, die durch Hardwareprobleme bei einigen Sensoren verursacht wurden (wie in Abbildung 32), und das Vorhandensein von Rauschen, das Unregelmäßigkeiten in den Signalformen verursacht (siehe Abbildung 33). Da der CA schon bei dem Probanden-abhängigen Datensatz keine vielversprechende Ergebnisse aufweiste, wurde darauf verzichtet den CA bei dem Probanden-unabhängigen Datensatz zu testen. Aus diesem Grund ist in Tabelle 10.4 an der entsprechenden Stelle nur ein “-“ eingefügt worden. Mögliche Lösungen zur Behebung dieses Problems könnten sein, zusätzliche Vorverarbeitungstechniken zur Rauschunterdrückung einzusetzen, wie z.B. Tiefpassfilterung.

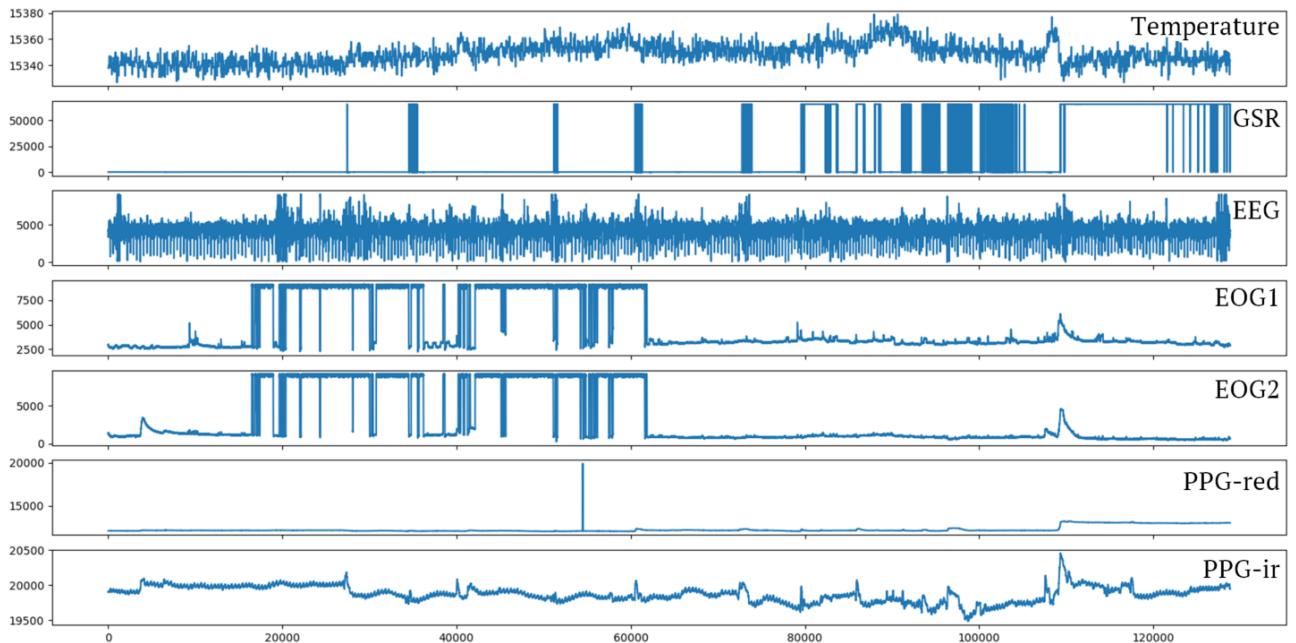


Abbildung 32: Sensorsignale, die im Rahmen des ELISE-Projekts von einem der drei getesteten Probanden erfasst wurden. Die x-Achse repräsentiert die Zeit und die y-Achse die Sensorwerte. Die in den Daten von GSR und den beiden EOG-Kanälen sichtbaren Unregelmäßigkeiten deuten auf Probleme mit der Hardware hin.

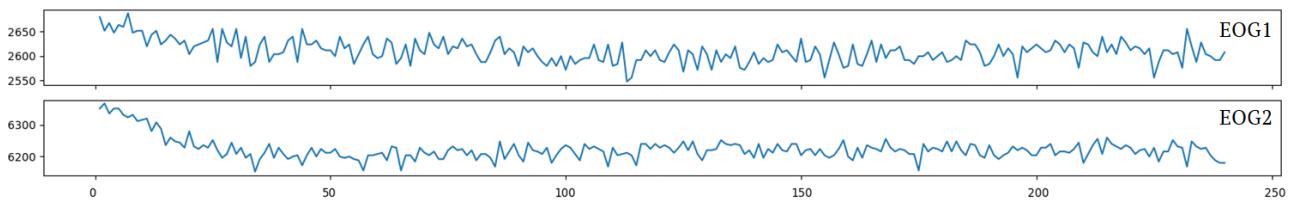


Abbildung 33: Nahaufnahme der im Rahmen des ELISE-Projekts erworbenen EOG-Kanäle von einem der drei getesteten Probanden. Das Vorhandensein von Rauschen in den Daten ist sichtbar, das zu Unregelmäßigkeiten in den Signalformen führt.

Der Hauptgrund, warum DNNs keine überzeugenden Ergebnisse lieferten liegt wahrscheinlich an der Erkennungsraten für die am wenigsten vertretenen Klassen (insbesondere “Frustration”). Wir gehen davon aus, dass dieses Phänomen durch die relativ geringe Größe unseres Datensatzes verursacht wird. Ähnlich wie bei den handgefertigten Merkmalen, wird bei den DNNs auch von einem Overfitting an die Probenaden ausgegangen. Das liegt daran, dass die Performance bei einem Probanden-unabhängigen Datensatz von 47,99% auf nur 30,91% sinkt. Mit einem deutlich größeren Datensatz erwarten wir eine bessere Performance der DNNs.

Allgemein deuten die Ergebnisse aber trotzdem darauf hin, dass unser biomedizinisches Datenerfassungssystem zur Emotionserkennung erfolgreich eingesetzt werden könnte, um ein intelligent adaptives Lernsystems zu verbessern. Zukünftige Arbeiten werden die Verfeinerung des Multisensor-Datenerfassungsgerätes, die Erfassung weiterer und größerer Datensätze für die weitere Mustererkennungsanalyse und die Analyse der Wirksamkeit des Emotionserkennungssystems in einem VR-affektiven Lernkontext beinhalten.

## Part II

# 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-Sensor

#### 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 Software

Verantwortlich: Jonas, Kevin

## Part III

### Dritter Prototype

## 13 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

### 13.1 Anforderungen

### 13.2 Konzept

### 13.3 Hardwareauswahl

#### 13.3.1 Auswahlkriterien

#### 13.3.2 Festlegung der genutzten Hardware

### 13.4 Hardwarearchitektur

#### 13.4.1 GSR-Sensor

#### 13.4.2 Temperatur-Sensor

#### 13.4.3 Pulsoximeter

#### 13.4.4 EEG

#### 13.4.5 EOG

#### 13.4.6 Datenübertragung

### 13.5 Programmierung

### 13.6 Aufnahme der übertragenen Daten

## 14 Software

Verantwortlich: Jonas, Kevin

# 15 Emotionsinduktion

Verantwortlich: Minas  
- RfP

Das Kapitel beinhaltet einen Ablauf, die verwendeten Fragebögen und die genutzten Szenarien (Glück, Langeweile, Frustration) der Emotionsinduktion für den dritten und finalen Prototypen. Kapitel Ablauf beschreibt in welcher Reihenfolge die Szenarien und die Fragebögen aufgerufen wurden. In den Kapiteln Fragebogen, Glücks-, Langeweile- und Frustration- Szenario wird die genaue Entwicklung und Implementierung in einer VR-Umgebung erläutert. Die Emotionsinduktion wurden sowohl in Englisch als auch in Deutsch angeboten.

