



# Erfassung biometrischer Daten mithilfe von Smartphones zur Emotionsbestimmung

Torben Brenner und Lukas Seemann

28.11.2017 - 28.05.2018

# Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Listings	V
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	2
1.3 Vorgehensweise . . . . .	2
<b>2 Theoretische Grundlagen</b>	<b>3</b>
2.1 Biometrie und biometrische Merkmale . . . . .	3
2.2 Emotionen . . . . .	4
2.2.1 Definition . . . . .	4
2.2.2 Eingrenzung der zu messenden Emotionen . . . . .	5
2.2.3 Wie lassen sich Emotionen messen? . . . . .	6
2.3 Umgang mit biometrischen Daten . . . . .	7
2.3.1 Indiz . . . . .	7
2.3.2 Kausalität . . . . .	8
2.4 Emotionsindizes . . . . .	8
2.4.1 Puls . . . . .	9
2.4.2 Hautleitfähigkeit/Hautwiderstand . . . . .	9
2.4.3 Mimik . . . . .	11
2.4.4 Tippverhalten . . . . .	12
2.4.5 Gangart . . . . .	13
2.4.6 Stimme . . . . .	14
2.5 Nützliche Hardware . . . . .	15
2.5.1 Smartphones . . . . .	15
2.5.2 Smartphone Sensoren . . . . .	16
2.5.3 Externe Sensoren . . . . .	16
2.6 Möglichkeiten der Erfassung von biometrischen Daten . . . . .	18
2.6.1 GSR-/EDA-Sensoren . . . . .	18
2.6.2 Stimmerkennung mit Smartphone-Mikrofon . . . . .	19
2.6.3 Gesichtserkennung mit Smartphone-Kamera . . . . .	20
2.6.4 Pulsmessung mit Smartphone-Kamera . . . . .	22
2.6.5 Pulsmessung mit externen Sensoren . . . . .	24
2.6.6 Analyse des Tippverhaltens bei Smartphone-Nutzung . . . . .	25
2.6.7 Gangerkennung mit dem Smartphone . . . . .	25

<b>3</b>	<b>Konzept</b>	<b>27</b>
3.1	Priorisierung der Erfassungsmöglichkeiten . . . . .	27
3.2	Übertragung biometrischer Daten in Indikatorscores . . . . .	29
3.3	Auswertung der Indikatorscores zur Emotionsbestimmung . . . . .	30
3.3.1	Kausalitätsregeln . . . . .	30
3.3.2	Unterteilung der Entscheidung in Intervalle . . . . .	31
3.4	Ionic Framework . . . . .	31
3.4.1	Aufbau und Einsatz des Frameworks . . . . .	31
3.4.2	Gründe für die Verwendung . . . . .	32
3.5	Architektur der mobilen Applikation . . . . .	32
3.5.1	Backendlogik . . . . .	33
3.5.2	Frontend . . . . .	35
3.5.3	Vor- und Nachteile der Architektur . . . . .	36
3.6	MockUps . . . . .	37
3.6.1	Hautleitfähigkeit . . . . .	38
3.6.2	Tippverhalten . . . . .	39
3.6.3	Gesichtserkennung . . . . .	40
3.6.4	Auswertung . . . . .	41
<b>4</b>	<b>Umsetzung des Prototypen emoTrix</b>	<b>42</b>
4.1	Umsetzung der Backendlogik . . . . .	42
4.1.1	Sensor-Klasse . . . . .	42
4.1.2	CausalityRule-Klasse . . . . .	43
4.1.3	Decider-Klasse . . . . .	44
4.2	Datenerfassung in Form von Emotionstests . . . . .	46
4.2.1	GSR-Test . . . . .	46
4.2.2	Face-Test . . . . .	52
4.3	Beschreibung der implementierten Kausalitätsregeln . . . . .	56
4.3.1	Regeln des GSR-Test . . . . .	56
4.3.2	Regeln des Face-Test . . . . .	57
4.4	Benutzeroberfläche der App . . . . .	58
4.4.1	HomePage . . . . .	58
4.4.2	GSRPage . . . . .	59
4.4.3	CameraPage . . . . .	60
4.4.4	DecisionPage . . . . .	61
<b>5</b>	<b>Schluss</b>	<b>62</b>
5.1	Anwendungsszenarien . . . . .	62
5.2	Fazit . . . . .	62
5.3	Ausblick . . . . .	62
	<b>Literaturverzeichnis</b>	<b>63</b>
	<b>Anhänge</b>	<b>70</b>

## Abkürzungsverzeichnis

<b>DAG</b>	Directed acyclic graph
<b>EDA</b>	Elektrodermale Aktivität
<b>EVM</b>	Eulersche Videoverstärkung
<b>GHz</b>	Gigahertz
<b>GSR</b>	Galvanic Skin Response

# Abbildungsverzeichnis

1	Relationships between organismic subsystems and the functions and components of emotion - Klaus R. Scherer . . . . .	5
2	Rad der Emotionen - Robert Plutchick . . . . .	6
3	Prototype version of the Geneva Emotion Wheel - Klaus R. Scherer . . .	7
4	Fiktives Beispiel eines DAGs . . . . .	8
5	Reaktion des Körpers auf Stress und Entspannung . . . . .	10
6	Event Study Compared with Studies of Facial Expression of Motion(Values in Percentages) . . . . .	14
7	Sensor als Schnittstelle zwischen nicht-elektrischen und elektrischen Raum	17
8	Positionsmöglichkeiten der EDA-Messung . . . . .	19
9	GSR-Sensor mit Finger-Elektroden . . . . .	20
10	Eulersche Videoverstärkung zur Pulsbestimmung . . . . .	23
11	Puls-Sensor für einen Arduino . . . . .	24
12	Ablauf der Erstellung von Indikatorscores . . . . .	29
13	Ablauf der Ausführung von Kausalitätsregeln . . . . .	30
14	Ionic Framework - Logo . . . . .	31
15	Architektur der App . . . . .	33
16	Geplantes Frontend der App . . . . .	35
17	Mockup - Homescreen(links) und Menü(rechts) . . . . .	37
18	Mockup - Arduino verbinden(links nicht verbunden, rechts verbunden) .	38
19	Mockup - Hautleitfähigkeit Test starten(links) und Test Ergebnis(rechts)	39
20	Mockup - Tippverhalten Test starten(links) und Test ergebnis(rechts) . .	40
21	Mockup - Gesichtserkennung vor Senden des Bilds(links) und Testergebnis(rechts) . . . . .	40
22	Mockup - Auswertung . . . . .	41
23	Arduino UNO R3 (links) und Grove Base Shield . . . . .	47
24	HC-05-Bluetooth-Modul für Arduino . . . . .	48
25	GUI der HomePage . . . . .	58
26	GUI der HomePage . . . . .	59
27	GUI der HomePage . . . . .	61
28	GUI der HomePage . . . . .	62

# Tabellenverzeichnis

1	Priorisierung der Erfassungsmöglichkeiten . . . . .	27
---	---	----

## Listings

1	abstrakte Klasse Sensor . . . . .	42
2	Typ condition . . . . .	43
3	execute Funktion der Klasse CausalityRule . . . . .	44
4	Attribute der Decider-Klasse . . . . .	44
5	decide-Funktion der Decider-Klasse . . . . .	45
6	assumeEmotion-Funktion der Decider-Klasse . . . . .	45
7	executeCausalityRules-Funktion der Decider-Klasse . . . . .	46
8	Quellcode des Arduinos . . . . .	48
9	JS Code . . . . .	50
10	JS Code . . . . .	51
11	Input und Output der Kamera Komponente . . . . .	52
12	Einbindung Kamera Komponente . . . . .	53
13	Aufbau Anfrage an Gesichtserkennungsschnittstelle . . . . .	53
14	Callback Methode auf FaceSensor Observable . . . . .	54
15	Ergebnis Gesichtserkennungsschnittstelle, Beispiel Microsoft . . . . .	55
16	Bedingung der ersten Kausalitätsregel(gestresst) . . . . .	56
17	Kausalitätsregel basierend auf Face-Test für Emotion Wut . . . . .	57

# 1 Einleitung

In diesem Kapitel wird zunächst die Motivation für die Studienarbeit beschrieben. Im Anschluss daran werden die Ziele der Arbeit definiert, um schließlich die Vorgehensweise zur Erreichung dieser Ziele aufzustellen.

## 1.1 Motivation

LUKAS SEEMANN

Smartphones sind aus dem Alltag vieler Menschen nicht mehr wegzudenken. Nach Prognosen der Statista GmbH nutzen im Jahr 2018 57 Millionen Menschen in Deutschland ein Smartphone.<sup>1</sup> Weltweit betrachtet vergrößert sich die Nutzerzahl für 2018 auf ungefähr 2,53 Milliarden Personen.<sup>2</sup> Hierbei muss der Unterschied zwischen Smartphones und normalen Mobiltelefonen, die als Hauptfunktionalität das Telefonieren besitzen, hervorgehoben werden. Die 2,53 Milliarden Smartphone-Nutzer machen circa 53,3% aller Mobiltelefonnutzer weltweit aus.<sup>3</sup> Smartphones unterscheiden sich von Mobiltelefonen in der Anzahl der Funktionalitäten, die bei Smartphones die übliche Nutzung eines Telefons bei Weitem überschreiten. Um diese zusätzlichen Funktionalitäten bereitzustellen, werden in Smartphones heutzutage viele Arten von Sensoren eingebaut und verwendet, um Daten zu erfassen. Hierzu zählen beispielsweise

- GPS-Sensoren zur Positionsbestimmung,
- Touchscreens zur einfachen Bedienung des Smartphones,
- Beschleunigungssensoren zur automatischen Ausrichtung des Bildschirms,
- Fingerabdrucksensoren zur Authentifizierung des Nutzers und
- Helligkeitssensoren zur Anpassung der Bildschirmhelligkeit.<sup>4</sup>

Diese Auflistung ist nur ein kleiner Ausschnitt der Technologien, die in der heutigen Zeit verwendet werden. Ein Potenzial, das sich hieraus ergibt jedoch nicht sehr häufig genutzt wird, ist die Erfassung von biometrischen Daten mithilfe dieser Sensoren. Bei biometrischen Daten handelt es sich um menschliche Merkmale, die als Grundlage für verschiedene Arten von Analysen herangezogen werden können. In der Biometrie gängige Verfahren sind beispielsweise

- die Pulsmessung,
- die Gesichtserkennung und
- die Spracherkennung.

Die Studienarbeit betrachtet verschiedene Möglichkeiten mithilfe von Smartphone-Sensoren und eventuell zusätzlicher Hardware, biometrische Daten zu erfassen, diese zu analysieren und so dieses selten genutzte Potenzial auszuschöpfen.

---

<sup>1</sup>Sta18a.

<sup>2</sup>Sta18c.

<sup>3</sup>Sta18b.

<sup>4</sup>Vgl. Bie14.

## 1.2 Zielsetzung

TORBEN BRENNER & LUKAS SEEMANN

Das Ziel dieser Studienarbeit ist es, Möglichkeiten zu erkunden, mit Smartphones biometrische Daten zu erfassen. Dabei werden in das Smartphone integrierte Sensoren, über zusätzliche Hardware angeschlossene Sensoren und die Interaktion des Nutzers mit seinem Smartphone betrachtet. Diese erfassten Daten werden anschließend für Analysen verwendet, die Rückschlüsse auf die Emotionen des Nutzers zulassen. Als finales Produkt soll eine mobile Anwendung für Smartphones entstehen, die den Nutzer verschiedene Tests anbietet, anhand denen die aktuelle Gemütslage beziehungsweise die Emotion des Nutzers bestimmt werden können.

## 1.3 Vorgehensweise

LUKAS SEEMANN

Um die definierten Ziele der Arbeit zu erreichen, unterteilt sich die Arbeit im Folgenden in vier weitere Kapitel.

Im nächsten Kapitel werden zunächst wichtige theoretischen Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Hierzu zählen Definitionen zu den Themen Biometrie, Emotion, Emotionserkennung und die Auswertung von biometrischen Daten. Außerdem werden technische Themen wie die Grundlagen zu Smartphones, Sensoren und mobilen Applikationen thematisiert.

Im dritten Kapitel wird das Konzept der mobilen Anwendung beschrieben. Nachdem festgelegt wurde, welche Art von Daten mithilfe des Smartphones erfasst werden sollen, wird geplant, wie die Daten erfasst werden. In diesem Schritt wird auch bestimmt, ob zusätzliche Hardware benötigt wird oder ob ein handelsübliches Smartphone ausreicht. Im Anschluss daran wird konzipiert, nach welchem Prinzip die erfassten Daten ausgewertet werden. Der letzte Schritt ist der Entwurf eines Entscheidungsalgorithmus, der aus den ausgewerteten Daten eine Emotion des Nutzers bestimmen kann.

Im vierten Kapitel steht die Umsetzung des Konzepts als mobile Applikation im Mittelpunkt. Als erstes wird hierbei die Architektur der App beschrieben. Anschließend werden die konkrete Umsetzung im Ionic-Framework thematisiert und somit im Detail auf den geschriebenen Quellcode eingegangen.

Im Schluss wird ein Fazit zum Ergebnis geliefert und weitere mögliche Schritte des Projekts dargestellt.



## 2 Theoretische Grundlagen

### 2.1 Biometrie und biometrische Merkmale TORBEN BRENNER

Bei der Biometrie handelt es sich um die Wissenschaft, die sich mit der Vermessung von biologischen Merkmalen beschäftigt<sup>5</sup>. Dabei werden insbesondere in der Informationstechnologie Technologien zur Messung und Analyse von körperlichen Merkmalen untersucht. Diese Merkmale werden auch als biometrische Merkmale bezeichnet. Dabei wird unterschieden zwischen den verhaltensbasierten und den physiologischen Merkmalen<sup>6</sup>. Erstere zeichnen sich dadurch aus, dass eine Person aktiv eine Handlung ausführen muss um das Merkmal zu zeigen, während dem die physiologischen Merkmale dauerhaft von einer Person getragen werden. Ein Beispiel für verhaltensbasierte Merkmale ist die Gangart eines Menschen, die sogar zur Authentifizierung verwendet werden kann<sup>7</sup>. Ein bekanntes physiologisches Merkmal ist der Fingerabdruck einer Person.

Ein häufiger Einsatzzweck der Biometrie ist die Authentifizierung eines Nutzers gegenüber einem System. Da sich nicht alle Merkmale für eine solche Identifikation eignen, müssen verschiedene Faktoren bei der Auswahl der Merkmale betrachtet werden. [Akj04] nennt zum Beispiel folgende Faktoren:

- **Universalität:** Jeder Mensch sollte dieses Merkmal besitzen.
- **Unterscheidbarkeit:** Das Merkmal soll sich so stark wie möglich zwischen zwei Personen unterscheiden.
- **Permanenz:** Das Merkmal sollte sich über die Zeit betrachtet nicht oder nur in geringem Maße ändern.
- **Erfassbarkeit:** Das Merkmal sollte möglichst einfach dauerhaft erfasst werden können.
- **Performanz:** Beschäftigt sich mit der Frage mit welchem Zeitaufwand und mit welcher Geschwindigkeit ein Merkmal gemessen werden kann.
- **Akzeptanz:** Beschäftigt sich mit der Frage in wie weit Nutzer mit der Messung eines Merkmals einverstanden sind.
- **Umgehbarkeit:** Beschäftigt sich mit der Frage, in wie weit ein Nutzer dem System vortäuschen kann das er ein anderer Nutzer ist.

Die ersten vier Faktoren beschäftigen sich im Allgemeinen mit der Eignung eines Merkmals für die Authentifizierung, während sich die letzten Faktoren mit der Eignung eines

---

<sup>5</sup>Vgl. Sea18.

<sup>6</sup>Vgl. Sam06.

<sup>7</sup>Vgl. Cla09.

Biometrischen Systems für eine Aufgabe beschäftigen. Da sich unsere Fragestellung aber nicht auf die Authentifizierung eines Nutzers gegenüber eines Informationstechnischen Systems bezieht, sondern sich mit der Auswertung der erfassten Daten für die Erkennung von Emotionen beschäftigt, ist der Faktor der *Unterscheidbarkeit* zu vernachlässigen. Die restlichen Faktoren werden aber später für die einzelnen biometrischen Merkmale untersucht wobei sich die Faktoren *Performanz*, *Akzeptanz* und *Umgehbarkeit* auf die Messung mit Smartphones beziehen.

## 2.2 Emotionen

### 2.2.1 Definition

TORBEN BRENNER

Da wir uns in dieser Arbeit mit der Erkennung von Emotionen beschäftigen, ist es notwendig, dass wir den Begriff der Emotion definieren. Das Problem an dem Begriff der Emotion ist, dass diese ein Hypothetisches Konstrukt ist, welches sich aus der physiologischen Erregung, dem motorischen Ausdruck, Handlungstendenzen und einem subjektiven Gefühl zusammensetzt <sup>(8)</sup>.

Als Beispiel nennt Klaus Scherer hier das plötzliche Auftreten eines Mannes mit einem Blut verschmierten Messer beim Sonntagsspaziergang. Er beschreibt daraufhin welche Aspekte in diesem Szenario eine Rolle spielen. So kann zum einen eine physiologische Reaktion gemessen werden, in Form eines erhöhten Herzschlages. Außerdem wird eine motorische Reaktion stattfinden, z.Bsp. weit aufgerissener Mund und Augen. Die Handlungstendenz wäre in seinem Beispiel der plötzliche Drang wegzulaufen und bei der späteren Befragung zu dieser Situation könnte eine Person sagen, dass sie Furcht gefühlt hat.

[Kla05] definiert eine Emotion im Zusammenhang mit dem *component process model* als “eine Abfolge von zusammenhängenden, synchronisierten Veränderungen der Zustände von allen oder den meisten der fünf Subsysteme des Organismus als Reaktion auf die Verarbeitung eines externen oder internen Stimulus Ereignis das relevant für den Organismus ist”<sup>9</sup>. In einer Tabelle zeigt Scherer die Verbindung zwischen den Organischen Subsystemen und den Komponenten und Funktionen einer Emotion. Anhand dieser Tabelle lässt sich eine Unterscheidung zwischen dem Begriff Gefühl und Emotion durchführen. Der Unterschied ist das ein Gefühl(*subjective feeling component*) eine einzelne Komponente der Emotion ist, welche erst in Verbindung mit den anderen Emotionskomponenten zu einer Emotion führt. Andere Komponenten sind zum Beispiel die Bewertung der Situation(*Cognitive component*) oder körperliche Symptome(*Neurophysiological component*). Der Gesichtsausdruck und die Stimme werden auch als eigene Komponenten(*Motor expression component*) gewertet.

---

<sup>8</sup>Vgl. Kla02, S.166 Abschnitt Emotion.

<sup>9</sup>Übersetzt aus Kla05, S.697 Z.32ff.

<sup>11</sup>Vgl. Kla05, S.698 Table 1.

**TABLE 1**  
**Relationships between organismic subsystems and the functions and components of emotion**

<b>Emotion function</b>	<b>Organismic subsystem and major substrata</b>	<b>Emotion component</b>
Evaluation of objects and events	Information processing (CNS)	Cognitive component (appraisal)
System regulation	Support (CNS, NES, ANS)	Neurophysiological component (bodily symptoms)
Preparation and direction of action	Executive (CNS)	Motivational component (action tendencies)
Communication of reaction and behavioral intention	Action (SNS)	Motor expression component (facial and vocal expression)
Monitoring of internal state and organism–environment interaction	Monitor (CNS)	Subjective feeling component (emotional experience)

*Note:* CNS = central nervous system; NES = neuro-endocrine system; ANS = autonomic nervous system; SNS = somatic nervous system.

Abbildung 1: Relationships between organismic subsystems and the functions and components of emotion - Klaus R. Scherer<sup>11</sup>

### 2.2.2 Eingrenzung der zu messenden Emotionen

TORBEN BRENNER

Bevor geklärt werden kann, wie Emotionen gemessen werden können, ist es notwendig sich auf eine Menge von Emotionen zu einigen. Da es viele unterschiedliche Emotionen gibt, laut Hokuma<sup>12</sup> sind es 34.000 unterschiedliche Emotionen, stellt sich die Einteilung als äußerst schwer da. Einen Versuch die Emotionen einzuteilen hat bereits Robert Plutchick unternommen. Dieser ist zu dem Schluss gekommen, das es acht primäre Emotionen gibt: Freude, Traurigkeit, Akzeptanz, Ekel, Angst, Wut, Überraschung und Erwartung. Um den Zusammenhang zwischen diesen und einigen anderen Emotionen darzustellen, wurde das Rad der Emotionen entwickelt (siehe Abbildung)

Das Rad stellt die primären Emotionen dabei in Relation, wobei die Kombinationen zwischen zwei Emotionen im Raum zwischen diesen steht und Emotionen die gegensätzlich wirken, z. Bsp. Traurigkeit und Freude, jeweils auch gegenüberliegend auf dem Rad sind.

<sup>12</sup>Vgl. Hok17, Absch. 1.

<sup>13</sup>Vgl. Hok17.



Abbildung 2: Rad der Emotionen - Robert Plutchick<sup>13</sup>

Außerdem wird die Stärke einer Emotion durch deren Nähe zum Zentrum des Rads gekennzeichnet, z. Bsp. Wut zu toben<sup>14</sup>.

Somit stellt das Rad der Emotionen eine Eingrenzung der vielen unterschiedlichen Emotionen in mehrere Kategorien dar. Außerdem lässt sich anhand des Rades der Zusammenhang und die Intensität verschiedener Emotionen besser verstehen.

### 2.2.3 Wie lassen sich Emotionen messen?

TORBEN BRENNER

Nach dem nun geklärt ist wie eine Emotion aufgebaut ist, müssen wir uns die Frage stellen, wie es möglich ist eine Emotion zu messen. Der naheliegendste Ansatz ist zu versuchen, die einzelnen Emotionskomponenten messbar zu machen. Das dies nicht so einfach umzusetzen ist, zeigt die Betrachtung der kognitiven Emotionskomponente. Diese steht im direkten Zusammenhang mit dem zentralen Nervensystem (siehe 2.2.1 S.5). Eine Messung der Veränderungen in diesem System ist äußerst kompliziert und insbesondere in unserem Anwendungsfall nicht möglich. Dafür lassen sich aber zum Beispiel körperliche Symptome, wie z.Bsp. der Puls, messen und auswerten.

Ein weiterer Ansatz Emotionen zu messen, ist das von Scherer entwickelte *Geneva Emotion Wheel*. Das *Geneva Emotion Wheel* baut auf dem Ansatz auf, die Emotionen anhand

<sup>14</sup>Vgl. Hok17, Absch. Elements of the Wheel.

verschiedener Dimensionen zu bestimmen. In einem Prototyp für das *Geneva Emotion Wheel* (zu sehen in 2.2.3 S.7) werden die beiden Dimensionen *arousal* und *valence* betrachtet. Die erste Dimension bezeichnet wie stark die Erregung eines Individuums

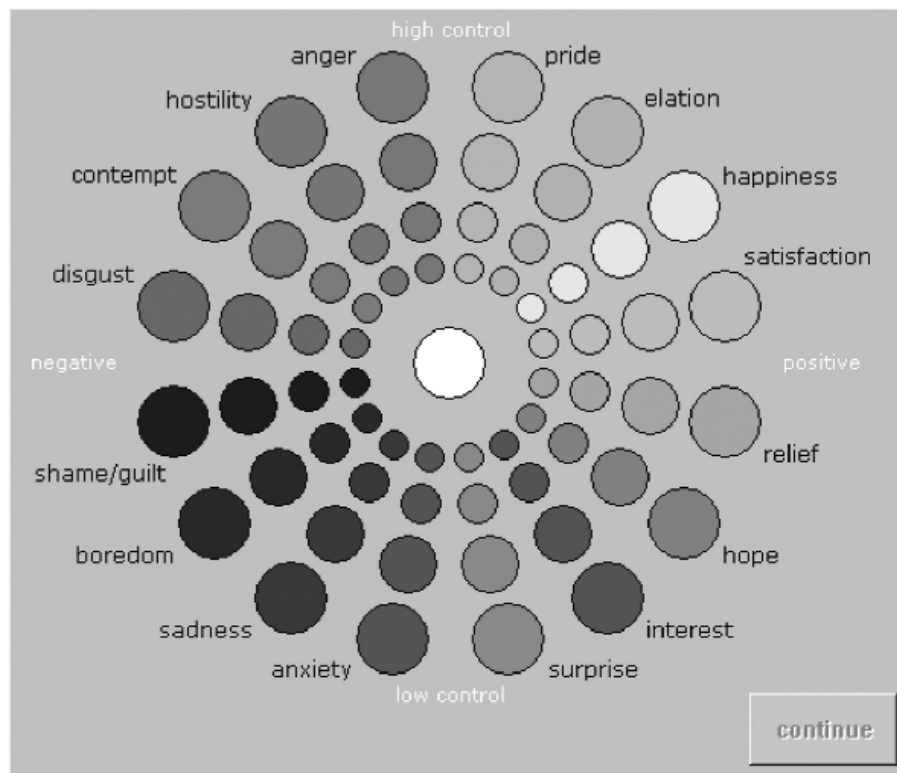


Abbildung 3: Prototype version of the Geneva Emotion Wheel - Klaus R. Scherer<sup>15</sup>

ausgeprägt ist in einer Situation. Die zweite Dimension gibt an wie Unwohl sich ein Individuum in einer Situation fühlt. Beide Dimensionen haben gemeinsam, dass sie durch die Befragung eines Individuums ermittelt werden müssen.

## 2.3 Umgang mit biometrischen Daten

TORBEN BRENNER

Eine Problematik, mit der wir uns in dieser Arbeit beschäftigen müssen, ist der Umstand das biometrische Daten nicht immer einen direkten Schluss auf einen Emotion zulassen. So lässt ein hochfrequenter Puls keinen direkten Schluss auf die Emotion zu, die ein Individuum gerade empfindet. Er kann maximal ein Indiz für verschiedene Emotionen sein, z. Bsp. Wut oder Angst. Um mit diesem Umstand umzugehen benötigen wir zwei neue Begriffe die im folgenden genauer erläutert werden.

### 2.3.1 Indiz

Ein Indiz ist im allgemeinen Sprachgebrauch ein Anzeichen für einen Umstand, an dem sich ein Zustand oder eine Entwicklung absehen lässt<sup>16</sup>. In unserer Arbeit, sehen wir Daten

<sup>15</sup>Vgl. Kla05, S.723 Figure 2.

<sup>16</sup>Vgl. Dud18.

die wir von den Sensoren bekommen, als Indizien an. Ein Indiz macht es wahrscheinlicher bzw. unwahrscheinlicher das ein Individuum eine bestimmte Emotion verspürt.

### 2.3.2 Kausalität

Als Kausalität wird im allgemeinen der Zusammenhang zwischen Ursache und Wirkung verstanden. In der Physik ist die Kausalität ein grundlegendes Prinzip, welches besagt, “daß in der Natur nichts ohne Grund passiert, d.h. zu jedem Ereignis (Wirkung) ein anderes (Ursache) existiert, das a) in seiner Vergangenheit liegt und b) zwingende Voraussetzung für das Eintreten der Wirkung ist”<sup>17</sup>.

In dieser Arbeit werden wir ebenfalls versuchen, kausale Zusammenhänge zwischen Reaktionen des Körpers und den gerade empfundenen Emotionen zu ermitteln. Ein Werkzeug um kausale Zusammenhänge darzustellen ist in der Literatur der Kausale Graph (im englischen *directed acyclic graph*).<sup>19</sup> Die Grafik zeigt ein fiktives Beispiel für einen kausalen

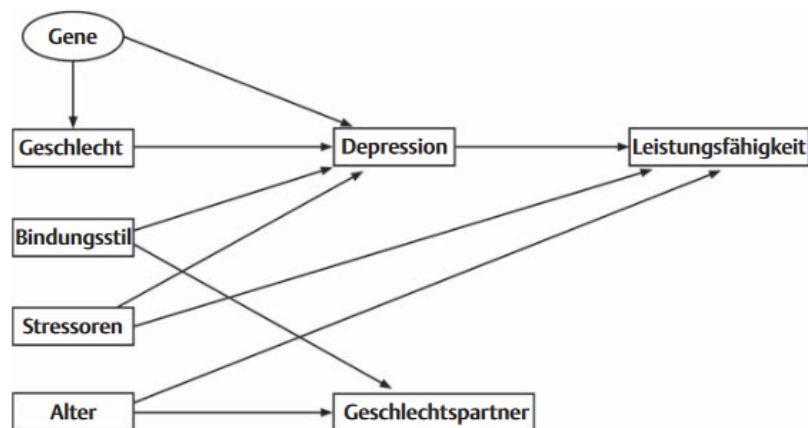


Abbildung 4: Fiktives Beispiel eines DAGs<sup>18</sup>

Graphen. In diesem Beispiel von Thoemmes wird dargestellt das Bindungsstil, Geschlecht, Stressoren und Gene Einfluss auf Depressionen haben. Wichtig ist, dass alle Annahmen die in einem solchen Graph gemacht werden theoretisch begründet werden müssen. Ist dies nicht der Fall, dürfen sie kritisiert und infrage gestellt werden<sup>20</sup>.

## 2.4 Emotionsindizien

TORBEN BRENNER

Dieser Abschnitt beschäftigt sich mit der Frage, welche biometrischen Merkmale existieren und inwiefern sie sich zur Bestimmung von Emotionen eignen. Daten die sich zur Bestimmung von Emotionen eignen nennen wir Emotionsindizien, da sie einen Hinweis auf die empfundenen Emotionen einer Person liefern.

<sup>17</sup>Sav18.

<sup>19</sup>Vgl. Tho11, Kausale Graphen - DAGs.

<sup>20</sup>Vgl. Tho11, S.3 Kausale Graphen - DAGs.

In diesem Abschnitt beschäftigen wir uns mit dem Puls als ein Indiz für verschiedene Emotionen. Hierbei wird der als der biologische Puls gesehen, das heißt “die in Abhängigkeit vom Herzrhythmus (Herzmechanik) erfolgende Schwankung von Blutstrom, Blutdruck oder Blutvolumen im Blutkreislaufsystem (Blutgefäßsystem, Blutkreislauf)”<sup>21</sup>. Neben dieser Definition wird im Lexikon der Biologie auch folgende Definition genannt: “die vom Herzschlag bewirkte, rhythmisch auftretende Druckwelle (Pulsschlag) in den Arterien”<sup>22</sup>, welche den Puls als arteriellen Puls definiert.

Es gibt mehrere Möglichkeiten den Puls zu messen. Die häufigste Methode, welche auch in Erste-Hilfe-Kursen gelehrt wird, ist die Messung des Pulses an der Arterie. Das Grundprinzip ist dabei, dass die vom Herzschlag verursachte Druckwelle an der Arterie für Menschen spürbar ist und somit dort mitgezählt werden kann.

Der Puls ist in der Biometrie als ein physiologisches Merkmal zu sehen. Eine Person muss keine speziellen Handlungen durchführen um einen Puls zu haben. Der Aspekt der *Universalität* ist bei diesem Merkmal gegeben, da jeder Mensch einen Puls besitzt. Eine Erfassung des Pulses ist quasi jeder Zeit möglich, zumindest für Menschen. Ein Smartphone bietet aber auch mehrere Möglichkeiten den Puls zu messen, die später in dem Abschnitt “Erfassung der biometrischen Daten” weiter erläutert werden. Der Aspekt der Permanenz ist nicht vollständig gegeben. Der Puls bewegt sich zwar standardmäßig in einem bestimmten Wertebereich, ist aber abhängig von Alter und körperlicher Verfassung einer Person.

Als Emotionsindiz kann der Puls unterstützend wirken, da dieser eine physiologische Auswirkung von verschiedenen Emotionen sein kann. Dennoch ist er kein alleiniges Merkmal für eine bestimmte Emotion, da sowohl Freude als auch Angst einen erhöhten Puls zur Folge haben können.

## 2.4.2 Hautleitfähigkeit/Hautwiderstand

LUKAS SEEMANN

Die menschliche Haut verfügt über „*aktive als auch passive elektrische Eigenschaften, die sich auf Strukturen und Funktionen der Haut und der in ihr enthaltenen Organe zurückführen lässt.*“<sup>23</sup> Diese elektrischen Phänomene der Haut sind in wissenschaftlichen Kreisen unter dem Sammelbegriff elektrodermale Aktivität (kurz EDA) bekannt.<sup>24</sup> Eine elektrodermale Aktivität, die sich sehr gut als Indikator für Emotionen eignet, ist die Hautleitfähigkeit. Hierzu wird mit einer externen Stromquelle mit geringer Spannung gemessen, wie gut die Haut eines Probanden diesen Strom leitet.<sup>25</sup> Häufig wird anstand

---

<sup>21</sup>Spe18.

<sup>22</sup>Spe18.

<sup>23</sup>Bou88, S. 2.

<sup>24</sup>Vgl. Bou88, S. 2.