## 15.1 Ablauf

Verantwortlich: Minas  
- RfP

Der Ablauf der Emotionsinduktion lief wie folgt ab:

1. Einführung
2. WarmUp
3. Glücks-Szenario → Kapitel 15.3.1
4. Langeweile-Szenario → Kapitel 15.3.2
5. Frustrations-Szenario → Kapitel 15.3.3
6. Schlussworte

Die Emotionsinduktion fing mit einer Einführung an, welche das Projekt vorstellt. Es sollte ein breites Spektrum von Probanden angesprochen werden. Viele von ihnen hatten jedoch noch keinen Kontakt zu einer VR-Anwendung oder zur VR-Hardware. Um die Emotionsinduktion dennoch erfolgreich abzuschließen, wurde ein WarmUp-Szenario erstellt. In diesem sollten die Probanden einen ersten Einblick in die VR-Anwendung bzw. VR-Hardware bekommen. Das WarmUp-Szenario bestand aus einem 360 Grad Bild, worauf ein kleiner Fragebogen erschien und aus einem Hot-Wire Level. Beide Szenarien wurden anhand von Hilfstexten und einer Bedienungsanleitung des VR-Controllers ausgiebig erklärt. Sobald das WarmUp-Szenario abgeschlossen wurde, folgte der Hauptteil. Dieser Bestand aus dem Glücks-Szenario, Langeweile-Szenario und dem Frustrations-Szenario. Vor jedem Szenario wurde erneut erklärt, was den Probanden bevorsteht und eine Bedienungsanleitung des VR-Controllers angezeigt. Nach jedem Szenario wurde ein Fragebogen angezeigt, welcher in Kapitel 15.2 erklärt wird. Nachdem alle Szenarien durchlaufen wurden, kam vor der Danksagung ein Fragebogen über die psychische und körperliche

Verfassung. In diesem wurde abgefragt, ob der Proband unter einer psychischen Störung leidet oder sich in einer speziellen medizinischen Behandlung befindet. Es wurde keine genaue Angabe gefordert, lediglich ein "ja" oder ein "nein" konnten angekreuzt werden.

## 15.2 Fragebogen

Verantwortlich: Boris  
- RfP

Für dieses Prototyp wurden zwei Arten von Fragebögen benutzt. Der erste Typ ist sehr ähnlich mit dem von der zweiten Prototyp. Es enthält in dem informativen Teil einen Text, wo es beschrieben wird, wie der Fragebogen ausgefüllt werden soll. Der andere Teil besteht aus vier Dropdown-Boxen von dreizehn Optionen, die zwölf verschiedene Emotionen und einen als null oder neutral geltenden Zustand enthalten. Jede Dropdown-Box entspricht einer Viertelzeit des Szenarios. Es soll zwischen den Optionen jeder Dropdown-Box gewählt werden, welche Emotion es am stärksten empfindet wird, je nachdem, wann man es fühlte, d.h. ob es das erste, zweite, dritte oder letzte Quartal der Zeit des Videos war, um die Emotion zu bewältigen. Unten gibt es einen Button wo man drücken kann, wenn man fertig ist. Allerdings hat man auch die Möglichkeit seine Wahl zu ändern, auch wenn man sich schon im nächsten Schritt befindet, indem man in diesem nächsten Schritt auf den Zurück-Button drückt und die gewünschten Änderungen vornimmt. Hierfür wird ein Widget Blueprint erstellt, die vier Dropdown-Boxen mit dem gewünschten Anzahl an Optionen hinzugefügt und die Labels von den unterschiedlichen Optionen der Dropdown-Boxen definiert. Ein Button "next" wird auch erstellt um zum nächsten Fragebogen zu navigieren. Es wird auch eine zwischen Speicherungsfunktion in ein anderes Skript definiert, die hier aufgerufen wird, um die Änderung auch nach dem Drücken von dem "next" Button zu Speichern. Dabei wird vier Variablen definiert und die Werte von den gewählten Optionen werden ihnen zugewiesen.



Abbildung 34: Bild des Fragebogen-Teils, entsprechend dem circumplex-Modell.

Der zweite Typ ähnelt dem berühmten Modell von James Russells “circumplex” [31]. Es ist ein klassisches Modell mit einer kreisförmigen Struktur, die auf zwei senkrechten Diagonalen ruht. Die vertikale Achse, die die Erregung darstellt, und die horizontale Achse, die die Valenz darstellt. Das Zentrum des Kreises stellt eine neutrale Valenz und ein mittleres Erregungsniveau dar. Andere Emotionen werden auf jeder Ebene des Kreises dargestellt. Hier wird ein weniger bekanntes Modell verwendet, das “Self Assessment Manikin” (SAM). Es besteht aus drei Reihen mit je fünf Piktogrammen. Diese Piktogramme stellen den Zustand eines Gesichts nach verschiedenen Arten von Emotionen dar. So repräsentiert der erste Bereich die Wertigkeit, der zweite die Erregung und der dritte die Dominanz. Eine Erklärung zu jedem dieser Begriffe ist ebenfalls neben dem Fragebogen enthalten, um die Testpersonen über diese Wörter aufzuklären. Bei jeder Avatar und in der Mitte jeder der beiden Avatar befindet sich eine Checkbox. So muss man für jede Zeile die Checkbox auswählen, die ihrem emotionalen Zustand am besten entspricht. Man kann nur eine Checkbox pro Zeile markieren und man hat auch die Möglichkeit wie bei dem ersten Modell seine Wahl zu ändern. Es kann einfach mit Branch-Bedingungen realisiert werden. Diese werden auch in eine Widget Blueprint wie für das erste Modell gemacht. Es gibt auch wieder die zwischen Speicherungsfunktion und das Button “next”. Was neues hier kommt ist das Button “back” um wieder zum ersten Fragebogen zu navigieren.

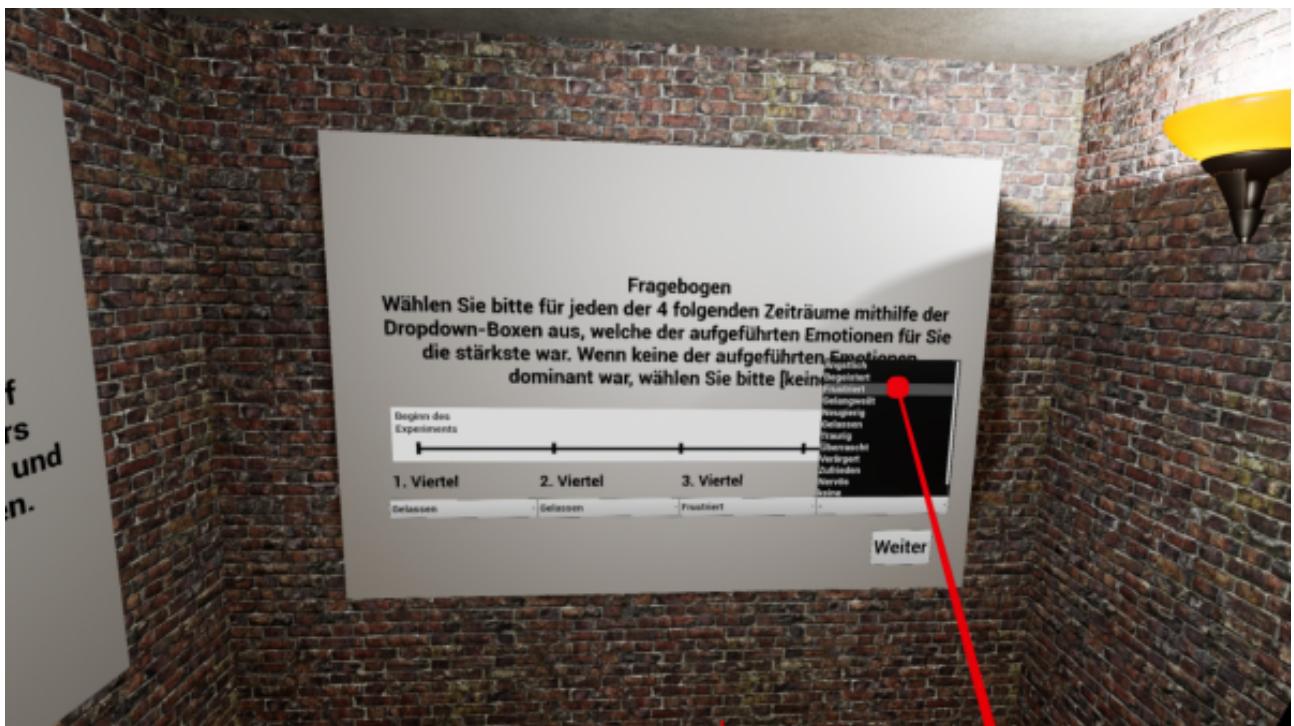


Abbildung 35: Bild des Fragebogen-Teils, wo die dominierenden Emotionen abgefragt werden.

## 15.3 Szenarien

Verantwortlich: Meryem  
 - RfP

Im Folgenden werden drei verschiedene Szenarien dargeboten. Wie bereits im Kapitel 7 sollen bei den Probanden die Emotionen Glück, Langweile und Frustration ausgelöst bzw angeregt werden. Die Emotionsinduktionen sollen dieses Mal jedoch innerhalb einer VR-Umgebung stattfinden.

### 15.3.1 Glück

Verantwortlich: Minas  
 - RfP

Um das Glück-Szenario in einer VR-Umgebung zu verwirklichen, wurden sich für HDR-Panorama-Bilder entschieden, die in Unreal eine Umgebung bilden sollen. Der Grund weswegen die Bilder HDR und Panorama sein müssen, werden im Laufe dieses Kapitels erklärt. Außerdem wird eine Audio-Datei im Hintergrund abgespielt und ein Text eingeblendet, welches thematisch zum Bild passt. Die Grundidee stammt vom ersten Prototypen (Kapitel 7.3.1), welches nicht in einer VR-Umgebung gelöst wurde. In Kapitel 7.3.1 wurde zudem erklärt, weshalb und welche Audio-Datei im Hintergrund abgespielt wird, weshalb die Texte eingeblendet werden und was sich unter Glück verstehen lässt.

Insgesamt besteht das Glück-Szenario aus acht HDR-Panorama-Bilder für das Hauptscenario und ein HDR-Panorama-Bild für das Warm-Up-Szenario. Abbildung 36 zeigt alle Bilder die genutzt werden und deren Texte (falls vorhanden).

## WarmUp



## Hauptteil



You spend a day in the mountains. The air is pure and clean, the day is sunny and you go swimming in a beautiful lake.



You wake up on a Saturday morning after a few cold and rainy days. The temperature is about 20 degrees now.



You leave school or work early. It is a wonderful day.



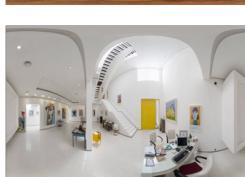
It is your birthday and your friends arrange a great surprise party.



You go with a friend or a pal to a nice restaurant. The food, the entertainment and the atmosphere are perfect.



A romantic evening with your girlfriend/boyfriend.



You have got a new job and it is much better than you have expected.



After a hard day, it is time to relax.

Abbildung 36: Im Glück-Szenario verwendete Bilder und deren Texte[32].

Die Bilder lassen sich jedoch nicht ohne weiteres in Unreal einbinden. Um dies zu realisieren wurde sich für eine CubeMap in DDS-Format entschieden. Hierfür müssen die Bilder zunächst durch verschiedene Tools (Blender, Photoshop) bearbeitet werden und dann mit bestimmten Konfigurationen in Unreal eingebunden werden.



Abbildung 37: Ein im Projekt verwendetes HDR-Panorama-Bild.

Anhand Abbildung 37 wird die Bearbeitung der Bilder erklärt. Nach Auswahl eines Panorama-HDR-Bildes, wird dieses in Blender bearbeitet. Blender wird benötigt, um aus dem gesamten Bild, sechs Einzelbilder mit der benötigten Rotation zu erzeugen. Das ist der erste Schritt um später eine CubeMap zu erzeugen. Im Internet existiert bereits eine Blender-File mit den benötigten Konfigurationen[33], um an diese Einzelbilder zu gelangen. Diese besteht aus einer Kamera, welche um das importierte Bild rotiert und dieses zurecht schneidet. Handelt es sich um kein Panorama Bild, werden die sechs Bilder falsch geschnitten und können dadurch die Kriterien einer CubeMap nicht erfüllen.

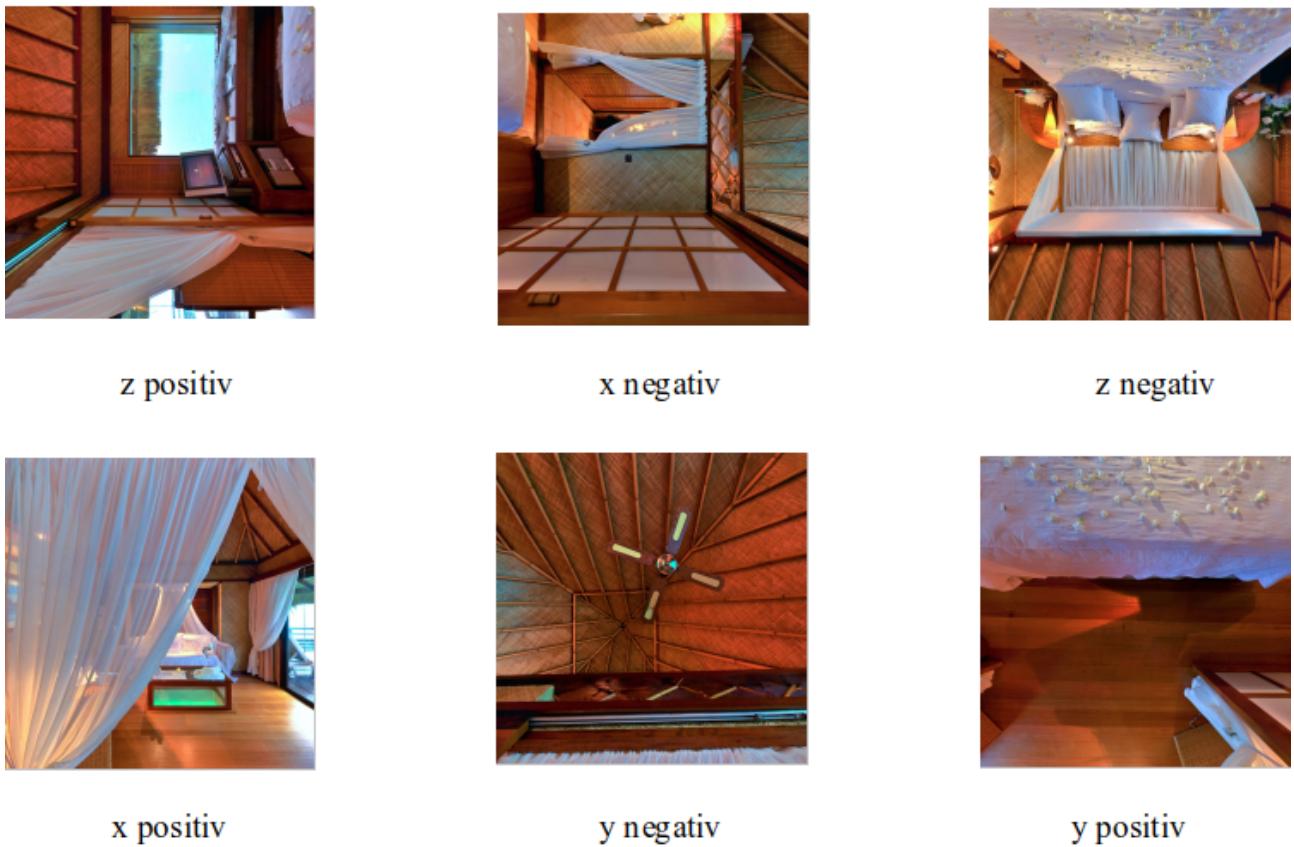


Abbildung 38: Ausgabe der Blender-File und deren Rotation.

Um eine CubeMap in Unreal einzubinden, wird jedoch nur ein Bild im DDS-Format benötigt und nicht sechs Einzelbilder. DDS steht für Direct Draw Surface und ist ein entwickeltes Format von Microsoft. Dieses Format wird hauptsächlich für die Speicherung von CubeMaps und Texturen verwendet und erhöht die Geschwindigkeiten in Spielen ohne Verlust von Details[34]. Nvidia bietet ein Plugin für einige Adobe Photoshop Versionen und GIMP, um CubeMaps in DDS-Format abzuspeichern. Der nächste Schritt eine CubeMap zu erstellen, wurde mit Adobe Photoshop CS2 und dem Nvidia Texture- Plugin realisiert. Hierfür müssen die Bilder in Abbildung 38 in richtiger Reihenfolge aneinander gereiht werden. Dies geschieht in Adobe Photoshop. Die benötigte Reihenfolge wird in Abbildung 39 gezeigt.



Abbildung 39: Rotation und Reihenfolge für eine CubeMap[35].

Somit ergibt sich für die Bearbeitung der Abbildung 38, Abbildung 40.



Abbildung 40: Rotation und Reihenfolge für die CubeMap der Abbildung 38.

Nach einer aneinander Reihung der einzelnen Bilder, lässt sich das dadurch entstandene Bild mit Hilfe des Nvidia Plugins in eine DDS-File exportieren. Somit wird die gewünschte Cubemap generiert. Abbildung 41 zeigt die aufgeklappte Form einer CubeMap. Jetzt kann die DDS-File in Unreal importiert bzw. konfiguriert werden.