<sup>25</sup>Vgl. Moe07, S.77.

der Hautleitfähigkeit auch der Hautwiderstand gemessen. Diese beiden Indizien stehen in einer negativ proportionalen Beziehung. Dies bedeutet, je höher der Widerstand der Haut ist, desto niedriger ist die Leitfähigkeit und umgekehrt. Letzten Endes sagen beide Indizien dasselbe aus, unterscheiden sich aber in der Betrachtungsrichtung.<sup>26</sup>

Die Hautleitfähigkeit wird anhand der Menge von Schweiß an den Ausgängen der Schweißdrüsen bestimmt, die sich über den gesamten Körper verteilen. Je mehr Schweiß, der elektrisch sehr gut leitend ist, sich auf der Haut befindet, umso größer ist die Hautleitfähigkeit. Am besten eignen sich Stellen, an denen die Schweißdrüsen sehr dicht angeordnet sind und die somit sehr schweißsensibel sind. Dies ist zum Beispiel an den Handinnenflächen beziehungsweise Fingerinnenseiten der Fall, die sich deshalb sehr gut für solche Messungen eignen.<sup>27</sup>

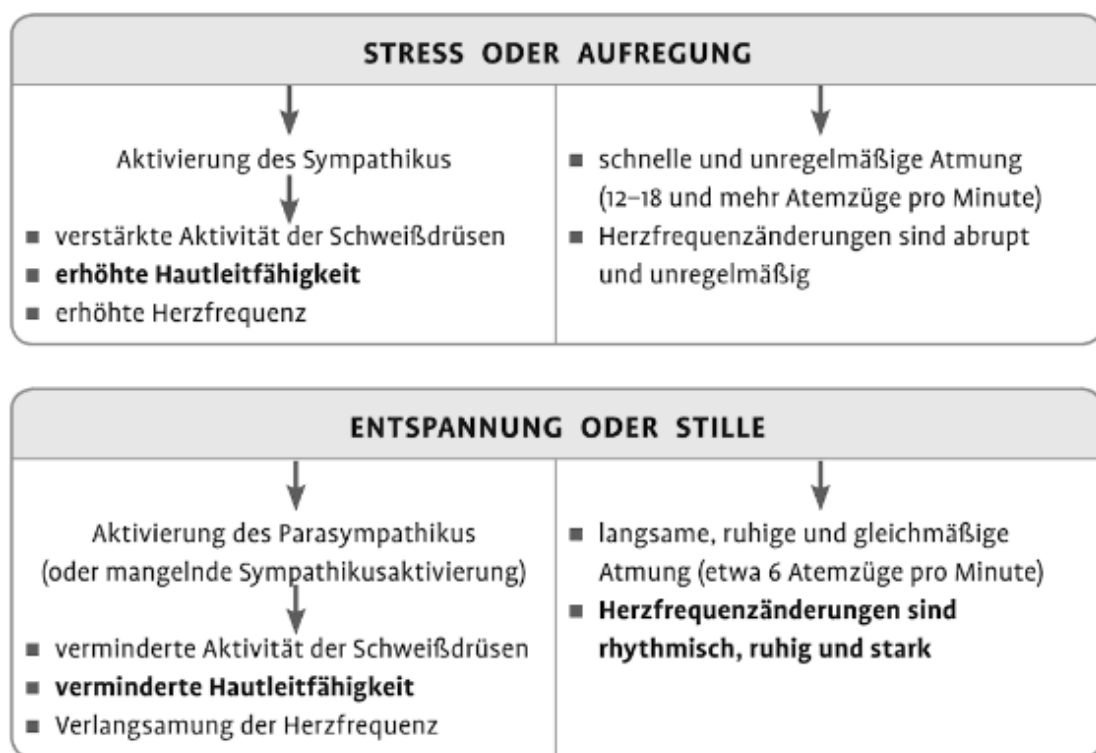


Abbildung 5: Reaktion des Körpers auf Stress und Entspannung<sup>28</sup>

„Die Aktivierung beschreibt das Ausmaß der physiologischen Aktiviertheit oder Wachheit eines Menschen“<sup>29</sup>. Unter Aktivierung versteht man bei Menschen generell jede Art von emotionaler Erregung. Hierzu zählen unter anderem Wut, Aufregung, Schreckmomente oder auch extreme Freude. In Abbildung 3 ist die Reaktion des Körpers auf Stress (darausfolgend auch Aktivierung) und auf Entspannung dargestellt. Die Schweißproduktion wird über das unwillkürliche Nervensystem gesteuert. Dieses besteht aus Sympathikus

<sup>26</sup>Vgl. Die06, S. 28.

<sup>27</sup>Vgl. Moe07, S.77.

<sup>28</sup>Dil13, S. 200.

<sup>29</sup>Die06, S. 28.



der für die Bereitstellung von Energie und Arbeitsleistung zuständig ist, und dem Parasympathikus, der zur Erholung und Wiederherstellung von Körperfunktionen dient.<sup>30</sup>

Bei Stress oder Aufregung wird der Sympathikus aktiviert, was eine verstärkte Schweißproduktion und somit auch eine erhöhte Hautleitfähigkeit hervorruft. Außerdem wird die Herz- und Atemfrequenz erhöht und der Rhythmus dieser ist unregelmäßig. Die Reaktion auf ein Ereignis, das emotionale Erregung hervorruft, lässt sich meistens innerhalb von einer bis vier Sekunden anhand der Änderung der Hautleitfähigkeit feststellen.<sup>31</sup>

Bei Entspannung hingegen wird der Parasympathikus aktiviert, was zu einer verminderten Aktivität der Schweißdrüsen führt. Die Hautleitfähigkeit sinkt somit auch. Des Weiteren werden Herz- und Atem verlangsamt und gelangen wieder in einen normalen Rhythmus.

Der Vorteil der Messung der Hautleitfähigkeit ist, dass diese unwillkürlich gesteuert wird und somit keine willentliche Mitarbeit des Probanden erfordert. Da die Aktivierung des Sympathikus automatisch geschieht, kann der Proband die Messung nicht verfälschen.

Ein Nachteil des Verfahrens ist, dass die Hautleitfähigkeit nur Rückschlüsse auf den Grad der Aktivierung schließen lässt, jedoch nicht gesagt werden kann, ob es sich um positive oder negative Reaktionen handelt. Die Wut über ein Ereignis würde zum selben Ergebnis führen, wie die übermäßige Freude über ein Ereignis. Aus diesem Grund müssen zur genauen Emotionsbestimmung weitere Indizien herangezogen werden.<sup>32</sup>

### 2.4.3 Mimik

TORBEN BRENNER

Der Begriff Mimik bezeichnet Bewegungen der Gesichtsmuskulatur mit dem Ziel, eine Emotion auszudrücken. Das nicht jede Gesichtsregung automatisch auch als Mimik gewertet werden kann sieht man beispielsweise beim Kauen<sup>33</sup>.

Der Vorteil der Mimik als Emotionsindiz ist, dass sie als eines der wenigen Indizien für direkte Aussagen zu den Emotionen gewählt werden kann. So ist es dem Menschen zum Beispiel möglich, alleine durch Beobachtung einer anderen Person Vermutungen anzustellen, welche Emotionen diese gerade empfindet<sup>34</sup>. Eine Untersuchung die diese Aussage zusätzlich unterstützt ist der FEEL-Test<sup>35</sup>. FEEL steht für Facial Expressed Emotion Labeling und ist ein Computerprogramm das auf den Arbeiten von Ekman<sup>36</sup> aufbaut. Bei dem Test werden den Probanden Bilder gezeigt die jeweils eine von sechs Basisemotionen widerspiegeln. Der Proband soll daraufhin versuchen die Emotionen zu erkennen wobei im Ergebnis die Richtigkeit der Antwort und die Antwortzeit betrachtet werden. Ziel des

---

<sup>30</sup>Vgl. Lie14, S. 5.

<sup>31</sup>Vgl. Sch14, S. 130f.

<sup>32</sup>Vgl. Moe07, S.77.

<sup>33</sup>Vgl. Kar16, Mimik: Eine kurze Definition.

<sup>34</sup>Vgl. Kar16, Die sieben Grundemotionen, Absatz 1.

<sup>35</sup>Kes02.

<sup>36</sup>Ekman92.

Tests war es “objektiv und reliabel die Fähigkeit eines Probanden erfassen, sechs misch kodierte Basisemotionen zu erkennen (Freude, Trauer, Ekel, Angst, Überraschung und Ärger)”<sup>37</sup>. In einer Pilotstudie mit 77 Teilnehmern konnten die Probanden unterschiedlich gut die verschiedenen Emotionen erkennen: “(Trauer 70%, Angst 71%, Ekel 80%, Überraschung 84%, Freude 87% und Ärger 94%)”<sup>38</sup>.

Die Ergebnisse der Studie zeigen, dass Menschen aus der Mimik einen Schluss auf eine Emotion ziehen können. Im Verlauf der Arbeit werden wir untersuchen, ob es Möglichkeiten gibt, mit der Smartphonekamera ebenfalls ein solches Ergebnis erzielen können oder dieses sogar übertreffen können.

#### 2.4.4 Tippverhalten

TORBEN BRENNER

Unter dem Tippverhalten wird die individuelle Art einer Person verstanden, wie sie auf virtuellen oder realen Tastaturen tippt. Dabei wird unter anderem der Eingaberythmus als Erkennungsmerkmal betrachtet<sup>39</sup>. Die Erkennungsgenauigkeit, und damit auch die Sicherheit des Verfahrens, ist dabei von der Frequenz der abgefragten Werte und der eingestellten minimalsten Ähnlichkeit abhängig. Bei diesem biometrischen Merkmal handelt es sich um ein verhaltensorientiertes Merkmal. Erfassbar ist dieses Merkmal immer, wenn der Nutzer einen vorgegebenen Text eingeben muss und für diesen ein Vergleichsprofil vorhanden ist. Dies kann zum Beispiel auch zur zusätzlichen Prüfung bei der Passworteingabe verwendet werden.

Es gibt wenige Arbeiten die sich mit der Verbindung von Emotionen und dem Tippverhalten beschäftigen. Eine dieser wenigen Arbeiten ist von Trojahn, Arndt, Weinmann und Ortmeier. Diese stellten bezüglich dieses Themas fünf Thesen auf und prüften diese innerhalb ihrer Arbeit<sup>40</sup>. Drei dieser Hypothesen beziehen sich dabei direkt auf die Verbindung zwischen Tippverhalten und Emotionen, weshalb sie hier genannt werden<sup>41</sup>:

- **H3:** [Je langsamer die Tippgeschwindigkeit, desto negativer werden Emotionen wahrgenommen]
- **H4:** [Je höher die Tippfehlerrate, desto negativer werden Emotionen wahrgenommen]
- **H5:** [Je höher der Fingerdruck, desto negativer werden Emotionen wahrgenommen]

Um die Hypothesen zu prüfen, führten sie eine Studie mit zwei Samsung Galaxy Nexus Smartphones aus, bei der es die Aufgabe der Probanden war mit Hilfe des Smartphones einen Text einzutippen. Dabei wurden denn Probanden unterschiedliche Zeitgrenzen gegeben um unterschiedliche Ebenen von Stress zu repräsentieren. Nachdem der Test

---

<sup>37</sup>siehe. Kes02, S.5 Z.11ff.

<sup>38</sup>siehe. Kes02, S.9 Z.9f.

<sup>39</sup>Vgl. Bio18b, Allgemein Abs. 1.

<sup>40</sup>Vgl. Tro13, S.32 2.3 Hypothesen.

<sup>41</sup>siehe Tro13, S.33 Z.8-13.

durchgeführt war, sollten die Probanden noch einen Fragetext zu ihren empfundenen Emotionen beantworten<sup>42</sup>. Durch ihre Studie konnten sie zeigen, dass Emotionen durch Tastendrucke beschrieben werden können. In diesem Zusammenhang wurden die Hypothesen 3 und 4 bestätigt, aber Hypothese 5 widerlegt<sup>43</sup>.

Durch diese Studie wurde gezeigt, dass das Tippverhalten ebenfalls als Emotionsindiz gewertet werden, weshalb es auch weiter untersucht werden soll.

### 2.4.5 Gangart

Wie bereits im Abschnitt Biometrie und biometrische Merkmale(siehe S.3) erläutert, ist auch die Gangart eines Menschen ein biometrisches Merkmal. Die Gangerkennung setzt dabei darauf, "dass jeder Mensch eine relativ spezifische Gangart besitzt, an der er unter Zuhilfenahme anthropometrischer Maße wie Beinlänge und Bein Form nahezu eindeutig erkannt werden kann"<sup>44</sup>. So entstehen beim gehen verschiedene sich wiederholende Muster, die sich mit Hilfe von Sensoren erkennen lassen. So existieren mehrere Merkmale, die sich dazu eignen den menschlichen Gang zuerkennen, hier sei z. Bsp. die Schrittweite oder die Schrittzeit genannt.

Verschiedene wissenschaftliche Quellen treffen die Aussage, dass sich Emotionen auch auf die Bewegung einer Person auswirken können. So stellen Walk & Homan in ihrer Arbeit folgende Theorie auf.

"Although emotion may be expressed in the face, it can also be expressed in the voice and in body movement. People jump with joy or cringe with fear."<sup>45</sup>

In ihrer Arbeit haben sie ein Experiment mit 24 Studenten durchgeführt, denen sie Videosequenzen vorgespielt haben in denen zwei verschiedene Schauspielerinnen, verschiedene Körperliche und Emotionale Zustände dargestellt haben. Die Studenten hatten in der ersten Phase des Experiments, die Aufgabe, die gesehenen Videosequenzen zu beschreiben. In der zweiten Phase, in der den Studenten die Aufnahme erneut gezeigt wurde, hatten diese die Aufgabe einzuschätzen wie viele verschiedene Personen sie in dem Video gesehen haben und welches Geschlecht diese haben<sup>46</sup>. Als ein Ergebnis dieses Experiments stellten die Autoren in einer Tabelle(siehe Abbildung) ihre Ergebnisse mit denen von Ekman<sup>47</sup> in Vergleich. In der Grafik ist deutlich zu sehen, dass die Teilnehmer der Studie von Walk & Homan beim ersten sehen der Videos Probleme hatten, die Emotionen zu erkennen. Einzig die Emotion *Wut* wurde von einem Großteil der Teilnehmer erkannt. Beim

---

<sup>42</sup>Vgl. Tro13, S.33 3.1 Study Object and Study Task.

<sup>43</sup>Vgl. Tro13, S.35 5.1 Summary.

<sup>44</sup>siehe Bio18a, Abschnitt Allgemein Abs.2 Z.2.

<sup>45</sup>siehe Wal84, S.437 Z.20-23.

<sup>46</sup>Vgl. Wal84, S.437+438 Method.

<sup>47</sup>Ekman92.

<sup>49</sup>siehe. Wal84, Tabelle 1 S.438.

**Table 1**  
**Event Study Compared with Studies of Facial Expression of Emotion (Values in Percentages)**

	Event Study		Ekman and Friesen's Data			Bassili's Study	
	1st Time	2nd Time	U.S.	Literate Cultures	Fore of New Guinea	Moving Full Face	Moving Spots
Happiness	25	96	97	92	92	31	31
Fear	42	71	88	74	80	69	6
Surprise	0	79	91	88	68	94	75
Disgust/Contempt	0	71	82	84	81	88	57
Anger	88	88	69	74	84	50	6
Sadness	8	75	73	83	79	56	25
No. of Observers	24	24	99	356	189	192	192

*Note—Literate cultures are Japan, Brazil, Chile, and Argentina. Percentages for them are a rough median of the four cultures (e.g., median of 87%, 97%, 94% = 92%).*

Abbildung 6: Event Study Compared with Studies of Facial Expression of Motion(Values in Percentages)<sup>49</sup>

zweiten Ansehen des Videos, konnten die Probanden aber deutlich besser die verschiedenen Emotionen erkennen. In diesem Fall war es für die sechs untersuchten Emotionen immer mindestens 70% möglich die dargestellte Emotion zu erkennen. Im Vergleich mit der Studie von Ekman sind diese Ergebnisse schon deutlich Konkurrenzfähiger. Dennoch scheinen die Probanden anhand der Untersuchungsergebnisse, eher anhand des Gesichtsausdrucks, als den Bewegungen eines Menschen dessen Emotionen erkennen zu können. Dennoch ist hier anzumerken, dass sich die Teilnehmer Anzahl der beiden Studien deutlich unterscheidet und bei dem Experiment von Ekman eine gemischte Personen Gruppe genutzt wurde, während dem Walk & Homan ausschließlich Studenten und Studentinnen untersuchten.

Dennoch zeigt das oben vorgestellte Experiment das eine Emotionserkennung anhand des Gangs möglich ist und deshalb wird diese Als Emotionsindiz aufgenommen.

#### 2.4.6 Stimme

TORBEN BRENNER

Die menschliche Stimme kann ebenfalls als biometrisches Merkmal gesehen werden. Dabei ist insbesondere zu beachten, dass bei der Stimme der Faktor *Permanenz* nicht gegeben ist, da diese sich im Verlaufe des Lebens ändert. Dies geschieht über Veränderungen in der Physiologie eines Menschen, z. Bsp. die Stimmbandlänge und die Kehlkopfgröße. Solche Änderungen geschehen unter anderem während dem Alterungsprozess, die stärkste davon in der Pubertät. Das Problem kann aber durch eine ständige Nachkalibrierung während jedem Identifikationsprozesses<sup>50</sup> behoben werden.

[Pat] haben in ihrer Arbeit nach möglichen akustischen Merkmalen für Emotionen gesucht. Dazu haben Sie die Aussprache des Vokals “a” von zehn unterschiedlichen französischen Schauspielern untersucht. Die Schauspieler haben für die GEMEP Datenbank diesen Vokal jeweils in 12 unterschiedlichen emotionalen Kontexten wieder gegeben<sup>51</sup>. Dabei her-

<sup>50</sup>Vgl. Til11, S.198 Z.22-26.

<sup>51</sup>Vgl. Pat, S.2 Z.13-16.

ausgekommen ist, dass einige dieser Merkmale sich dazu eignen, Erregung in der Stimme festzustellen<sup>52</sup>. Mit der Erregung lässt sich bereits zwischen einigen Emotionen unterscheiden, weshalb diese auch im *Geneva Emotion Wheel* als eigene Dimension gewählt wurde. Aus diesem Grund ist auch die Analyse der Stimme ein für die Emotionserkennung relevanter Bereich.

## 2.5 Nützliche Hardware

### 2.5.1 Smartphones

TORBEN BRENNER

Auch wenn der Begriff des Smartphones heutzutage häufig Synonym zu dem des Mobiltelefons verwendet wird, beschreiben die beiden Begriffe dennoch unterschiedliche Gerätearten. Smartphones sind im allgemeinen eine Weiterentwicklung des herkömmlichen Mobiltelefons. Neben der Standardfunktion des mobilen Telefonierens und der Unterstützung des *Short Message Service* (kurz SMS), bieten Smartphones mittlerweile viele weitere Funktionen wie z. Bsp. GPS oder Internetzugriff<sup>53</sup>.