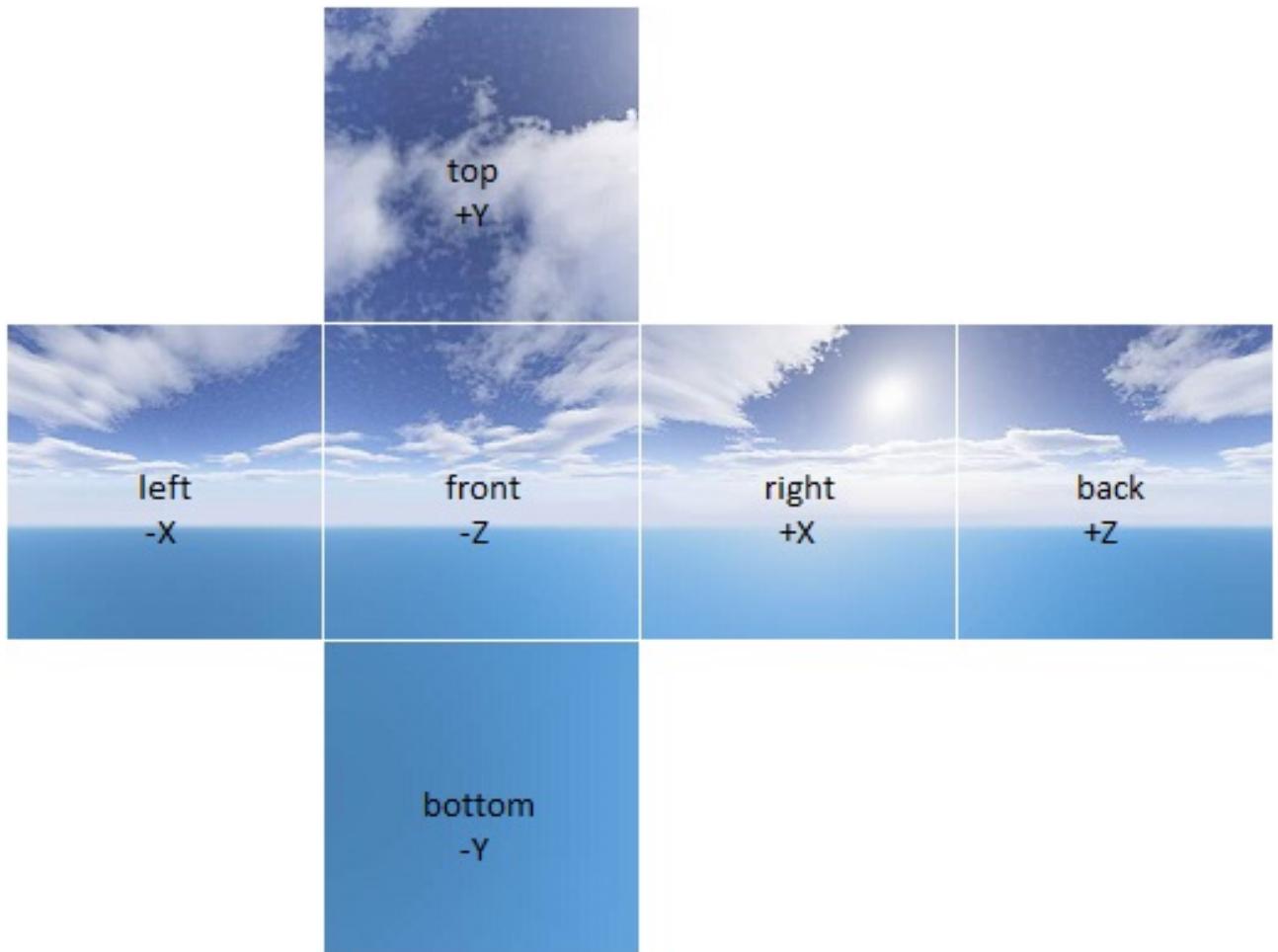


Abbildung 41: Aufgeklappte CubeMap[36].

Sobald die DDS-File im Unreal-Projekt importiert wurde, muss an der DDS-File selbst Einstellungen vorgenommen werden. Abbildung 42 beinhaltet die optimalen Einstellungen.

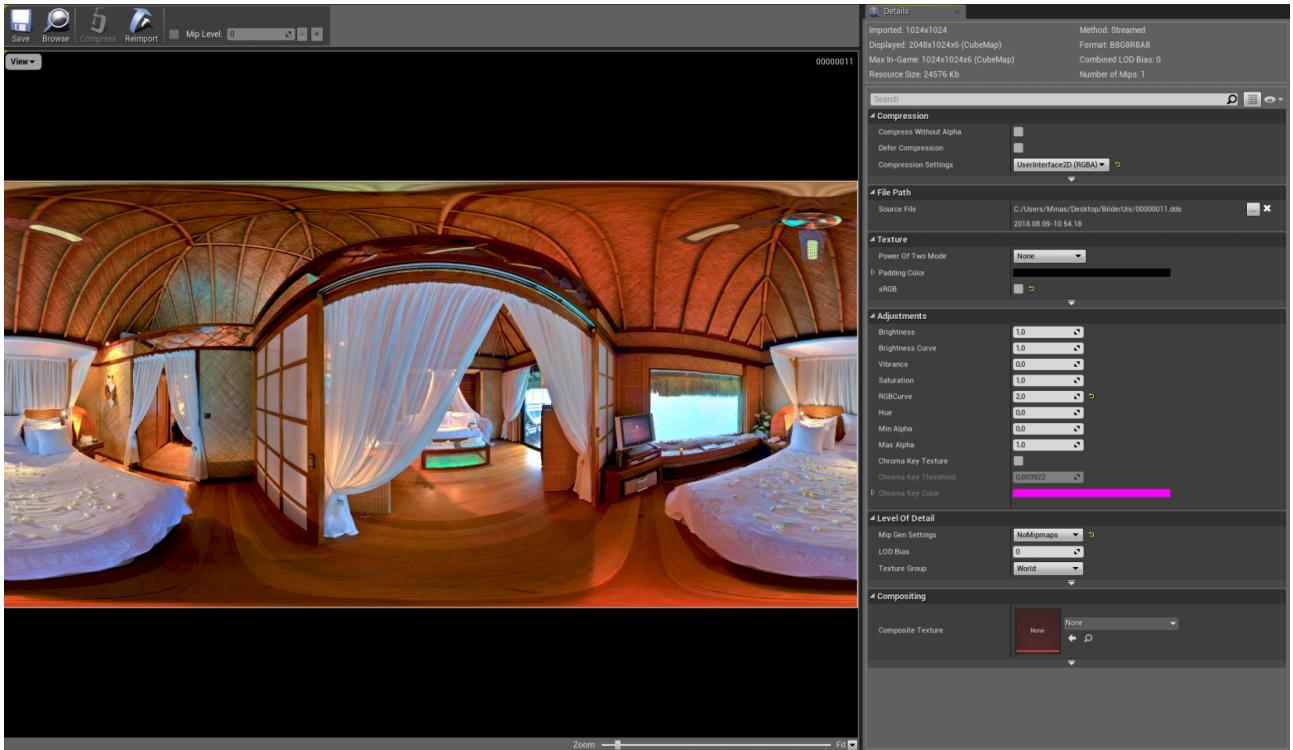


Abbildung 42: Optimale Einstellungen des Bildes in Unreal.

Diese Einstellungen lassen sich für alle Bilder übernehmen. Lediglich die Einstellung RGBCurve unter “Adjustments” muss von Bild zu Bild variiert werden. Desto höher der eingetragene Wert, umso kräftiger werden die Farben des Bildes. Wird der Standardwert von “1,0” gelassen, wirkt das Bild sehr blass und ist damit nicht anschaulich. Hier spielt auch die Qualität des Bildes eine große Rolle. Am Anfang des Kapitels wurde HDR erwähnt. HDR steht für High Dynamic Range und ermöglicht einen größeren Helligkeitsbereich als SDR (Standard Dynamic Range). Es lässt das Bild realistischer wirken, ohne Farbtöne im dunklen oder hellen Bereichen zu vernachlässigen[37]. Abbildung 43 zeigt den Qualitätsunterschied zwischen SDR und HDR. Es ist von Vorteil ein solches Format zu verwenden, um eine gute VR-Umgebung zu realisieren.

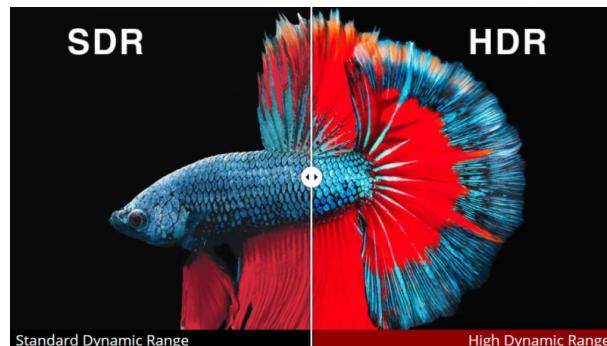


Abbildung 43: SDR vs. HDR[38].

Nach dem das Bild in Unreal fertig konfiguriert wurde, muss eine "Blueprint Class" angelegt werden. Diese wird in diesem Beispiel "SkySphere-BP" genannt. In diesem Blueprint wird eine Variable angelegt mit dem Namen "SkyMaterial". Außerdem sollte das Construction Script wie in Abbildung 44 aufgebaut sein.

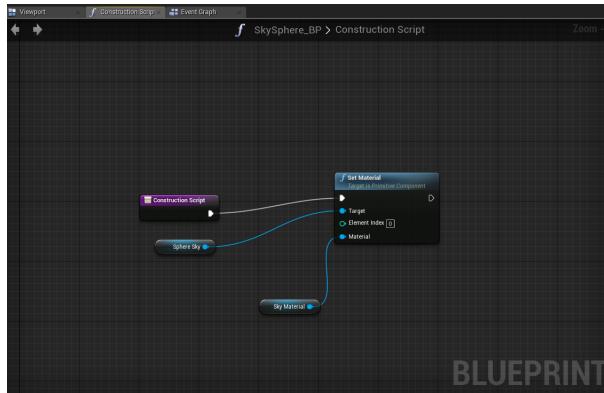


Abbildung 44: SkySphere-BP Construction Script.

Mit diesem Construction Script lassen sich CubeMaps in die SkySphere importieren und bei Bedarf auch austauschen.

Nun muss ein Material für die SkySphere erstellt werden. Abbildung 45 zeigt wie diese Konfiguriert werden sollte.

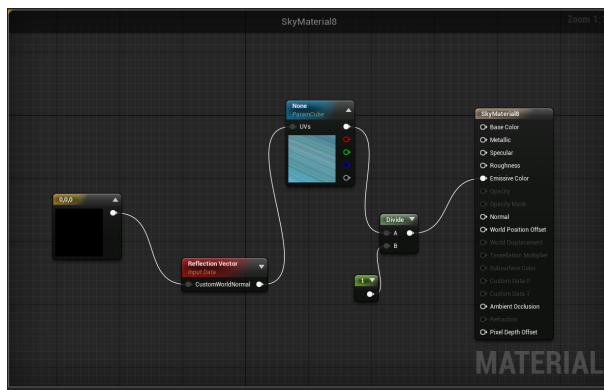


Abbildung 45: Konfigurationen von Material.

Im Knoten "ParamCube" muss die Textur ausgesucht werden. In diesem Fall handelt es sich um die erstellte CubeMap. Als letztes muss das erzeugte Material der SkySphere zugewiesen werden. Nun lässt sich die Umgebung in Unreal anzeigen. Der Text im Bild lässt sich mit einem Blueprint-Pawn erzeugen. Die Position des Textes wird manuell vorgenommen, in dem der Text im Bild verschoben wird. Damit der Übergang zwischen den Bildern angenehm für die Probanden wirken soll, wird ein "fade in" und ein "fade out" für jedes einzelne Bild erzeugt. Hierfür muss in Unreal unter Cinematics eine Matinee hinzugefügt werden. In dieser Matinee wird das Fade erzeugt, in dem die Anzeigedauer angegeben wird und vier Keys

hinzugefügt werden. Der erste Key sagt dem Fade bei welcher Sekunde das Bild anfängt, der zweite Key bis wann das Bild seine maximale Helligkeit erreichen soll, der dritte Key ab wann das Bild wieder dunkler werden soll und der vierte Key bis wann das Bild komplett verschwunden sein soll. Abbildung 46 zeigt das Konfigurationsfenster der Matinee.

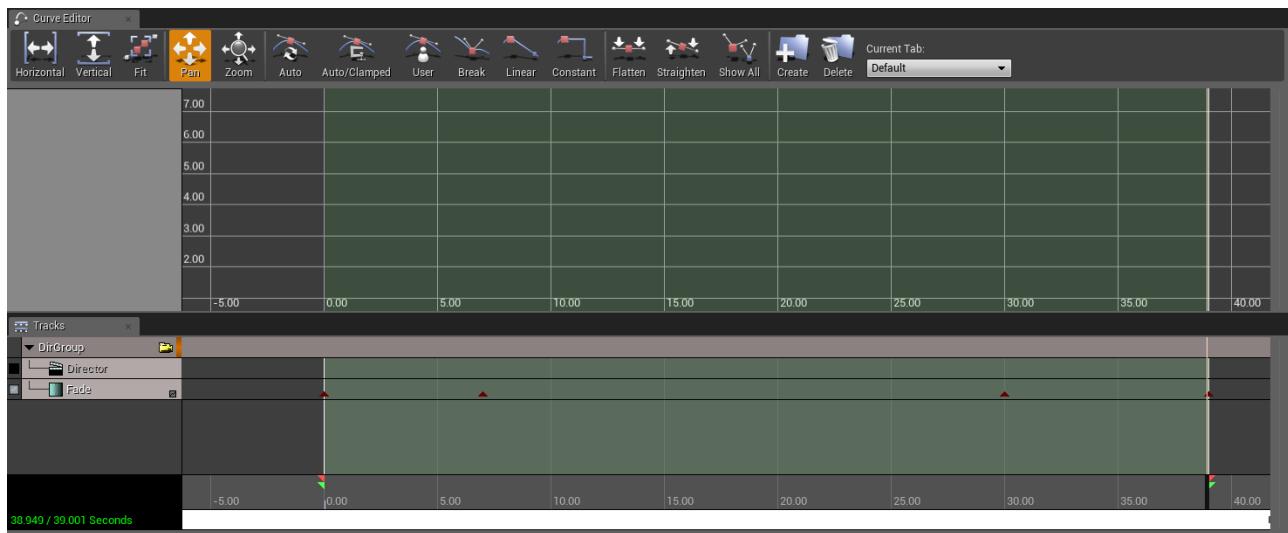


Abbildung 46: Konfiguration der Matinee; rote Dreiecke = Keys; grüner Bereich = Anzeigedauer.

Somit ist das erste Bild für das Szenario fertig konfiguriert. Dieser Ablauf wurde für acht weitere Bilder durchgeführt.

Der Wechsel zwischen den Bildern für das Hauptszenario wurde anhand der Level-Option von Unreal vorgenommen. Diese Option besteht aus einem anhaltenden Level und beliebig viele Level, die hinzugefügt werden können. In diesem Fall acht Level für die verwendeten Bilder. Das anhaltende Level beinhaltet die Audio-Datei und verwaltet den Wechsel der Bilder, welches alle 39 Sekunden durchgeführt wird. Die 39 Sekunden kamen zustande, in dem die Länge der Audio-Datei durch die Anzahl der Bilder geteilt wurde. Damit wurde vermieden das die Audio-Datei von Anfang an abgespielt wird. Ein großer Vorteil der Level-Option ist, dass die Audio-Datei fortläuft, auch während dem Level-Wechsel und somit keine Unterbrechungen entstehen. Da das WarmUp für das Glücks-Szenario nur aus einem Bild besteht, musste hier keine Level-Option konfiguriert werden.

### 15.3.2 Langeweile

Verantwortlich: Boris  
- RfP

In diesem Prototyp wird nur das "Peg-Turning" Spiel als Szenario benutzt, da man

die Zeit des Experiments reduzieren wollte. Allerdings gib es ein kleines Unterschied mit dem Spiel von dem zweiten Prototyp. Hier wird die grüne kreisförmige Scheibe durch den Proband selbst in Bewegung gebracht. Nach jeder Bewegung des Kreis muss man mindesten fünften Sekunde warten bis die nächste Bewegung möglich ist. Um dies Szenario in Unreal Engine zu verwirklichen habe ich erstmal ein Widget Blueprint erstellt. Dann habe ich vier mal das Bild von dem Spiel genau an der gleiche Stelle und in der gleiche Größe in der Widget hinzugefügt. Das erste Bild ist in der Ausgangsposition. Dann wird jedes der nächsten Bilder eine Drehung des vorherigen Bildes um neunzig Grad sein. Es wird von Anfang an das Ausgangsbild auf sichtbar gesetzt und die anderen auf unsichtbar. Danach habe ich einen transparenten Button immer auf die Bilder der Scheiben eingefügt, um das Wechsel von Bilder zu steuern. Zur Programmierung erstelle ich eine Funktion mit Timeout und Branch-Bedingung und der Algorithmus wird so rekursiv definiert: Wenn das Button gedrückt wird, wird das nächste Bild auf sichtbar ("visible") gesetzt und die anderen Bilder auf unsichtbar ("invisible"), dann wird ein Timeout gesetzt und eine Branch-Bedingung soll Prüfen ob das Timeout durch ist. Sollte man vom Ende des Timeouts der Button drücken, so sollte nichts passieren. Wenn das Timeout fertig ist, soll das Szenario sich wiederholen. Es wird auch jede zeit durch einen Text in einem Textfeld gezeigt, wenn man die Scheibe drehen kann und wenn man warten muss. Ein letztes Timeout wird hinzugefügt, um zu prüfen, dass das Szenario ein gewünschtes Zeit dauert.