Dieser erhöhte Umfang an Funktionen ermöglicht es, Entwicklern Anwendungen für viele verschiedenen Anwendungsfälle bereitzustellen und somit den Nutzer deutlich besser in seinem Alltag zu unterstützen als mit einem Mobiltelefon<sup>54</sup>. Eine weitere Entwicklung, die vom Mobiltelefon zum Smartphone stattfand, war die Weiterentwicklung des Betriebssystems. Während Mobiltelefone meist mit simplen Betriebssystemen ausgestattet sind, werden Smartphones von komplexen Systemen wie Android oder IOS angetrieben. Diese bieten Nutzern die Möglichkeit zusätzliche Programme zu installieren wie z.Bsp. bei Windows<sup>55</sup>.

Der erhöhte Funktionsumfang von Smartphones im Vergleich zu Mobiltelefonen führt uns zur Frage, ob Smartphones auch dazu genutzt werden können, einen Computer zu ersetzen. Verschiedene Hersteller wie z. Bsp. Samsung mit den Dex-Stationen bieten Nutzern bereits die Möglichkeit, ihr Smartphone als PC-Ersatz zu nutzen<sup>56</sup>. Obwohl Smartphones immer höhere Leistungsspitzen erreichen, werden sie wahrscheinlich auch in Zukunft nicht als vollständiger PC-Ersatz genutzt werden können. Das liegt insbesondere daran das Smartphone CPUs (Central Processing Unit) aufgrund des Formfaktors nicht so gekühlt werden können wie PC-CPU's<sup>57</sup>.

Nun kommen wir zu der Frage warum wir Smartphones verwenden wollen um Emotionen zu erkennen. Zum einen sind Smartphones in der Lage, durch verschiedenen verbaute Sensoren zu sehen, fühlen und zu hören<sup>58</sup>. Wie in dem Artikel beschrieben, geben uns

---

<sup>52</sup>Vgl. Pat, S.4 Z.28-32.

<sup>53</sup>Vgl. Bou11, S.3 Z.5ff.

<sup>54</sup>Vgl. Ada18, Smartphones are tiny Computers.

<sup>55</sup>Vgl. Ada18, Mobile Operating Systems.

<sup>56</sup>Kai18.

<sup>57</sup>Vgl. Gav18, Power and Heat.

<sup>58</sup>Vgl. Bie14, S. 1 Abs. 2.

die verbauten Sensoren enorme Möglichkeiten mit dem Smartphone die Umwelt wahrzunehmen. Wie leistungsfähig die verschiedenen Komponenten mittlerweile geworden sind, zeigt Apple mit dem iPhone X, welches mithilfe der Kamera und Infrarotsensoren, die Veränderungen von Gesichtszügen erkennt und auf sogenannte Animojis überträgt<sup>59</sup>. Ein weiterer Grund aus dem wir die Emotionserkennung mit Smartphones untersuchen wollen ist die Tatsache, dass Smartphonebenutzer ihr Gerät häufig rund um die Uhr bei sich tragen. Dies ermöglicht es für verschiedene Anwendungsszenarien eine Überwachung der Emotionen des Nutzers über den Tag hinweg durchzuführen.

### 2.5.2 Smartphone Sensoren

TORBEN BRENNER

Wie bereits erwähnt, werden in Smartphones mittlerweile massenweise Sensoren verbaut, die eine Umgebungswahrnehmung möglich machen. Eine Auflistung dieser Sensoren stellt Biermann in seinem Artikel für die Zeit Online vor<sup>60</sup>.

So nennt er den Beschleunigungssensor, das sogenannte Akzelerometer. Bei diesem wird mithilfe eines kleinen Siliziumstabs und einer Elektrode die Beschleunigung gemessen. Diese Konstruktion wird dreimal verwendet um für die x-,y- und z-Achse eine Beschleunigungsbestimmung durchführen zu können.

Das Gyroskop ist ein weiterer Sensor, der in Smartphones platz findet. Dieser dient dazu, mit Hilfe von in Schwingung versetzten Metallelementen und Kondensatoren, zu prüfen, ob das Handy hoch oder quer gehalten wird.

Das sogenannte Magnetometer misst auf der X-, Y- und Z-Achse die Stärke und Richtung des Erdmagnetfeldes. Laut Biermann ist z. Bsp. im iPhone 4 ein solcher Sensor verbaut. Dieser kann unter anderem dazu genutzt werden, Stromleitungen in einer Wand zu finden.

Der Touchscreen selbst ist ebenfalls ein Sensor. Dieser besteht aus zwei Gittern, zwischen denen ständig Stromsignale ausgetauscht werden. Dabei wird von der oberen Schicht ein Impuls an die untere Schicht gesendet. Durch Berührung mit dem Finger, und dessen Feuchtigkeit, wird das Signal lokal schwächer. Diese Veränderung wird von einem Prozessor zur Position des Fingers umgerechnet.

Dies war nur eine kleine Auswahl aller Sensoren die in Smartphones verbaut sind. Neben den genannten Sensoren werden z.Bsp. auch die Kamera und das Mikrofon als Sensoren verstanden. Diese werden aber in späteren Kapiteln noch einmal genauer beschrieben.

### 2.5.3 Externe Sensoren

TORBEN BRENNER & LUKAS SEEMANN

Da es nicht möglich ist, alle Indizien die in Kapitel 2.4 aufgezählt wurden mithilfe der Smartphone-internen Sensoren zu erfassen, müssen zusätzliche externe Sensoren zur Erfassung von biometrischen Daten hinzugezogen werden.

---

<sup>59</sup>Vgl. Com17, Animoji: So funktioniert es.

<sup>60</sup>siehe Bie14.

Sensoren dienen zur Erfassung eines physikalischen Zustandes und der Transformation dieses Zustandes in einen Impuls, der verarbeitet werden kann<sup>61</sup>. In der Computertechnik dienen Sensoren häufig dazu, den physikalischen Zustand verständlich für einen Computer zu machen, indem dieser in einen Datensatz umgewandelt wird. Generell sind für einen Computer elektrische Signale erkennbar, jedoch sind nicht alle Messgrößen, die am Eingang eines Sensors anliegen elektrisch. Die Aufgabe von Sensoren ist es also, nicht-elektrische Messgrößen in ein elektrisches Signal umzuwandeln.

In Abbildung 7 ist die Umsetzung dieses Vorgangs gezeigt.

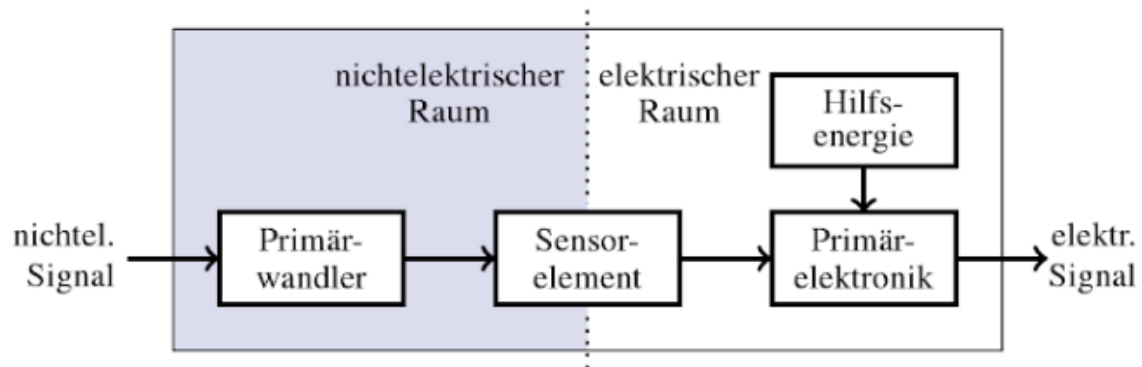


Abbildung 7: Sensor als Schnittstelle zwischen nichtelektrischen und elektrischen Raum<sup>63</sup>

Ein Sensor erhält eine nichtelektrische Messgröße als Input und erfasst diese Messgröße mithilfe eines vorgelagerten Primärwandlers. Ein Primärwandler kann je nach Messgröße unterschiedlich gestaltet sein. Bei mechanischen Größen kann es zum Beispiel eine Membran oder ein Biegebalken sein, bei optischen Größen optisch abbildende Systeme wie zum Beispiel eine Linse oder eine Blende.<sup>64</sup>

Das sogenannte Sensorelement stellt den eigentlichen Sensor dar, der den Übergang von nichtelektrischen in den elektrischen Raum realisiert. Hier geschieht die Umwandlung der Messgröße in ein elektrisches Signal.<sup>65</sup>

Da die Änderungen der elektrischen Größe am Ausgang des Sensorelements in der Regel sehr klein sind, muss das Signal mittels einer Primärelektronik in ein gut verarbeitbares elektrisches Signal verstärkt werden. Dazu ist häufig eine Hilfsenergie notwendig, die dem Sensor zugeführt werden muss.<sup>66</sup> Ist dies geschehen, kann das elektrische Signal des Sensors von Computern analysiert werden.

Eine Möglichkeit, externe Sensoren mit dem Smartphone zu verbinden, ist der Einsatz von Mikrocontrollern, wie zum Beispiel ein Arduino.<sup>67</sup> Da nur wenige externe Sensoren direkt an das Smartphone angeschlossen werden können und Mikrocontroller meist sehr

<sup>61</sup>Vgl. Web18.

<sup>63</sup>Ros13, S. 347.

<sup>64</sup>Vgl. Ros13, S. 347.

<sup>65</sup>Vgl. Ros13, S. 347.

<sup>66</sup>Vgl. Ros13, S. 347.

<sup>67</sup>Arduino UNO R3 Mikrocontroller: Ard18a.

handlich sind, ist diese Möglichkeit sehr praktisch. Der Mikrocontroller übernimmt dann zum einen die Erfassung des elektrischen Signals des Sensors und die Übertragung der Daten zum Smartphone.

## 2.6 Möglichkeiten der Erfassung von biometrischen Daten

TORBEN BRENNER

In diesem Abschnitt werden unterschiedliche Möglichkeiten vorgestellt, die in 2.4 vorgestellten Emotionsindizes zu erfassen. Da es nicht immer nur eine Möglichkeit gibt, diese zu erfassen, werden hier verschiedene Methoden diskutiert. Pro Emotionsindex wird dann eine Methode ausgewählt, die untersucht werden soll.

### 2.6.1 GSR-/EDA-Sensoren

LUKAS SEEMANN

Die bereits in Kapitel 2.4.2 beschriebene elektrodermale Aktivität kann mithilfe von EDA-Sensoren gemessen werden. Diese Art von Sensoren sind üblicherweise nicht in Smartphones eingebaut, weswegen die Hautleitfähigkeit mit externer Hardware gemessen werden muss.

Die für heute eher unübliche Messungsart der elektrodermalen Aktivität stellt die endosomatische Messung dar. Indem winzige Elektroden in die Haut eingestochen werden, kann die Aktivität der Nerven in der Haut gemessen werden.<sup>68</sup> Hierbei handelt es sich jedoch nicht um eine Messung der Hautleitfähigkeit sondern um eine Messung des Hautpotenzials.

Bei der exosomatischen Messung hingegen wird ein schwacher Strom von ungefähr 0.5 Volt an die Haut angelegt. Die Spannung wird hierbei konstant gehalten, wodurch die Leitfähigkeit der Haut gemessen werden kann. Hierbei gemessen wird entweder in der Einheit Siemens für den elektrischen Leitwert oder in Ohm für den Widerstand. Die Elektroden des Sensor sind meistens aus Silber oder Silberchlorid und werden meist an zwei Stellen der nicht dominanten Hand angebracht.<sup>70</sup> Wie in Abbildung ? dargestellt gibt es verschiedenen Möglichkeiten der Handinnenfläche, die gut geeignet sind, (#1, #2 oder #3).

Diese Art von Sensoren sind die heutzutage übliche Vorgehensweise bei EDA-Messungen. Häufig sind sie auch unter dem Namen GSR-Sensor<sup>71</sup> bekannt und darunter im Internet erhältlich. GSR-Sensoren sind als Modul für den Mikrocontroller Arduino verfügbar.<sup>72</sup> Dies stellt eine Möglichkeit dar, einem mobilen Endgerät die Sensordaten zum Beispiel

---

<sup>68</sup>Vgl. Sch12, Folie 25.

<sup>69</sup>Sch12, Folie 25.

<sup>70</sup>Vgl. Sch12, Folie 25.

<sup>71</sup>GSR: Galvanic Skin Response

<sup>72</sup>beispielsweise: Gro18.



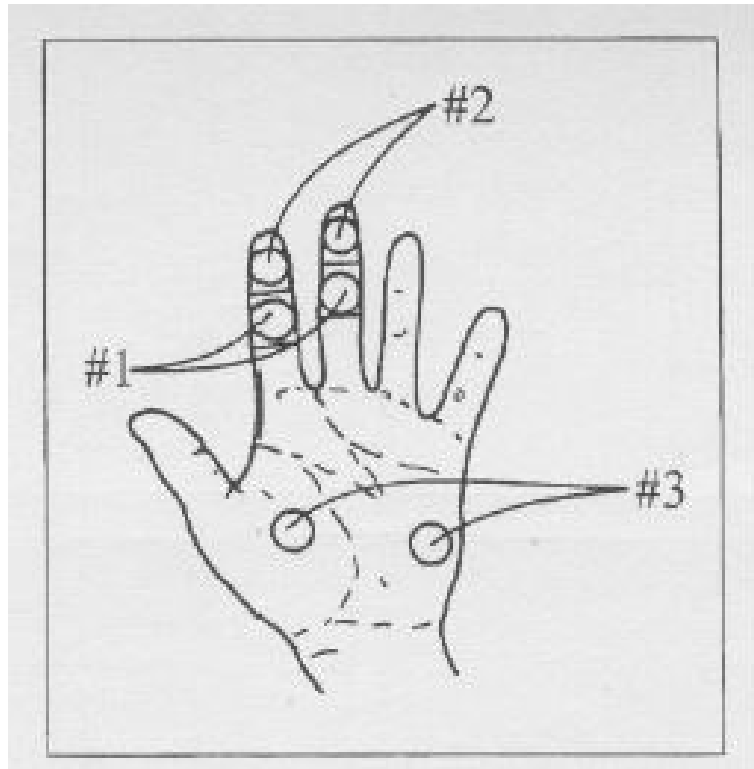


Abbildung 8: Positionsmöglichkeiten der EDA-Messung<sup>69</sup>

über Bluetooth oder WiFi zur Verfügung zu stellen. In Abbildung ? ist ein GSR-Sensor inklusive der Elektroden für die Fingerinnenseiten, der mit Arduinos kompatibel ist, abgebildet.

### 2.6.2 Stimmerkennung mit Smartphone-Mikrofon

TORBEN BRENNER

Grundsätzlich haben alle Smartphones ein Mikrofon, da dieses insbesondere für das Telefonieren gebraucht wird. Biermann schreibt in seinem Artikel außerdem davon, dass heutige Smartphones nicht nur ein Mikrofon, sondern gleich mehrere Mikrofone besitzen. Die zusätzlichen Mikrofone werden unter anderem dazu verwendet, Störgeräusche herauszufiltern<sup>74</sup>.

Da die Stimmerkennung sich im allgemeinen immer weiterentwickelt, zeigen so genannte *Vocal Computing* Schnittstellen. Dieser Begriff bezeichnet laut Ferdinand und Jetzke “die Möglichkeit, über Sprache mit Computern, mobilen und stationären Endgeräten sowie deren softwarebasierten Anwendungen zu interagieren”<sup>75</sup>. War es früher noch schwer möglich einzelne Wörter zu erkennen, gibt es mittlerweile Smartphone Anwendungen wie Apples “Siri” und Googles “Google Now”, “die nicht auf einzelne Befehle reduziert [sind], sondern ganze Sätze und den Kontext ihrer Äußerung erfassen [können]”<sup>76</sup>.

<sup>73</sup>Gro18.

<sup>74</sup>Vgl. Bie14, S.2 Mikrofon Abs.2.

<sup>75</sup>siehe Fer17, S.1 Z.1ff.

<sup>76</sup>siehe Fer17, S.1 Z.17ff.

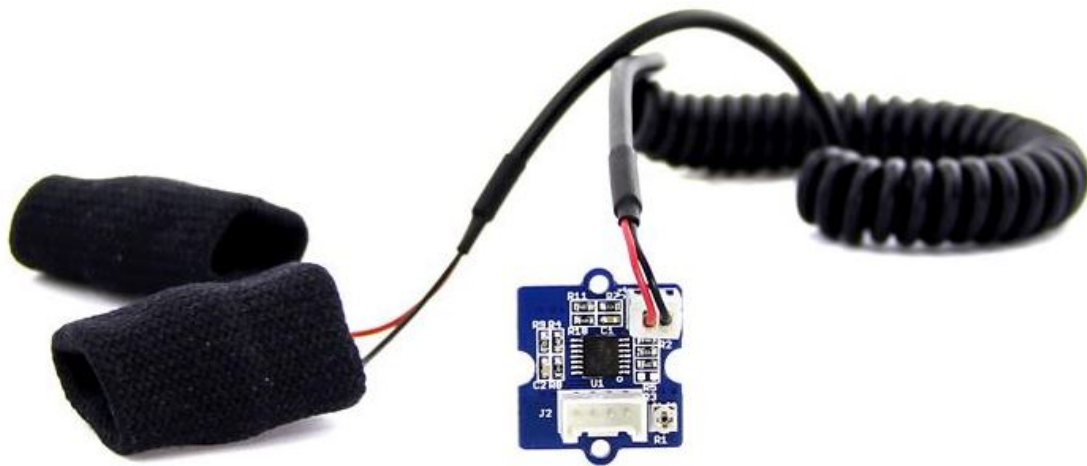


Abbildung 9: GSR-Sensor mit Finger-Elektroden<sup>73</sup>

Das auch in der Emotionserkennung über die Stimme ein Fortschritt erreicht wurde, zeigen Russische Forscher mit einem KI-Algorithmus<sup>77</sup>. Der Algorithmus versucht im Allgemeinen zwischen acht verschiedenen Basisemotionen zu unterscheiden. Dabei scheint es insbesondere bei der Unterscheidung zwischen Freude und Ärger Probleme zu haben. Diese zeichnen sich beide durch eine starke Erregung aus. Wie bereits bei der Beschreibung der Stimme als Emotionsindiz erwähnt, ist dies eines der Hauptmerkmale die sich über die Stimme erfassen lassen.