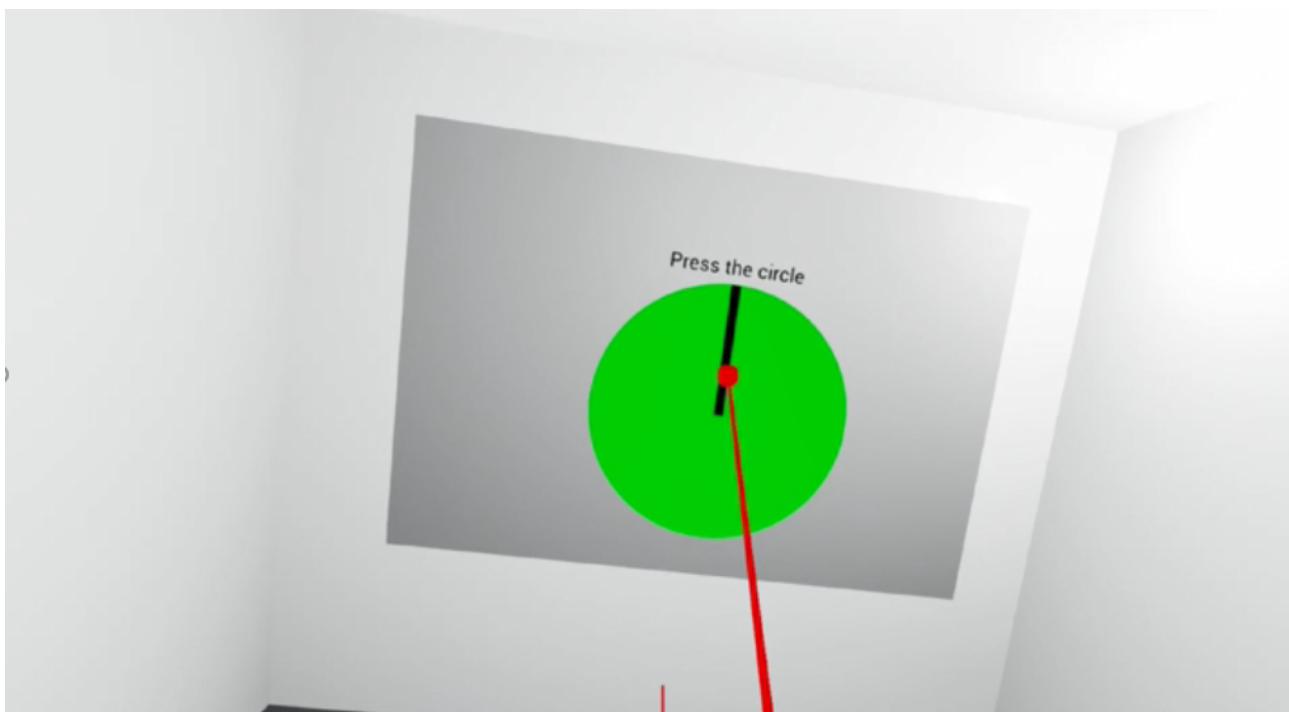


Abbildung 47: Bild des Langeweile-Szenarios in VR.

### 15.3.3 Frustration

Verantwortlich: Meryem  
- add technical aspects? Jonas?

Wie bereits in Abschnitt 7.3.3 dargeboten, stellt die Frustration einen negativen Zustand des Menschen dar, welcher durch Misserfolgserlebnisse sowie durch Versagungs- und Enttäuschungserlebnisse einhergehen kann. Die Frustration stellt einen negativen Zustand des Menschen dar, welcher mehrere Indikatoren haben kann. Dieser Zustand kann sowohl eine Gefühlslage als auch eine Folge vorhergehender Emotionen sein.

Für das Frustrationsexperiment der Studie soll erneut auf das Auslösen der Misserfolgserlebnisse sowie eine empfundene Ungerechtigkeit zurückgegriffen werden.

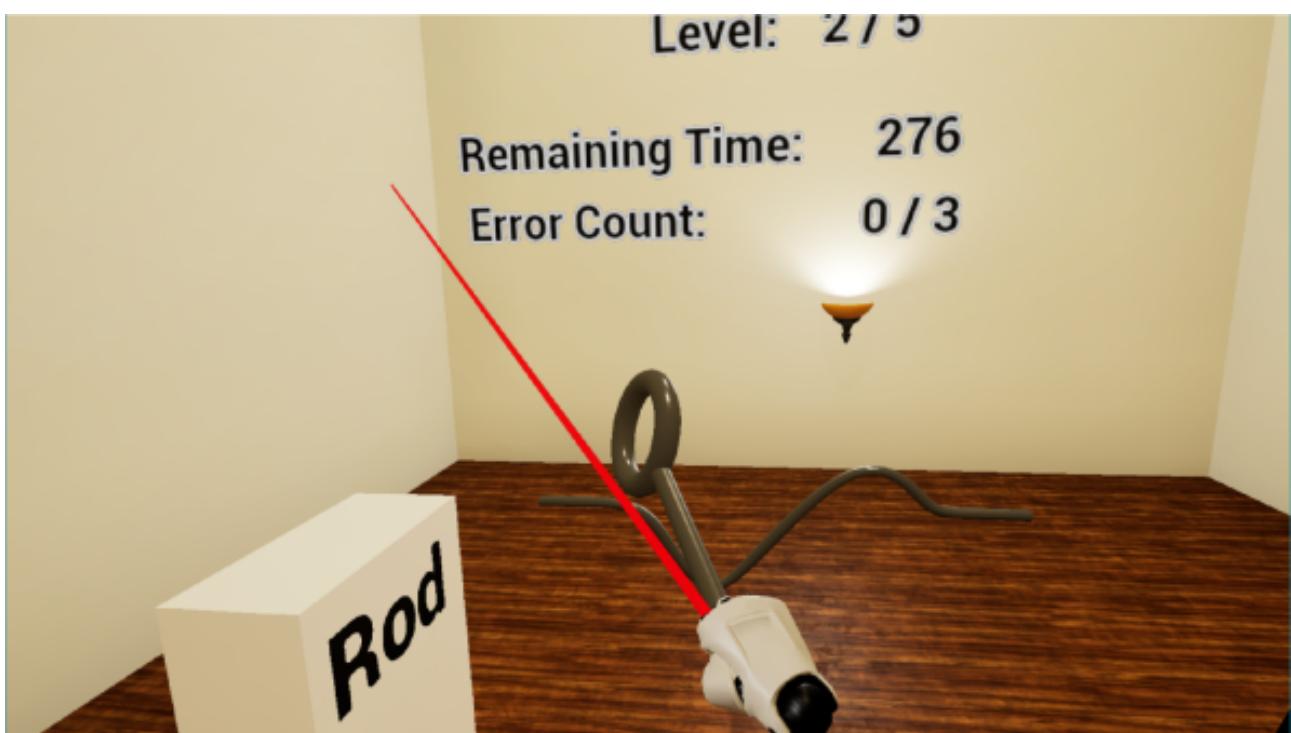


Abbildung 48: Bild des Frustrations-Szenarios in VR.

Dem Probanden wird die Aufgabe gestellt, in eine VR-Umgebung das Spiel "heißer Draht" zu spielen. In diesem Spiel besteht die Aufgabe darin, den Ring, welcher sich in der Hand des Spieler befindet, von dem Startpunkt zum Endpunkt eines Drahts zu befördern. Der Ring darf währenddessen den Draht nicht berühren. Dieses Spiel ist durch die geschwungene Form des Drahts schwierig und fordert daher viel Ruhe und Geschick. Dem Probanden wird eine Zeit vorgegeben, in welcher dieser von dem Startpunkt zum Endpunkt gelangen muss. Der Faktor des Zeitdrucks kann eine Stressreaktion auslösen. Da das Spiel jedoch den Ehrgeiz erwecken kann, kann es dazu führen, dass es nicht zu der erhofften Frustration kommt. Um diese Emotion

trotzdem einleiten zu können, wird das Spiel an einigen Stellen so vorbereitet, dass der Proband durch Spielfehler den Draht berührt und erneut von dem Startpunkt aus starten muss. Die empfundene Ungerechtigkeit und Ohnmächtigkeit des Probanden soll als Indikator der Frustration dienen.