Das sich solche Anwendungen auch in Form einer Smartphone App umsetzen lassen, haben Ingenieure der University of Rochester gezeigt<sup>78</sup>. Diese untersuchen in ihrem Programm zwölf unterschiedliche Charakteristiken in gesprochenen Worten, um so zwischen sechs unterschiedlichen Emotionen unterscheiden zu können. Ein App Prototyp zeigt anscheinend schon je nach Stimme einen traurigen oder fröhlichen Smiley an.

### 2.6.3 Gesichtserkennung mit Smartphone-Kamera

TORBEN BRENNER

Zur Erfassung der Mimik, benötigt das Smartphone die Fähigkeit ein Gesicht zu erfassen und den Gesichtsausdruck zu erfassen. Hierfür eignet sich die Kamera des Smartphones. Diese war früher ausschließlich für das machen von Bildern gedacht. Mittlerweile werden aber die in Smartphones verbauten Kameras immer besser und können unter anderem für die Gesichtserkennung oder die Verfolgung der Augenbewegungen genutzt werden<sup>79</sup>. [Can01] beschreibt in seiner Arbeit ein Verfahren, um mit Kameras die menschliche Mimik automatisiert zu analysieren. Basierend auf seinen Methoden, kann ein System entwickelt werden, das aus der Mimik eines Menschen dessen Emotionen analysiert. Canzler

<sup>77</sup>Vgl. Ion17b.

<sup>78</sup>Vgl. Wed12.

<sup>79</sup>Vgl. Bie14, Seite 2 Kamera Abs. 1+2.

beschreibt in seiner Arbeit folgende vier Phasen der Analyse<sup>80</sup>:

1. **Gesichtsdetektion**
2. **Tracken charakteristischer Punkte**
3. **Extraktion von Mermalen**
4. **Gesichtsanalyse**

Die vier von Canzler ausgearbeiteten Phasen können bis auf wenige Anpassungen übernommen werden. Die Gesichtsdetektion wird mit Hilfe von verschiedenen Verfahren durchgeführt. Die formorientierten Verfahren suchen nach Kanten und Konturen welche dem menschlichen Gesicht entsprechen und farborientierten Verfahren die nach charakteristischen menschlichen Hautfarben suchen<sup>81</sup>. In der zweiten Phase wird anhand eines vordefinierten durchschnittlichen Gesichts eine Verfolgung von wichtigen Punkten durchgeführt. Basierend auf den geometrischen Eigenschaften des Gesichts können daraufhin mit Template Matching verschiedene Merkmale, wie z. Bsp. die Blickrichtung und Stirnfaltenbildung, extrahiert werden. In der letzten Phase, der Gesichtsanalyse, werden die vorher ermittelten Merkmale in sogenannte *Action Units* umgesetzt<sup>82</sup>. Diese wurden im *Facial Action Coding System* definiert. Sie beschreiben die kleinsten sichtbaren Einheiten von Muskulatur im Gesicht<sup>83</sup>. Die Action Units wurden bereits von verschiedenen Forschern in Zusammenhang mit den Emotionen gebracht. Problematisch ist aber laut [SK98], dass vollständig vordefinierte Gesichtsausdrücke selten genauso in der Realität vorkommen, weshalb es sinnvoller ist den Zusammenhang zwischen einzelnen AUs und bestimmten Emotionen zu untersuchen<sup>84</sup>. Kaiser & Scherer haben in ihrer Arbeit z. Bsp. das auftreten verschiedener AUs in Zusammenhang mit emotionalen Störungen wie Depressionen beschrieben<sup>85</sup>. Es gibt bereits mehrere Systeme, die es ermöglichen mit der Analyse des Gesichts Emotionen zu bestimmen. So gibt es vom Fraunhofer-Institut für Integrierte Schaltungen das SHORE System<sup>86</sup>. Dieses ermöglicht die Gesichtsanalyse ohne Internetanbindung. Dabei lassen sich sowohl das Geschlecht und Alter, als auch vier unterschiedliche Emotionen erkennen. Es soll außerdem auch auf mobilen Endgeräten funktionieren. Es gibt eine kostenlose Demo, die aber auf eine Laufzeit von 90 Tagen beschränkt ist, weshalb sie sich für unsere Zwecke nicht eignet<sup>87</sup>.

Da Smartphones sich unter anderem durch die Möglichkeit auszeichnen, sich unterwegs

---

<sup>80</sup>Vgl. Can01, S.2-5.

<sup>81</sup>Vgl. Can01, S.2 Z.14ff.

<sup>82</sup>Vgl. Can01, S.4 Z.17-21.

<sup>83</sup>Vgl. SK98, S.94 Z.9ff.

<sup>84</sup>Vgl. SK98, S.94 Z.16-30.

<sup>85</sup>Vgl. SK98, S.95 Table 6.3 Facial Action Units Predicted as Indicators of Selected Types of Affect Disorders.

<sup>86</sup>Vgl. FraB.

<sup>87</sup>Vgl. Fraa.

mit dem Internet zu verbinden ist auch die Nutzung einer Server Anwendung denkbar. Microsoft bietet z. Bsp. eine Schnittstelle für die Gesichtserkennung<sup>88</sup>. Diese bietet nicht nur die Möglichkeit, Personen zu erkennen und deren Alter zu bestimmen, sondern ermöglicht es auch die Emotionen einer Person zu schätzen. Die kostenlose Nutzung erlaubt 30.000 Transaktionen im Monat mit bis zu 20 Transaktionen in der Minute<sup>89</sup>.

#### 2.6.4 Pulsmessung mit Smartphone-Kamera

LUKAS SEEMANN

In den App-Stores werden heutzutage mehrere Apps angeboten, die angeben, eine Pulsmessung ohne zusätzliche Geräte zu ermöglichen.<sup>90</sup> Diese Apps nutzen lediglich die im Smartphone integrierte Kamera. Das Ergebnis wird deutlich genauer, wenn das Smartphone außerdem über ein LED-Blitzlicht verfügt.<sup>91</sup>

Die Vorgehensweise bei dieser Art der Pulsmessung ist nicht komplex. Der Smartphone-Nutzer muss dazu lediglich die oberste Fingerkuppe eines beliebigen Fingers auf die Kamera und falls vorhanden auch auf das Blitzlicht des Smartphones legen. Ist kein Blitzlicht vorhanden, ist es auch möglich, die Messung bei gutem Tageslicht durchzuführen.<sup>92</sup> Dies kann jedoch Auswirkungen auf die Genauigkeit des Ergebnisses haben. Über ein gewissen Zeitraum, der meistens im Sekunden- oder niedrigen Minutenbereich liegt, werden anschließend die Helligkeitsschwankungen in der Fingerkuppe des Nutzer aufgezeichnet.<sup>93</sup> Diese Helligkeitsschwankungen werden durch den Puls des Benutzer ausgelöst, da der Finger bei hoher Durchblutung nicht so lichtdurchlässig ist, wie bei niedriger Durchblutung. Aus diesen dokumentierten Schwankungen berechnen die Apps mithilfe von signalanalytischen Methoden den Puls des Benutzers.<sup>94</sup>

Diese Methoden sind bereits teilweise laut App-Hersteller-Angaben medizinisch anerkannt<sup>95</sup> und liefern unter guten Bedingungen dieselben Ergebnisse, die auch herkömmliche Blutdruckmessgeräte ermitteln würden.<sup>96</sup> Dieses Verfahren der Pulsmessung basiert auf der Pulsoxymetrie. Das Ziel der Pulsoxymetrie ist es, mithilfe von Lichtquellen die Sauerstoffsättigung und die Pulswelle des Probanden zu messen.<sup>97</sup> Als Messorte eignen sich hierfür die Ohren, Zehen oder die Finger, was bei der Smartphone-Messung ausgenutzt wurde.

*„Bei der Pulsoxymetrie werden von einer Lichtquelle 2 unterschiedliche Wellenlängen (im infraroten und roten Bereich) ausgesendet, die nach dem Durchleuchten des Gewebes (z.B.*

---

<sup>88</sup>Vgl. Mic18b.

<sup>89</sup>Vgl. Mic18a.

<sup>90</sup>Vorstellung von Pulsmessungs-Apps auf dem Markt: Vgl. Pet17.

<sup>91</sup>Vgl. Gil14.

<sup>92</sup>Vgl. Gil14.

<sup>93</sup>Vgl. Pre16.

<sup>94</sup>Vgl. Pre16.

<sup>95</sup>Vgl. Pre16.

<sup>96</sup>Vgl. Hen17.

<sup>97</sup>Vgl. Deu09, S. 74.

am Finger) von einem Sensor aufgefangen werden.“<sup>98</sup> Neben der hierbei durchgeführten Messung der Sauerstoffsättigung kann durch die Absorption des Lichts durch die Puls-  
welle über die insgesamt aufgefangene Lichtmenge eine Aussage über die Puls-  
welle des Probanden abgeleitet werden.<sup>99</sup> Obwohl die meisten Smartphones keine verschiedenen  
Lichtwellenlängen verwenden, sind bereits die Messungen mit dem Blitzlicht des Smart-  
phones für den Puls sehr aussagekräftig. Die unterschiedlichen Lichtwellenlängen sind  
vor allem für die Sauerstoffsättigung notwendig, deren Messung die meisten Apps nicht  
anbieten.

Des Weiteren gibt es noch eine weitere Möglichkeit, wie die Smartphone-Kamera zur  
Pulsbestimmung genutzt werden kann. Mithilfe der Eulerschen Videoverstärkung (*Eu-  
lerian Video Magnification*, kurz EVM), die am Massachusetts Institute of Technology  
(MIT) von einer Gruppe von Informatikern entwickelt wurde, können kleinste Bewegun-  
gen in Videomaterial verdeutlicht werden.<sup>100</sup> Mit einer Videoaufnahme des Gesichts eines  
Probanden kann so der Puls mit einer hohen Genauigkeit bestimmt werden. Durch die  
pulsierende Blutzirkulation ändert sich die Gesichtsfarbe einer Person ungefähr einmal  
pro Sekunde. Dies ist mit bloßem Auge nicht sichtbar, kann jedoch mit Hilfe von EVM  
verstärkt und somit deutlich sichtbar gemacht werden. Mithilfe dieser Farbtonschwan-  
kungen kann der Puls sogar in Echtzeit bestimmt werden.<sup>101</sup>

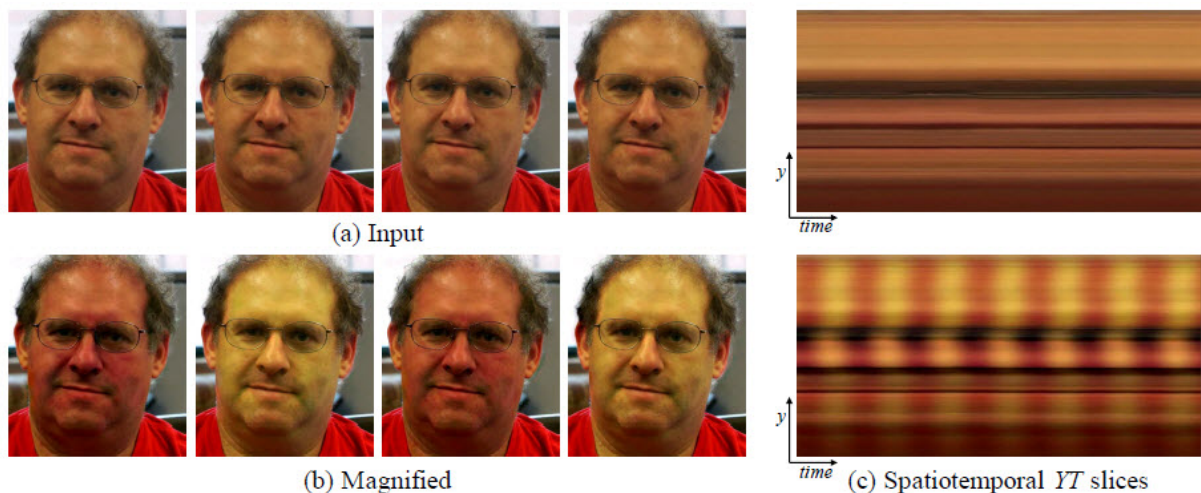


Abbildung 10: Positionsmöglichkeiten der EDA-Messung<sup>102</sup>

In Abbildung ? ist die Vorgehensweise bei der Pulsbestimmung gezeigt. Zunächst wird  
das aufgenommene Videomaterial in einzelne Frames unterteilt (siehe (a)). Die einzelnen

<sup>98</sup>Deu09, S. 74.

<sup>99</sup>Vgl. Deu09, S. 74.

<sup>100</sup>Vgl. Dam12.

<sup>101</sup>Vgl. Dam12.

<sup>102</sup>MIT15.

Frames werden dann mithilfe von EVM verstärkt, sodass die durch die Durchblutung veränderte Gesichtsfarbe deutlich wird (siehe (b)). Rot deutet dabei auf eine hohe Durchblutung hin, wohingegen eine blassere Farbe eine niedrige Durchblutung bedeutet. In (c) wird anschließend ein Graph angefertigt, der Ausschnitte des Gesichts im Verlauf der Zeit anzeigt. Bei nicht verstärktem Input sind keine Unterschiede sichtbar. Mit EVM kann jedoch deutlich ausgemacht werden, wie sich die Gesichtsfarbe im Verlauf der Zeit geändert hat. Auf Basis dieses Graphen kann anschließend der Puls bestimmt werden.<sup>103</sup> Da Smartphones heutzutage über sehr gute Kameras verfügen, kann die Nutzung von EVM eine gute Möglichkeit zur Pulsbestimmung darstellen.

### 2.6.5 Pulsmessung mit externen Sensoren

LUKAS SEEMANN

Neben der Pulsmessung über die Smartphone-Kamera gibt es auch die Möglichkeit, mit externen Sensoren den Puls zu messen. Hierbei gibt es wie auch bei den GSR-Sensor passende Module, die an einen Arduino-Mikrocontroller angeschlossen werden können. In Abbildung ? ist ein häufig verwendeter Puls-Sensor des Unternehmens World Famous Electronics llc. zu sehen, der mit einem Arduino verbunden werden kann.

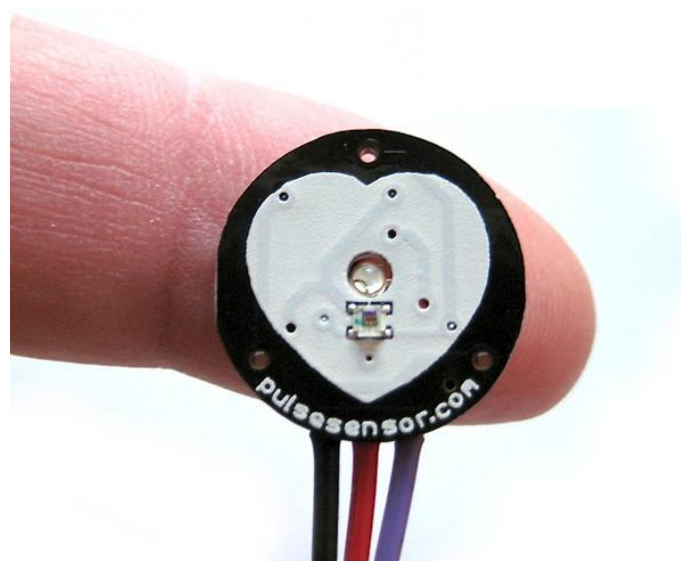


Abbildung 11: Puls-Sensor für einen Arduino<sup>104</sup>

Dieser Sensor kann entweder am Finger oder am Ohr des Probanden angebracht werden. Das Messverfahren basiert hierbei ebenfalls auf der Pulsoxymetrie, es handelt sich also um einen optischen Sensor. Bei Sensor handelt es sich um einen Photoplethysmograph (kurz PPG), der auf dieser Art und Weise häufig für medizinische Untersuchungen eingesetzt wird.<sup>105</sup> Einen PPG macht aus, dass ein zu untersuchendes Hautareal mit Infrarotlicht

---

<sup>103</sup>Vgl. MIT15.

<sup>104</sup>Wor18a.

<sup>105</sup>Vgl. Wor18b.

ausgesetzt wird und anschließend untersucht wird, wie viel Licht vom Areal absorbiert wurde.<sup>106</sup> Dies ist auch der Unterschied zur Smartphone-Kamera, bei der nur das Blitzlicht beziehungsweise das Tageslicht zur Messung und kein Infrarotlicht verwendet wird.

### 2.6.6 Analyse des Tippverhaltens bei Smartphone-Nutzung

Grundsätzlich benötigt man bei der Tippverhaltensanalyse keine zusätzliche Hardware, da sich Aspekte wie der Tipprhythmus, die Tippgeschwindigkeit und Tippfehler Softwareseitig analysieren lassen. Hierzu wird aber der Zugriff auf eingegebene Texte benötigt, weshalb es grundsätzlich nicht möglich ist auf einem Smartphone diese Daten zu erhalten. Das liegt daran, dass der Nutzer vor dem mitlesen seiner Passwörter geschützt werden soll. Das es dennoch möglich ist, auf dem Smartphone Informationen über das Tippverhalten eines Nutzers auszulesen, zeigt die schwedische Firma Behaviorsec. Diese Firma hat eine Software entwickelt, die bei der Eingabe einer Pin oder eines Passwortes die Tippgeschwindigkeit, die aufliegende Fläche der Fingerkuppe oder den Winkel in dem das Smartphone gehalten wird prüft<sup>107</sup>.

### 2.6.7 Gangerkennung mit dem Smartphone

TORBEN BRENNER

Zur Ermittlung der Gangart, gibt es mehrere Methoden um diese umzusetzen. Im folgenden sollen diese Methoden vorgestellt werden und geprüft werden, ob diese mit einem Smartphone umsetzbar sind.

Eine bereits in der Praxis auch eingesetzte Methode, ist die Verwendung von speziellen mit Sensoren ausgestatteten Bodenplatten<sup>108</sup>. [Jan06] untersuchen in ihrer Arbeit, die Möglichkeit mit solchen Bodenplatten auch Rückschlüsse auf Emotionen zu ermöglichen. Sie kommen zu dem Schluss, dass mit der Erkennung durch Bodenplatten es zumindest möglich ist zwischen einzelnen Emotionen wie Trauer und Wut zu unterscheiden<sup>109</sup>. Um diese Möglichkeit aber mit dem Smartphone umzusetzen müsste das Smartphone mit diesen Sensoren verbunden werden, z.Bsp. über Bluetooth oder NFC(Near Field Communication).

Dies ist aber nicht die einzige Möglichkeit den Gang zu erkennen. So kann laut Claudia Nickel z. Bsp. auch videobasiert eine Gangerkennung durchgeführt werden<sup>110</sup>. Diese wäre grundsätzlich auch mit dem Smartphone möglich, würde aber das dauerhafte aufnehmen eines Videos voraussetzen. Außerdem müsste in diesem Video eine Person dauerhaft erkennbar sein, was bei der normalen Bedienung eines Smartphones nicht ohne weiteres

---

<sup>106</sup>Vgl. Rab06, S. 38.

<sup>107</sup>Vgl. Hub14.

<sup>108</sup>Vgl. Bio18a, Praktische Anwendung.

<sup>109</sup>Vgl. Jan06, S.4 Diskussion.

<sup>110</sup>Vgl. Cla09, S.281 Z.9-15.

möglich ist. Deshalb stellt Nickel noch eine dritte Variante der Gangerkennung vor. Die Erkennung anhand eines Beschleunigungssensors. Diese Variante eignet sich laut Nickel besonders gut für die Erkennung in mobilen Geräten, vor allem da diese Art von Sensor in den meisten Geräten verbaut ist<sup>111</sup>. In ihrer Dissertation beschäftigt sich Nickel insbesondere mit der Gangerkennung durch diesen Sensor.

---

<sup>111</sup>Vgl. Cla09, S.281 Z.25-36.



### 3 Konzept

Hier wird ein Konzept mit Mock Ups und Architektur entstehen

#### 3.1 Priorisierung der Erfassungsmöglichkeiten LUKAS SEEMANN

Im Anschluss an die Auswahl der Erfassungsmöglichkeiten werden diese Möglichkeiten nun priorisiert. In der folgenden Tabelle (Tabelle 1) ist die Priorisierung abgebildet.

Priorität	Indiz	Sensor	Im Smartphone?
1	Gesicht	Kamera	JA
2	Hautleitfähigkeit	GSR/EDA-Sensor	NEIN
3	Stimme	Mikrofon	JA
4	Tippverhalten	Touchscreen	JA
5	Puls	Kamera	JA
6	Puls	Externe Sensoren	NEIN
7	Gangart	Accelerator	JA

Tabelle 1: Priorisierung der Erfassungsmöglichkeiten

In der ersten Spalte ist die Priorität dargestellt. Je niedriger die Zahl ist, desto höher ist die Erfassungsmöglichkeit priorisiert. Die Möglichkeiten werden in der Reihenfolge der hier dargestellten Priorisierung thematisiert und letzten Endes in den Prototyp der mobilen Applikation integriert, um Daten zu erfassen. Je nachdem wie viel Zeit die einzelnen Features benötigen, können mehr und mehr Möglichkeiten der Datenerfassung in die App eingebaut werden, wenn sie noch im Zeitrahmen der Studienarbeit umsetzbar sind. Bei den einzelnen Möglichkeiten werden das Indiz, anhand dessen Rückschlüsse auf eine Emotion gemacht werden kann, und ein Sensor, der Daten zum Indiz für die App erfassen soll, aufgelistet. In der letzten Spalte ist festgehalten, ob der benötigte Sensor in den meisten aktuellen Smartphones bereits enthalten ist oder nicht.

Die höchste Priorität hat die Erfassung des Gesichts mit der Smartphone-Kamera. Grund hierfür ist, dass die Gesichtserkennung sehr genaue Rückschlüsse auf die Emotionen zulässt. Die Ergebnisse liefern außerdem Hinweise für viele unterschiedliche Emotionen, aus welchem Grund die Gesichtserkennung eine gute Basis für die Emotionsbestimmung der mobilen Applikation darstellt. Zusätzliche Indizien können die Ergebnisse der Gesichtserkennung verstärken oder relativieren. Des Weiteren ist vorteilhaft, dass die Erfassungsmöglichkeit ohne externe Sensoren auskommt und lediglich die Smartphone-Kamera genutzt werden kann.

Die nächst höhere Priorität hat das Indiz der Hautleitfähigkeit, das mithilfe von GSR-

beziehungsweise EDA-Sensoren erfasst werden kann. Diese Art von Sensoren befinden sich wie bereits beschrieben nicht in handelsüblichen Smartphones, weshalb man hierzu einen Arduino verwenden muss, der anschließend Daten an die App weiterleitet. Obwohl dies ein Nachteil ist, wurde diese Erfassungsmöglichkeit so hoch priorisiert, da dies ein Bereich in der App-Entwicklung ist, der nur selten thematisiert wird. Dieses Potenzial ist zusammen mit Aussagekraft der Aktivierung zu bestimmten Emotionen der Hauptgrund für die hohe Priorisierung. Jedoch muss die Messung der Hautleitfähigkeit mit einem anderen Erfassungstest kombiniert werden, da die Messung alleine keine eindeutigen Anzeichen für einzelne Emotionen zurückliefert, sondern für mehrere.

An dritter Stelle steht die Stimmerkennung mithilfe des Smartphone-Mikrofons. Obwohl aus der Stimme auch viele Anzeichen für Emotionen herausgelesen werden können und ein Mikrofon in jedem Smartphone integriert ist, hat die Stimmerkennung einen Nachteil. Der Benutzer müsste dazu über einen längeren Zeitraum in das Mikrofon sprechen. Da er das für eine Durchführung der Messung zwingend machen muss, besteht auch die Gefahr, dass die Stimme des User nicht natürlich ist, sondern aufgesetzt ist, da er sich über die Durchführung des Tests bewusst ist. Da mit der Gesichtserkennung außerdem bereits eine umfangreiche Basis für die Emotionsbestimmung vorhanden ist, erfolgt die Priorisierung auf Platz 3.

Auf dem vierten Platz folgt die Erfassung des Tippverhaltens mit dem Touchscreen beziehungsweise mit der eingeblendeten Tastatur des Smartphones. Hier ist der selbe Nachteil vorhanden wie bei der Stimmerkennung. Der User muss über eine längere Zeit Eingaben über den Touchscreen vornehmen und er ist sich nach Start des Tests bewusst, dass seine Eingaben analysiert werden. Deshalb ist das Ergebnis des Tippverhaltens-Tests einfach manipulierbar. Außerdem ist es auch bezüglich der Privatsphäre des Users problematisch, seine eingegebenen Texte zu analysieren. Trotzdem können bei ehrlicher, nicht manipulierter Benutzung des Tests interessante Ergebnisse für die Emotionsbestimmung gewonnen werden.

Auf Platz 5 und 6 folgt die Pulsmessung. Da die meisten Erkenntnisse, die mit der Messung des Puls bezüglich Emotionen gewonnen werden können, bereits mit der Hautleitfähigkeits-Messung abgedeckt werden, sind diese beiden Möglichkeiten niedrig priorisiert. Die Pulsmessungen können im besten Fall die Ergebnisse des GSR-Sensors bekräftigen, liefern aber keine neuen Erkenntnisse. Die Messung mit der integrierten Smartphone-Kamera wird trotz der vermeintlich geringeren Genauigkeit höher priorisiert als die Messung mit dem externen Arduino-Sensor, da dort keine zusätzliche Hardware verwendet werden muss.

Auf den letzten Platz liegt die Messung der Gangart. Obwohl es sich hierbei auch um eine Messung biometrischer Daten handelt, können hieraus nur sehr wenige Anzeichen für Emotionen herausgelesen werden. Von allen beschriebenen Methoden liefert diese Möglichkeit den geringsten Nutzwert für die Zwecke der mobilen Applikation. Aus die-

sem Grund hat die Implementierung dieser Messung die geringste Priorität.

## 3.2 Übertragung biometrischer Daten in Indikatorscores

LUKAS SEEMANN

Die von den Smartphone-internen und externen Sensoren zurückgelieferten Daten müssen in der App einheitlich verarbeitet werden. Die zurückgelieferten Daten sind unterschiedlich und meistens selbst nicht aussagend. Mit dem im Theorieteil erklärten Hintergrundwissen ist es jedoch möglich, die Daten zu interpretieren.

Im Umfeld der App soll diese Interpretation mithilfe von Indikatoren geschehen. Ein Indikator ist in diesem Kontext ein Anzeichen, das für die Emotionsbestimmung herangezogen werden kann. Es existieren einerseits Indikatoren, die sich nur für die Bestimmung einer Emotion eignen, und andererseits auch Indikatoren, die für mehrere Emotionen ausschlaggebend sind. Folgende Indikatoren sind für die Umsetzung geplant:

- activation,
- happyIndicator,
- sadIndicator,
- angryIndicator und
- suprisedIndicator.

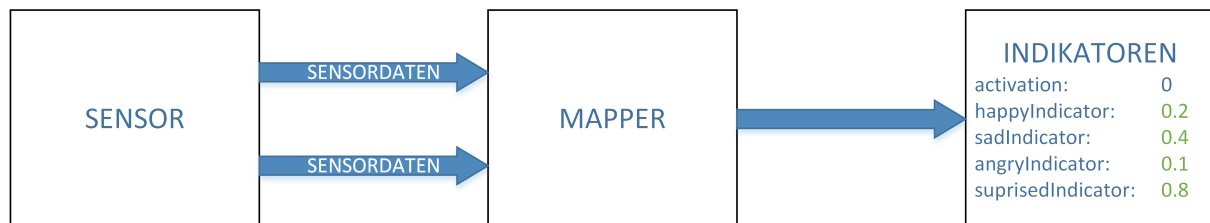


Abbildung 12: Ablauf der Erstellung von Indikatorscores

In Abbildung ? ist der Ablauf der Bestimmung von Indikatorscores abgebildet. Unabhängig davon, welcher Sensor verwendet wird, seine Daten müssen immer auf diese Indikatoren abgebildet werden. Für jeden Sensor muss dabei ein Bereich bestimmt werden, wann immer eine neue Auswertung der Sensordaten zu den sogenannten Indikatorscores geschieht. Eine Auswertung kann generell dann durchgeführt werden, wenn genug Daten vorhanden sind, die aussagekräftig für die Indikatoren sind. Dies ist von Sensor zu Sensor unterschiedlich und muss dementsprechend berücksichtigt werden.

Bei einer Auswertung werden die einzelnen Indikatoren mit Scores von null bis eins versehen, sodass man pro Auswertung der Sensordaten mehrere Indikatorscores erhält. Die Logik für das Setzen der Indikatorscores muss mithilfe von Mappern umgesetzt werden. Für jeden Sensor muss dafür ein individueller Mapper existieren. Die Mapper werden

immer dann aufgerufen, wenn genügend Daten des Sensors vorhanden sind.

### 3.3 Auswertung der Indikatorscores zur Emotionsbestimmung

LUKAS SEEMANN

Aus den angesammelten Indikatorscores muss nun entschieden werden, welche Emotion am wahrscheinlichsten beim Benutzer der App vorliegt. Diese Auswertung der Indikatorscores zu konkreten Emotionen erfolgt mit einer Menge von Kausalitätsregeln. Als Ergebnis wird pro Emotion ein Emotionscore erwartet, der angibt, inwiefern die Emotion beim User zum getesteten Zeitpunkt vorliegt. Anschließend muss geplant werden, in welchem zeitlichen Abstand ein neue Emotionscores angelegt werden.

#### 3.3.1 Kausalitätsregeln

LUKAS SEEMANN

Eine Kausalitätsregel erwartet eine Menge von Indikatorscores als Eingabe und wendet basierend auf den Indikatorscores Effekte auf die vorhandenen Emotionscores an. Die Indikatoren werden also je nach ihren Werten in verschiedene Emotionen übersetzt.

Eine Kausalitätsregel besteht immer aus einer Bedingung und einer Score-Transformation. Eine Bedingung betrifft einen oder maximal zwei Indikatorscores und legt fest, wann die Score-Transformation der Kausalitätsregel ausgeführt wird. Die Score-Transformation beschreibt Effekte, die auf die Emotionscores angewandt werden. Beispielsweise würde einer hoher Score des Stress-Indikators zu positiven Effekten für die Emotionen happy, angry und suprised führen. In Abbildung ? ist der Ablauf der Ausführung einer solchen Kausalitätsregeln gezeigt. Die Formulierung von realitätsgetreuen Kausalitätsregeln ist ausschlaggebend dafür, wie präzise die Anwendung letzten Endes die Emotionen bestimmen kann.

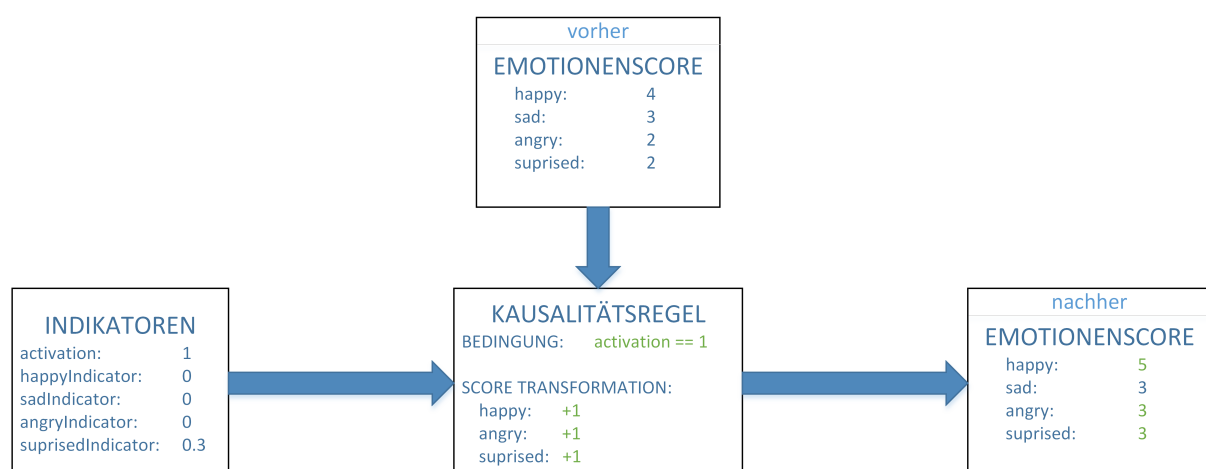


Abbildung 13: Ablauf der Ausführung von Kausalitätsregeln

Es wurde bereits beschrieben, wie aus Sensordaten Indikatorscores gewonnen werden und wie wiederum aus diesen mithilfe der Kausalitätsregeln Emotionscores ermittelt werden. Nun muss bestimmt werden, wie oft jeweils neue Emotionscores erstellt und eine Anwendung der Kausalitätsregeln durchgeführt wird.

Um später einen zeitlichen Ablauf der Emotionen des Nutzers anzeigen zu können, muss ein minimaler Zeitbereich bestimmt werden, für den die Auswertung in Emotionscores durchgeführt werden kann. Angedacht ist hierbei, dass der User alle 10 Sekunden eine neue Auswertung erhält. Folglich bedeutet das, dass alle Indicatorscores die in einem 10-Sekunden-Intervall registriert wurden, zusammengefasst werden und für diese dann alle Kausalitätsregeln durchgeführt werden. Dies bedeutet das alle 10 Sekunden eine neue Emotionscores erzeugt werden und anhand der im Intervall vorhandenen Indicatorscores verändert werden. So wird dem User ermöglicht, alle 10 Sekunden eine Bestimmung seiner Emotion in diesem Intervall zu erhalten.