## 16 Messreihe

Verantwortlich: Kevin, Artur

## 17 Mustererkennung

Verantwortlich: Artur  
- RfP

Mit dem relativ großen Datensatz von 88 Testpersonen, wurde sich entschieden die Mustererkennung vor allem auf DNN-Ansätze mit Verwendung von den Circumplex-Labels zu fokusieren. Die Gründe dafür waren, dass sich durch dieses Vorgehen die besten Ergebnisse erhofft wurden und ein gewisser Zeitdruck durch das anstehende Projektende. Die getesteten DNN-Ansätze wurden bereits ausführlich in Kapitel 9.4.3 erklärt. Weitere Informationen zu den verwendeten Circumplex-Lables können in Kapitel 15.2 nachgeschlagen werden.

## 18 Ergebnisse

Verantwortlich: Artur

Wegen anderer Studien, wurden jeweils DNNs für ein zwei-Klassen- und ein drei-Klassen-Mustererkennungsproblem getestet. Zudem wurde zum einem Probanden-abhängige und zum anderen Probanden-unabhängige Datensätze verwendet, um z.B. Overfitting feststellen zu können. Im Folgendem sind die F1-Scores für Erregung und Wertigkeit (entsprechend dem Circumplex-Modell) vorgestellt, wobei MLP die besten Ergebnisse erzielten.

Erregung F1-Score (%)	Probanden-abhängig	Probanden-unabhängig
2 Klassen	90,57%	58,35%
3 Klassen	83,08%	42,65%

Tabelle 7: F1-Score der DNNs für Erregung.

Wertigkeit F1-Score (%)	Probanden-abhängig	Probanden-unabhängig
2 Klassen	88,21%	44,07%
3 Klassen	84,72%	21,86%

Tabelle 8: F1-Score der DNNs für Wertigkeit.

Es ist klar zu erkennen, dass die Probanden-anhängige Ergebnisse relativ gut sind und die Probanden-unabhängige immer sanken, d.h. relativ schlecht sind. Wie bei dem ersten Datensatz, findet hier wahrscheinlich auch Overfitting bei den Probanden-anhängigen Datensatz statt. Auch hier wurden ähnliche t-SNE Plots wie in Kapitel 10.4 erstellt, die diese Vermutung bestätigen. Bei dem Probanden-unabhängigen Datensatz liefern DNNs keine gute Ergebnisse, wobei die Performance für Erregung klar besser ist als für Wertigkeit.

# 19 Alternative Lösungen

Verantwortlich: Artur

Da davon ausgegangen wird, dass weder Codebook Approach noch die handgefertigten Merkmale die ernüchternden Ergebnisse der DNN übertreffen können, wurde stattdessen nach alternativen Lösungsmöglichkeiten gesucht. Im folgenden werden die beiden vielversprechendsten Optionen vorgestellt.

## 19.1 Kalibrierung

Verantwortlich: Jonas

## 19.2 Plan B

Verantwortlich: Meryem

---

# 20 Zusammenfassung und Ausblick

Verantwortlich: Meryem

In diesem Kapitel wird die Zusammenfassung der Arbeit sowie das resultierende Fazit und schließlich ein Ausblick dargeboten.

## 20.1 Zusammenfassung

Verantwortlich: Arnaud

## 20.2 Ausblick

Verantwortlich: Boris  
- RfP

Insgesamt wird diese Arbeit im Rahmen von ELISE "umgesetzt". Im Rahmen der Hardware sollte erst mal weitere Minimierung von der Leiterplatte gemacht werden, was dazu beitragen würde, das Rauschen der analogen Sensoren (EEG/EOG/GSR) zu minimieren. Da dieser Punkt wahrscheinlich auch die Verwendung kleinerer Komponenten erfordern würde, wenn möglich (Widerstände etc.), was wiederum sehr schwierig von Hand zu sodern wäre, müsste also Komponenten vom Leiterplattenhersteller montieren werden. Danach sollte das 6-Kanal-ADS1299-6 durch das 8-Kanal-ADS1299, wegen der Einbeziehung der GSR-Messungen in den TI-Chip die derzeit keine freien Kanäle mehr ermöglicht, ersetzt werden. Dann sollte es der automatische Reset für den  $\mu$ C (und das gesamte System) aktiviert werden und es sollte auch einen besseren Platz für den Reset-Knopf zu finden oder ihn durch einen abgewinkelten Schalter (wie den Programmierschalter) zu ersetzen, da die aktuelle Stelle diesen Knopf schlecht ist. Zudem konnte die Kommunikation zwischen PCB und Software überarbeitet werden. Es könnte zum Beispiel anderes Protokoll, Bluetooth etc verwendet werden. Diese könnte zu einer Erhöhung der Sampling-Rate führen, die im Moment bei 250 Samples/Sekunde liegt. Letztendlich könnte auch eine gute Idee sein, die Gesamtgröße unserer Maske zu verkleinern, da sie für einige Probanden etwas zu groß war (z.B. um unter dem Auge manchmal die Nase zu berühren usw.), sowie das Loch für den Temperatursensor, die vergrößert werden sollte, damit man sich dort nicht durchschmelzen muss. Als Erweiterung bei der Software, könnte die VR-Emotionsinduktionen verbessert werden. Es könnte auch andere Arten von Induktionswerkzeugen wie zum Beispiel ein Gerät die mehr Mobilität erlaubt verwendet werden, damit man bessere Ergebnis bei der Messungen bekommt. Die allgemeine Gestaltung von dem ganzen Prozess sollte auch noch mal bearbeitet werden. Bei der Mustererkennung sollte andere Modelle getestet werden. Diese könnte das Endergebnis verstärken. Auch andere Klassifikatoren oder Ler-

---

nansätze (halbüberwachte und verstärkte Lernsätze) konnte auch für die nächste Arbeiten verwendet werden. Bei Plan B konnte es versucht werden, die Top-Methode zu implementieren und zu testen, um mehr Gewicht an diese Alternativelösung zu geben.

# Abbildungsverzeichnis

1	Große Übersicht des Gesamtprojekts[1]. . . . .	7
2	Einordnung von Emotion nach Plutchik. . . . .	15
3	Einordnung von emotionale Erfahrungsprozess nach Wundt. . . . .	16
4	Einordnung von emotional Erfahrung nach Russell. . . . .	17
5	HTC VIVE mit VR-Brille, Controllern und Base. . . . .	18
6	VR-Brille . . . . .	19
7	VR-Basisstation . . . . .	19
8	VR-Controller . . . . .	19
9	Beispiel eines Blueprintcodes. . . . .	21
10	Emotion Recognition Chain . . . . .	24
11	Aufteilen eines Datensets in ein Trainings- und Testset . . . . .	26
12	Beispiel eines SVM-Klassifikators im 2D Merkmalsraum . . . . .	27
13	Beispiel für die Verwendung des Kernel-Tricks . . . . .	28
14	Beispiele für einen hard- und soft-margin SVM . . . . .	29
15	Bild des verwendeten Fragebogens. . . . .	33
16	Vignetten für das Glücks-Szenario. . . . .	35
17	Bild des Langeweile-Szenarios. . . . .	36
18	Bild des Frustabit Spieles für das Frustration-Szenario. . . . .	37
19	Schiebefenster-Segmentierung . . . . .	40
20	Merkalsextraktion aus frequenzbezogener Domain . . . . .	43
21	Spitzenzähler-Merkmal . . . . .	44
22	Codebuchkonstruktion . . . . .	45
23	Codewortzuweisung . . . . .	46
24	Architektur eines MLP Network. . . . .	48
25	Architektur eines CNN Network. . . . .	48
26	Eigene Architektur eines Multimodal-MLP-Network. . . . .	49
27	Architektur eines LSTM-Networks. . . . .	49
28	Architektur eines CNN/LSTM-Hybriden. . . . .	50
29	Leave-One-Subjekt-Out-Cross-Validation (LOSOCV) . . . . .	52
30	Gitter-Suche . . . . .	53
31	t-SNE Plots der handgefertigten Merkmale. . . . .	56
32	Sensoraufzeichnungen von Daten . . . . .	57
33	Nahaufnahme von Rauschen in Daten . . . . .	58
34	Bild des Fragebogen-Teils, entsprechend dem circumplex-Modell. . . . .	65
35	Bild des Fragebogen-Teils, wo die dominierenden Emotionen abgefragt werden. . . . .	66
36	Im Glück-Szenario verwendete Bilder und deren Texte . . . . .	67
37	Ein im Projekt verwendetes HDR-Panorama-Bild. . . . .	68