## 3.4 Ionic Framework

In diesem Kapitel wird das Ionic Framework, das für die Umsetzung der mobilen Applikation verwendet wird, vorgestellt und die Gründe für die Verwendung aufgezeigt.

### 3.4.1 Aufbau und Einsatz des Frameworks

Ionic ist ein unter der MIT License stehendes Open-Source-Framework, das zur Entwicklung von plattformübergreifenden, mobilen Applikationen dient. Mit Ionic entwickelte Apps sind damit unter anderem auf Endgeräten lauffähig, die die Betriebssysteme Android, iOS und Windows Phone benutzen.<sup>112</sup>



Abbildung 14: Ionic Framework - Logo<sup>113</sup>

Aktuell befindet sich das Framework in der Version 3.9.2<sup>114</sup> und befindet sich in stetiger Weiterentwicklung. Das Ionic Framework basiert wiederum auf Angular, einem Fra-

---

<sup>112</sup>Ion18d.

<sup>113</sup>Wik18.

<sup>114</sup>Vgl. Ion17a.

mework für die Entwicklung von Web-Applikationen. Dementsprechend nutzen Ionic-Anwendungen in der Web-Entwicklung etablierte Technologien wie HTML 5, CSS und JavaScript.<sup>115</sup> Wie auch im Angular Framework, wird auch die Programmiersprache TypeScript verwendet, die auf JavaScript aufbaut, sich in der Syntax sehr stark mit JavaScript ähnelt und zusätzliche Optionen zur Typisierung von Variablen oder Funktionen anbietet.<sup>116</sup>

Ionic-Anwendungen sind im Wesentlichen normale Webanwendungen, die von jedem JavaScript-fähigen Browser ausgeführt werden können. Während mithilfe von Ionic das Frontend der Anwendung festgelegt wird, kann anschließend mit Apache Cordova die Plattformunabhängigkeit umgesetzt werden. Apache Cordova bewirkt, dass sich die Webanwendungen wie native Android-, iOS- oder Windows Phone-Applikationen anfühlen. Egal auf welcher Plattform die Ionic-Anwendung installiert wird, es wird die selbe Code-Basis verwendet. Diese wird dann vor dem Installieren von Cordova so angepasst, dass sie auf den Endgeräten ausgeführt werden können.<sup>117</sup>

### 3.4.2 Gründe für die Verwendung

LUKAS SEEMANN

Das Ionic Framework bietet eine Möglichkeit mit nur einer Code-Basis, eine plattformübergreifende Applikation zu erstellen. Dies erspart einiges an Entwicklungsaufwand, da nicht jede Plattform einzeln entwickelt werden muss. Außerdem wird einem Großteil der Smartphone-Nutzer die Nutzung der erstellten App ermöglicht und ist nicht nur für Android- oder Apple-Nutzer beschränkt.

Dadurch dass Applikationen des Ionic Frameworks eigenständige Webanwendungen sind, können diese problemlos von allen Browsern interpretiert und ausgeführt. Dies ist ein großer Vorteil für das Debuggen von Code, da hierfür kein Emulator oder Endgerät verwendet werden muss. Stattdessen kann die Anwendung im Browser ausgeführt und debuggt werden.

Außerdem wurden bereits in anderen Projekten und im privaten Bereiche Erfahrungen mit dem Ionic Framework gemacht, sodass keine Einarbeitung bei den Entwicklungsarbeiten notwendig ist. Dies spart wiederum Zeit, die effektiv für das Entwickeln genutzt werden kann.

## 3.5 Architektur der mobilen Applikation

In diesem Kapitel wird die Architektur der mobilen Applikation beschrieben. Die Architektur wird in Front- und Backend der App unterteilt.

---

<sup>115</sup>Vgl. Ion18b.

<sup>116</sup>Vgl. Til17.

<sup>117</sup>Ion18c.

Zunächst wird explizit auf das Backend beziehungsweise die Hintergrundlogik der App eingegangen. Für das oben beschriebene Konzept bietet sich eine Umsetzung mit Javascript Klassen an, aus welchem Grund ein UML-Diagramm angefertigt werden kann. Dieses ist in Abbildung 15 gezeigt.

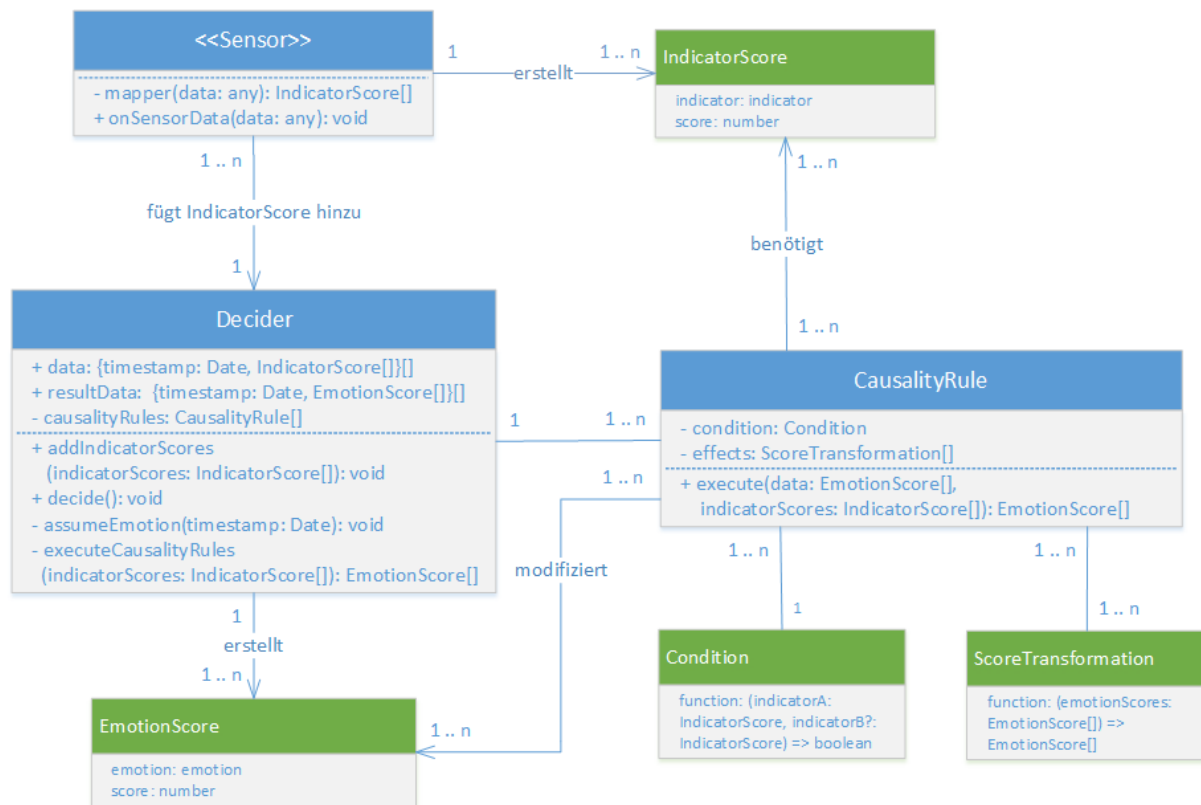


Abbildung 15: Architektur der App

Die Klassen sind in der Abbildung blau dargestellt. Typescript bietet die Möglichkeit, eigene Typen zu definieren. Diese Typen verfügen über Attribute, jedoch keine Methoden und sind in der Abbildung grün dargestellt.

Die App verfügt über eine abstrakte Klasse Sensor, die zwei Methoden definiert. Jeder Sensor, der später Daten für die App liefern soll, muss diese abstrakte Sensor-Klasse erweitern. Dabei muss jeder Sensor die mapper-Funktion überschreiben, die die Sensordaten in mehrere IndicatorScores umwandelt. Auf diese Weise muss jeder Sensor seinen individuellen Mapper implementieren. Der IndicatorScore ist als Typ definiert worden, der einerseits einen Indicator und einen Score als Attribut hat. Indicator wiederum ist ein eigener Typ, der nur die definierten Indikatoren (stress, happyIndicator, sadIndicator, angryIndicator, suprisedIndicator) als String zulässt. Da die Implementierung des IndicatorScore-Typen sehr trivial ist, wurde sie aus Übersichtsgründen nicht in die Abbildung aufgenommen. Die onSensorData-Funktion ist bereits vordefiniert und muss dementsprechend nicht

überschrieben werden. Diese Funktion führt lediglich den Mapper aus, wenn genug Sensordaten für eine Auswertung vorhanden sind, und fügt die resultierenden IndicatorScores zum Decider hinzu.

Die Decider-Klasse ist der Hauptbestandteil der App, da sie die Bestimmung der Emotion durchführt. Die Decider-Klasse ist als Singleton anzusehen, da es pro laufende Anwendung nur einen Decider geben kann. Der Decider verfügt über ein data-Array, das allen Input der Sensoren enthält. Konkret handelt es sich hierbei um die IndicatorScores, die die Sensoren ermittelt haben, erweitert mit einem Timestamp für den Zeitpunkt der Ankunft. Nach und nach wird dieses Array mit den IndicatorScores der verschiedenen Sensoren gefüllt. Dies wird mit der Funktion `addIndicatorScores()` realisiert, die von den Sensoren mit der `onSensorData`-Funktion aufgerufen wird.

Der Decider verfügt außerdem über ein Array von `CausalityRules`. Die `CausalityRule`-Klasse ist die Implementierung der oben beschriebenen Kausalitätsregel. Eine `CausalityRule` besitzt immer eine `Condition` (Bedingung) und eine Menge von `ScoreTransformations`.

Eine `Condition` wurde als eigener Typ definiert und ist eine Funktionen, die einen `IndicatorScore` (optional auch zwei) als Input hat und auf einen Boolean-Wert abbildet. So kann eine Bedingung realisiert werden, die je nach Erfüllung `true` oder `false` zurückgibt. Eine `ScoreTransformation` ist ebenfalls ein eigener Typ, der eine Funktion definiert. Diese Funktion erhält als Input die vorhandenen `Emotionscores`, wendet auf diesen Effekte an und gibt die modifizierten `Emotionscores` zurück.

Die `execute`-Funktion der `CausalityRule` führt alle hinterlegten `ScoreTransformations` aus, wenn die `Condition` von den `IndicatorScores` erfüllt worden ist.

Um die Decider-Klasse weiter zu verstehen, muss zunächst der Typ `EmotionScore` erklärt werden. Dieser ist ähnlich wie der `IndicatorScore` aufgebaut. Zunächst wird eine `Emotion` angegeben, die dem Typ `Emotion` entsprechen muss. Der String muss dementsprechend mit den definierten Emotionen (`happy`, `sad`, `angry`, `surprised`) übereinstimmen. Anschließend folgt eine Nummer, die den Score angibt.

Die Decider hat ein `resultData`-Array, das mehrere `Emotionscores` enthält und die Timestamp der `IndicatorScores`, aus denen die `Emotionscores` erzeugt wurden. Dieses Array wird mithilfe der `decide`-Funktion gefüllt. Wenn die `decide`-Funktion ausgeführt wird, wird anschließend für alle im data-Array vorhandenen Timestamps die `assumeEmotion()`-Funktion ausgeführt. Als Parameter erhält diese Methode eine Timestamp und filtert alle `IndicatorScores` aus dem data-Array zu diesem Zeitpunkt. Für alle auf diese Weise gefilterten `IndicatorScores` wird anschließend die `executeCausalityRules()`-Funktion ausgeführt. Diese Funktion wiederum iteriert über alle vorhandenen `CausalityRules` und führt diese auf Basis der `IndicatorScores` aus. So erhält man am Ende `Emotionscores`.



Diese erhalten die Timestamp, die bei `assumeEmotion()` mitgegeben wurde und werden in das `resultData`-Array gespeichert. Hat man die `assumeEmotion`-Funktion für alle Timestamps ausgeführt, ist das `result`-Array komplett gefüllt und die Emotionsbestimmung abgeschlossen.

### 3.5.2 Frontend

LUKAS SEEMANN

Nachdem bereits das Backend beschrieben wurde, wird nun das Frontend der App thematisiert. Unter dem Frontend werden die Screens der App verstanden, die der User bei Nutzung der App aufrufen kann. Diese Screens werden bei Ionic Pages genannt und bestehen meist aus folgenden Bestandteilen:

- Eine Typescript-Datei (.ts),
- eine HTML-Datei (.html) und
- eine CSS-Datei (.scss).

In der Typescript-Datei wird die Anwendungslogik umgesetzt, außerdem erfolgt dort der Zugriff auf die Backendkomponenten wie zum Beispiel den Decider, die durch das Ionic Framework als sogenannte Provider bereitgestellt werden. Die HTML-Datei beschreibt den Aufbau der GUI der Page, wohingegen die CSS-Datei das Design der Page beschreibt. In Abbildung ? ist zu sehen, über welche Pages die App verfügen soll, welchen Zweck diese haben und wie sie miteinander in Verbindung stehen.

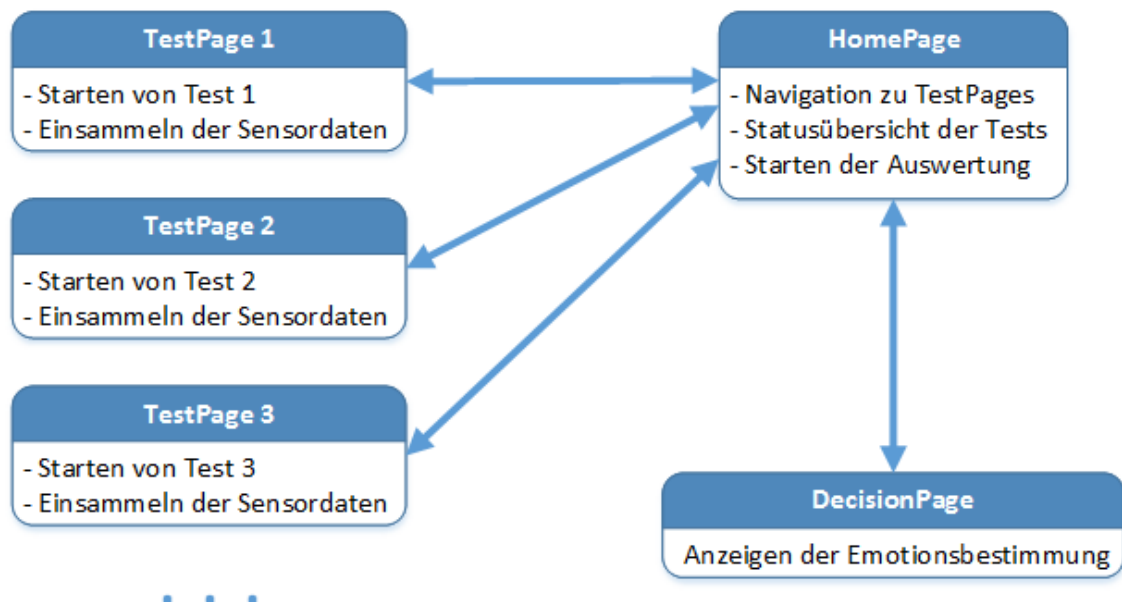


Abbildung 16: Geplantes Frontend der App

Für die App sollen nach und nach sogenannte Emotionstest entwickelt werden. Unter einem Emotionstest versteht man im Kontext dieses Projekts die Messung eines Sensors, die Rückschlüsse auf die Emotion zulassen. Ein Emotionstest liefert aus Backend-Sicht

immer eine Menge von Indikatorscores zurück. Pro Emotionstest muss eine Page der App angelegt werden, auf der der Test gestartet werden kann und so die Sensormessdaten eingesammelt werden können.

Zwangsläufig verfügt die App über eine HomePage, auf die der User beim Starten der App gelangt. Über diesen Screen kann der User zu den einzelnen TestPages gelangen, um die Tests durchzuführen. Von den TestPages kann der User entweder durch Abschließen der Tests oder durch manuelles Navigieren und Unterbrechen des Test zurück zum Homescreen gelangen. Um dem User anzuzeigen, welche Test er schon durchgeführt hat, wird außerdem eine Statusübersicht zu den Tests auf der HomePage zu sehen sein. Des Weiteren muss der User die Möglichkeit haben, die Auswertung der Indicatorscores zu Emotionen starten zu können. Wählt der User diese Option, werden alle Indicatorscores, die sich zu diesem Zeitpunkt im data-Array des Deciders befinden für die Auswertung verwendet.

Für das Ergebnis der Auswertung und somit die Emotionsbestimmung muss ebenfalls eine eigene Page existieren, worauf der User nach Abschluss der Auswertung gelangt. Der User kann von diesem Screen wieder zurück zum Homescreen gelangen, um erneut Emotionstests zu starten.

### **3.5.3 Vor- und Nachteile der Architektur**

LUKAS SEEMANN

In diesem Abschnitt werden die Vor- und Nachteile der vorgestellten Architektur gegenübergestellt werden.

Ein Vorteil der Architektur ist es, dass beliebige viele Sensoren und somit Emotionstest hinzugefügt werden können, ohne dass das Backend abgeändert werden müsste. Neue Sensoren müssen lediglich, dass Sensor-Interface implementieren und folglich die mapper-Funktion überschreiben. So kann die App in Zukunft beliebig erweitert werden.