38	Ausgabe der Blender-File und deren Rotation. . . . .	69
39	Rotation und Reihenfolge für eine CubeMap . . . . .	69
40	Rotation und Reihenfolge für die CubeMap der Abbildung 38. . . . .	70
41	Aufgeklappte CubeMap . . . . .	70
42	Optimale Einstellungen des Bildes in Unreal. . . . .	71
43	SDR vs. HDR . . . . .	71
44	SkySphere-BP Construction Script. . . . .	72
45	Konfigurationen von Material. . . . .	72
46	Konfiguration der Matinee . . . . .	73
47	Bild des Langeweile-Szenarios in VR. . . . .	74
48	Bild des Frustrations-Szenarios in VR. . . . .	75

# Tabellenverzeichnis

1	Vergleich einige Basisemotions-Theorien . . . . .	14
2	Einige Basisemotionen jeweils mit Verhalten und Funktion . . . . .	15
3	Statistische Merkmale . . . . .	41
4	Durchschnittlicher F1-Wert . . . . .	54
5	Confusion Matrix . . . . .	54
6	Vergleich der Ergebnisse der unterschiedlichen Merkmalsextraktionen.	55
7	F1-Score der DNNs für Erregung. . . . .	78
8	F1-Score der DNNs für Wertigkeit. . . . .	78

# Abkürzungen

<b>ADADELTA</b>	An Adaptive Learning Rate Method
<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>CNN</b>	Convolutional Neural Network
<b>CRID</b>	Center for Responsible Innovation & Design
<b>C-SVM</b>	Soft-margin Support-Vector-Machine
<b>CSV</b>	Comma-Separated-Values
<b>DDS</b>	Direct Draw Surface
<b>DNN</b>	Deep Neural Networks
<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>LOSOCV</b>	Leave-One-Subject-Out-Cross-Validation
<b>LSTM</b>	Long Short-Term Memory Network
<b>MLP</b>	Multilayer Preceptron
<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

# Referenzen

- [1] D. Krönert. *Entwurf eines kompakten mikrocontrollergestützten Systems zur Emotionserkennung in einer Virtual-Reality-Umgebung*. 2017.
- [2] B. Fehr and J. A. Russel. Concept of emotion viewed from a prototype perspective. *Journal of Experimental Psychology: General*, volume 113(issue 3):pages 464–486, 1984.
- [3] J. Plamper. *Geschichte und Gefühl: Grundlagen der Emotionsgeschichte*. Siedler, November 2012.
- [4] E. Geslin. *Processus d'induction d'émotions dans les environnements virtuels et le jeu vidéo*. Psychologie, 2013.
- [5] C. Darwin. *Der Ausdruck der Gemütsbewegungen bei dem Menschen und den Tieren*. November 1872.
- [6] W. James. What Is an Emotion? [https://www.scirp.org/\(S\(351jmbntvnsjt1aadkposzje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1331721](https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1331721), letzter Zugriff am 8. März 2019.
- [7] W. B. Cannon. The Wisdom of the Body. [https://www.scirp.org/\(S\(351jmbntvnsjt1aadkposzje\)\)/reference/ReferencesPapers.aspx?ReferenceID=1533450](https://www.scirp.org/(S(351jmbntvnsjt1aadkposzje))/reference/ReferencesPapers.aspx?ReferenceID=1533450), letzter Zugriff am 8. März 2019.
- [8] S. Schachter. The psychology of affiliation. <https://journals.sagepub.com/doi/abs/10.1177/002076406100700316>, letzter Zugriff am 8. März 2019.
- [9] T. Goschke. Emotionspsychologie I. [https://tu-dresden.de/mn/psychologie/allgpsy/ressourcen/dateien/lehre/lehreveranstaltungen/goschke\\_lehre/ws\\_2013/vl\\_motivation/VL-Emotion-1.pdf?lang=de](https://tu-dresden.de/mn/psychologie/allgpsy/ressourcen/dateien/lehre/lehreveranstaltungen/goschke_lehre/ws_2013/vl_motivation/VL-Emotion-1.pdf?lang=de), accessed on 19 November 2017.
- [10] A. Kolb. Virtual Reality Introduction. [http://www.cg.informatik.uni-siegen.de/sites/www.cg.informatik.uni-siegen.de/files/Dateianhaenge/vr-part01-script\\_3.pdf](http://www.cg.informatik.uni-siegen.de/sites/www.cg.informatik.uni-siegen.de/files/Dateianhaenge/vr-part01-script_3.pdf), letzter Zugriff am 8. März 2019.
- [11] HTC Corporation. VIVE VR-SYSTEM. <https://www.vive.com/de/product/>, letzter Zugriff am 9. März 2019.
- [12] X. Zhu. Semi-supervised learning literature survey, July 2008.
- [13] C. Cortes and V. Vapnik. *Support-Vector Networks*, volume volume 20. Kluwer Academic Publishers, Boston, September 1995.

- [14] E. Kim. Everything you wanted to know about the kernel trick. September 2013.
- [15] R. H. Gault. A history of the questionnaire method of research in psychology. <https://www.tandfonline.com/doi/abs/10.1080/08919402.1907.10532551>, accessed on 30 January 2019.
- [16] G. Bien. Über das Glück. <http://www.joachimschummer.net/books/glueck-und-ethik/bien.pdf>, letzter Zugriff am 5. März 2019.
- [17] S. J. Vodanovich. Psychometric measures of boredom: A review of the literature. *Journal of Psychology*, volume 137(issue 6):pages 569–595, November 2003.
- [18] J. Grus. *Data Science from Scratch*. O'Reilly Media, April 2015.
- [19] H.P. Martinez, Y. Bengio, and G.N. Yannakakis. Learning deep physiological models of affect. *IEEE Computational Intelligence Magazine*, volume 8(issue 2):pages 20–33, April 2013.
- [20] A. Piet. *Bachelor-thesis: Emotion recognition using 1D physiological signals following a supervised learning approach*. December 2017.
- [21] J. Littau. *Bachelor-thesis: Analysis of fourier features for emotion recognition using physiological signals*. June 2018.
- [22] P. Gouverneur. *Bachelor-thesis: Classification of physiological data for emotion recognition*. September 2016.
- [23] K. Shirahama, L. Koeping, and M. Grzegorzek. Codebook approach for sensor-based human activity recognition. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 197–200, September 2016.
- [24] J.C. van Gemert, C.J. Veenman, and A.W.M. Smeulders. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32(issue 7):pages 1271–1283, June 2009.
- [25] K.P. Murphy. *Machine Learning and A Probabilistic Perspective*. The MIT Press, London, September 2012.
- [26] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. Early versus late fusion in semantic video analysis. *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402, November 2005.
- [27] K. Schnieber. *Bachelor-thesis: Processing and analysis of sensor data in the context of emotion recognition*. 2018.

- [28] M. D. Zeiler. *ADADELTA: An adaptive learning rate method*. Decemer 2012.
- [29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. *ICML'10 Proceedings of the 27th International Conference on Machine Learning*, pages pages 807–814, June 2012.
- [30] S. Hochreiter and J. Schmidhuber. Long shirt-term memory. *Neural Computation*, volume 9(issue 8):pages 1735–1780, November 1997.
- [31] J. Russel. A circumplex model of affect. *Journal of Personality and Social Psychology*, volume 39(issue 6):pages 1161–1178, 1980.
- [32] J. Xiao and K. A. Ehinger and A. Oliva and A. Torralba. Recognizing Scene Viewpoint using Panoramic Place Representation. <https://people.csail.mit.edu/jxiao/SUN360/>, letzter Zugriff am 12. März 2019.
- [33] Coldacid. Adding custom cube backgrounds. <https://steamcommunity.com/sharedfiles/filedetails/?id=589533498>, letzter Zugriff am 12. März 2019.
- [34] V. Oberst. .dds Dateiendung. <https://www.dateiendung.com/format/dds>, letzter Zugriff am 12. März 2019.
- [35] M. Franczak. Creating custom sky from HDR image in Unreal Engine. <https://evermotion.org/tutorials/show/10738/creating-custom-sky-from-hdr-image-in-unreal-engine>, letzter Zugriff am 12. März 2019.
- [36] M. Belanec. GLSL cube mapping. <http://www.3dcpptutorials.sk/index.php?id=24>, letzter Zugriff am 12. März 2019.
- [37] M. Kontani and Y. Jitsumori. HDR im Detail – was ist HDR? <https://www.eizo.de/praxiswissen/monitorwissen/hdr-im-detail-was-ist-hdr/>, letzter Zugriff am 12. März 2019.
- [38] G. Finch. What is HDR? HDR vs. SDR compared. <https://www.viewsonic.com/library/photography/what-is-hdr-hdr-vs.-sdr>, letzter Zugriff am 12. März 2019.

# **Anhang**