Ein weiterer Vorteil ist, dass die Auswertung der IndicatorScores zu Emotionscores zentral über die Kausalitätsregeln verwaltet werden kann, deren Aufbau bereits vorgegeben ist. Da die Kausalitätsregeln ausschlaggebend für die Emotionsbestimmung sind, müssen diese bei Bedarf angepasst werden, wenn Endergebnisse der Emotionsbestimmung nicht passend sind. Diese Anpassung kann zentral im Array der Kausalitätsregeln geschehen, und es müssen keine Funktionen umgeschrieben werden oder die Architektur abgeändert werden.

Des Weiteren hat die Aufteilung in IndicatorScores und Emotionscores den Vorteil, dass im gesamten Backend eine einheitliche Datengrundlage vorliegt. Da verschiedene Sensoren ganz unterschiedliche Daten zurückliefern, müssen diese vereinheitlicht werden. Dies wurde mithilfe der IndicatorScores und der EmotionScores umgesetzt. Alle Sensoren, die in Zukunft hinzugefügt werden, müssen sich an die Datenstruktur dieser Scores anpassen. Dies wird durch die Mapper und Kausalitätsregeln sichergestellt.

Ein Nachteil der Architektur ist, dass eine Anpassung der IndicatorScores und der Emotionscores Änderungen an vielen Stellen hervorrufen würde. Will man beispielsweise einen neuen Indikator *disgustIndicator* und eine neue Emotion *disgust* hinzufügen, müssen zunächst alle Funktionen des Deciders angepasst werden, damit diese neue Emotion aufgenommen werden kann. Des Weiteren müssen gegebenenfalls viele vorhandenen Mapper und Kausalitätsregeln angepasst werden, da diese den neuen Indikator beziehungsweise die neue Emotion nicht berücksichtigt haben.

### 3.6 MockUps

TORBEN BRENNER

Bevor wir mit der Arbeit an dem Prototypen begonnen haben, diskutierten wir zuerst das Design der einzelnen Komponenten der Anwendung. Wir einigten uns darauf, dass die Übersicht über alle Teile des Prototypen auf der Hauptseite des Startbildschirms sein soll (siehe 3.6). Wie auf der Grafik zu sehen ist, sind für die fertige Anwendung 5 verschie-



Abbildung 17: Mockup - Homescreen(links) und Menü(rechts)

dene Arten von Tests vorgesehen. Die dazugehörigen Anwendungsabschnitte sollen von der Startseite, durch anklicken des Testnamens aufgerufen werden können und dann ausgeführt werden können. Durch das Berühren des Auswertungsknopfes soll es möglich sein, eine Analyse der verschiedenen Testergebnisse durchzuführen. Diese gibt dann als Ergebnis eine Übersicht heraus, welche Emotionen wie wahrscheinlich sind. Eine Übersicht der bereits durchgeführten Auswertungen soll über das Menü (siehe 3.6 rechts) erreichbar sein. In der Anleitung(ebenfalls über das Menü erreichbar) sollen sich Beschreibungen zur Durchführung der einzelnen Tests befinden.

Im folgenden soll neben den einzelnen Test Seiten, auch die Auswertungsseite genauer beschrieben werden.

Der Test der Hautleitfähigkeit benötigt die Verbindung mit dem Arduino Microcontroller. Deshalb muss dem Nutzer die Möglichkeit geboten werden, sich mit diesem über die Software möglichst einfach zu verbinden. Hierzu ist eine Seite für die Verbindung vorgesehen<sup>118</sup>. Auf dieser Seite hat der Nutzer die Möglichkeit, nach verfügbaren

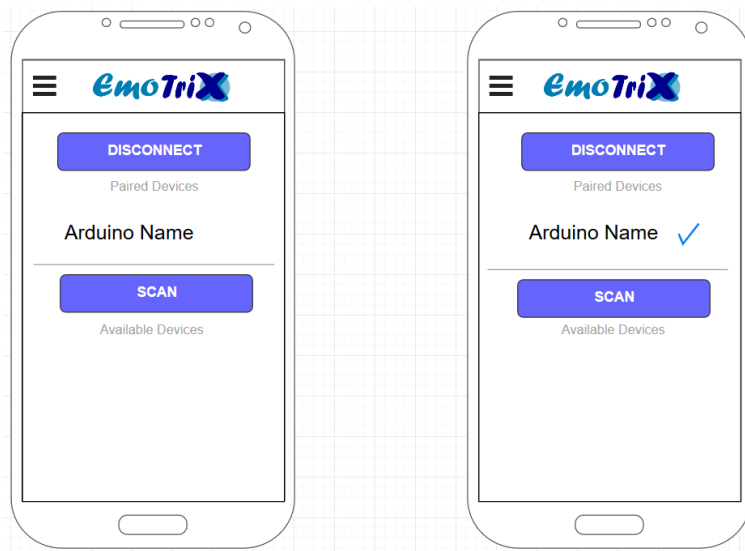


Abbildung 18: Mockup - Arduino verbinden(links nicht verbunden, rechts verbunden)

Geräten zu suchen. Da der Arduino mittels Bluetooth verbunden wird, werden in der Liste der verfügbaren Geräte auch andere Bluetoothgeräte angezeigt. Bei der Auswahl eines Gerätes wird versucht eine Verbindung mit diesem aufzubauen. Ist dieser Versuch erfolgreich, wird der Name des Gerätes zusammen mit einem Hacken dargestellt. Beim öffnen der Seite ist es ebenfalls möglich, dass der Name des Arduino bereits angezeigt wird. In diesem Fall soll es möglich sein, den Namen zu berühren um eine Verbindung aufzubauen. Gelingt diese wird ebenfalls ein Hacken hinter den Namen gesetzt.

Wo die Verbindung mit dem genau durchgeführt wird, ist noch nicht festgelegt. Denkbar wäre zum einen eine Settingsseite von der aus diese Einstellung erreichbar wäre, zum anderen ist es auch möglich, dass diese Seite angezeigt wird sobald der Test aufgerufen wird.

Der Test ist folgendermaßen aufgebaut. Der Nutzer schließt sich extern an den Hautleitfähigkeitssensor an. Hier sollte die Anleitung genauere Details enthalten wie das zu bewerkstelligen ist. Wenn der Nutzer an den Hautleitfähigkeitssensor angeschlossen ist und mit dem Arduino verbunden ist, kann der Test gestartet werden. Die Messungen beginnen nach dem drücken des Startknopfes. Danach sendet der Sensor über Bluetooth die Daten, welche in Form eines Graphen auf der Oberfläche dargestellt werden<sup>119</sup>. Auf

<sup>118</sup>zu sehen auf der Abbildung Mockup - Arduino verbinden

<sup>119</sup>siehe Abbildung *Mockup - Hautleitfähigkeit Test* auf S.39

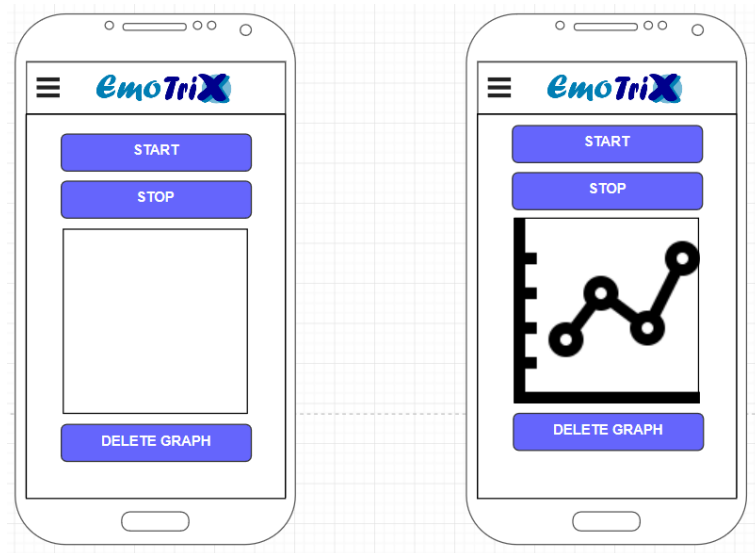


Abbildung 19: Mockup - Hautleitfähigkeit Test starten(links) und Test Ergebnis(rechts)

der Seite direkt werden keine Auswertungen der Emotionen dargestellt, sondern nur die Daten des Sensors. Durch betätigen des Stopknopfes, soll die Messung gestopt werden. Mit dem Knopf “DELETE GRAPH” wird der gezeichnete Graph zurückgesetzt, aber nicht die bisher erfassten Daten.

### 3.6.2 Tippverhalten

TORBEN BRENNER

Beim Test des Tippverhaltens, ist es angedacht das der Nutzer mit einem ChatBot kommuniziert und die Anwendung im Hintergrund bei Nutzereingaben das Tippverhalten überprüft. In wie fern die Kommunikation mit dem ChatBot abläuft ist noch nicht ganz klar. Es wäre zum denkbar, dass der ChatBot einfach nur einen Dienst zu Verfügung stellt und somit wenig Interaktion bietet. Interessant wäre auch dass vorbereiten eines bestimmten Gesprächs in dem der Nutzer auch von dem ChatBot provoziert werden könnte. Die Oberfläche für die Interaktion mit dem ChatBot soll einem normalen Chat ähneln, d.h. Nachrichten des ChatBots und Antworten des Nutzers werden angezeigt. Der Nutzer bekommt außerdem eine Bildschirmstatatur angezeigt, auf der er seine Antworten eingeben kann.

Ein Alternative zu diesem Testszenario war, dass der Nutzer einen Zeitraum angibt in dem die App seine Eingaben überwachen darf. Dieser Fall ist aber nicht umsetzbar, da die Betriebssystem natürlich verhindern wollen, dass Passwörter des Nutzers mit gelesen werden können. Deshalb soll das erste Szenario umgesetzt werden.



Abbildung 20: Mockup - Tippverhalten Test starten(links) und Test ergebnis(rechts)

### 3.6.3 Gesichtserkennung

TORBEN BRENNER

Bei der Gesichtserkennung wird der Nutzer von der Anwendung dazu aufgefordert, die Kamera zu starten und daraufhin ein Bild mit dieser zu machen. Die Anwendung soll dem Nutzer daraufhin die Aufnahme anzeigen und es ihm ermöglichen das er eine neue Aufnahme machen kann. Wenn der Nutzer eine Aufnahme hat die er analysieren lassen möchte, soll die Anwendung mit Hilfe eines "Send Picture" Knopfes die Möglichkeit bieten, dass Bild analysieren zu lassen.



Abbildung 21: Mockup - Gesichtserkennung vor Senden des Bilds(links) und Testergebnis(rechts)

Während der Analyse soll dem Nutzer durch einen Ladebalken signalisiert werden, dass

die Daten gerade verarbeitet werden. Auf der Ergebnisseite soll neben dem analysierten Bild das Ergebnis angezeigt werden und außerdem die Möglichkeit bestehen ein neues Bild zu machen oder das gleiche Bild erneut zu analysieren.

### 3.6.4 Auswertung

TORBEN BRENNER

Durch betätigen des Auswertungskopfes auf der Startseite der Anwendung, sollen die bisher erfassten Daten analysiert werden. Das Ergebnis dieser Analyse soll dem Nutzer auf der Auswertungsseite dargestellt werden. Diese besteht aus einem Kreisdiagramm und einem Zeitstrahl.

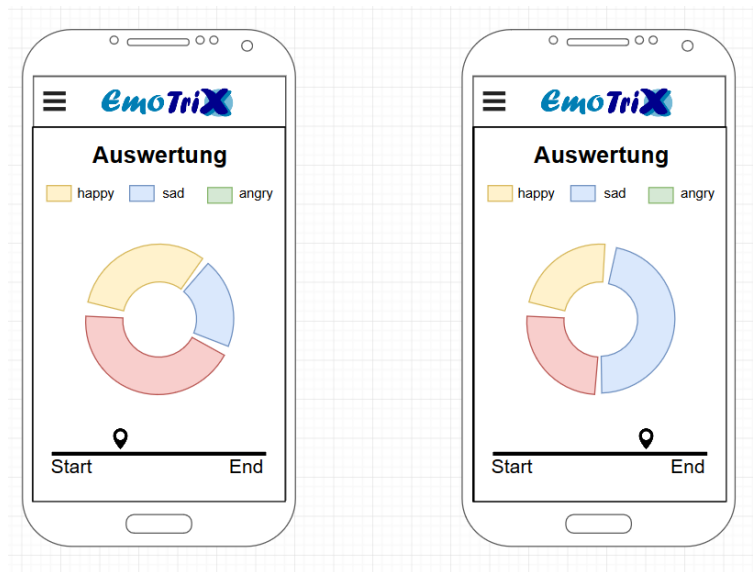


Abbildung 22: Mockup - Auswertung

Die Farben des Kreisdiagramms werden durch eine Legende erklärt, d.h. sie werden den einzelnen Emotionen zugeordnet. Durch berühren einer Farbe soll der Nutzer den konkreten Wert für die Emotion angezeigt bekommen. Unter dem Kreisdiagramm soll außerdem ein Zeitstrahl angezeigt werden. Da es später möglich sein soll, Daten im Hintergrund zu erfassen ist es deutlich sinnvoller, die erfassten Daten nicht alle auf einmal zusammen zu rechnen, sondern den Nutzer einen Bestimmten Zeitpunkt auswählen zu lassen. Mit Hilfe des Zeitstrahls soll der Nutzer diese Möglichkeit geboten bekommen.

## 4 Umsetzung des Prototypen emoTrix

In diesem Abschnitt wird die Umsetzung des Prototyps beschrieben. Dabei werden neben dem Programmcode auch aufgetretene Probleme beschrieben.

### 4.1 Umsetzung der Backendlogik

#### 4.1.1 Sensor-Klasse

TORBEN BRENNER

Wie im Konzept beschrieben besteht das Backend der Anwendung aus den Klassen *Decider*, *CausalityRule* und der Abstrakten Klasse *Sensor*. Im folgenden wird die Umsetzung der einzelnen Klassen beschrieben.

Die Klasse *Sensor* wurde als Abstrakte Klasse umgesetzt. Dies ermöglicht es dem Entwickler, für jeden Test eine Klasse zu implementieren die von der Architektur verstanden wird und in der er die im Test ermittelten Daten auswerten kann. Die Implementation der *Sensor* Klasse sieht folgendermaßen aus:

```
1 export abstract class Sensor{
3     observable: Observable<string>;
4     sensorObserver: Observer<string>;
5
6     constructor(public decider: Decider){
7         this.observable = Observable.create(observer => {
8             this.sensorObserver = observer;
9         });
10    };
11
12    // this forces the specific sensor evaluators to implement this method
13    abstract mapper(data: any): IndicatorScore [];
14
15    onSensorData(data: any) {
16        let indicatorScores = this.mapper(data)
17        this.decider.addIndicatorScores(indicatorScores);
18    }
19 }
```

Listing 1: abstrakte Klasse Sensor

Der Konstruktor der Klasse empfängt eine Instanz der Klasse *Decider*, die wie im Konzept beschrieben ein Singleton ist. Die Funktion *onSensorData* wird von den Subklassen aufgerufen, sobald Daten empfangen wurden und stellt die Logik der Sensor Klasse dar. Zuerst wird bei Empfang der Daten, das von erbenenden Klassen definierte *Mapping* auf die Daten ausgeführt. Danach werden die Daten dem *Decider* hinzugefügt, welcher diese



dann im späteren Verlauf verarbeiten kann (Vgl. 3.2). Durch das Schlüsselwort *abstract* wird bei der Funktion *mapper* gekennzeichnet, dass dieser von den erbbenden Klassen implementiert werden soll. Die Abstrakte Klasse definiert in diesem Fall nur, dass der Mapper aus den Daten des Sensors eine Liste von *IndicatorScores* erzeugen muss. Ein *IndicatorScore* besteht dabei aus einem *indicator* und einem *score*. Der Score ist als numerischer Wert definiert und für die Indicator wurde ein eigener Typ definiert. Dieser Typ ermöglicht es, eine einfache Prüfung der Werte gegenüber dem Typ String und bestimmten Werten durchzuführen. Erlaubte Werte für den Typ Indicator sind zum Beispiel: *“stress”* oder *“angryIndicator”*.

Wie im Konzept beschrieben, verwaltet die Klasse *Decider* alle *IndicatorScores*. Um sicherzustellen, dass die Daten nach Zeitpunkt sortiert gespeichert werden, ermittelt die Methode *addIndicatorScores* erst einen Zeitpunkt und speichert danach die Daten in einer List von Objekten, welche den Zeitpunkt und die dazu gehörige Liste von *IndicatorScores* enthalten. Dadurch können Sensoren die *IndicatorScores* einfach dieser Liste hinzufügen, ohne Gefahr zu laufen, sich gegenseitig zu überschreiben.

Um eine Kommunikation der speziellen Sensoren mit den implementierenden Seiten zu erlauben, wurde der Sensor Klasse ein *Observable* hinzugefügt. Dieses ermöglicht es der Seite sich als *Subscriber* einzutragen. Ein *Subscriber* erhält jedes mal eine Mitteilung, wenn die *next* Funktion des Observers aufgerufen wird. Der Observer wird im Konstruktor aus dem *Observable* erzeugt. Subscriber müssen eine Methode implementieren, in der Sie die Nachrichten des *Observable* verarbeiten. Diese Kommunikation ist Beispielsweise in der Implementation des *FaceSensor* zu sehen.

#### 4.1.2 CausalityRule-Klasse

TORBEN BRENNER

Um die Auswertung der Daten zu ermöglichen, greift die Klasse *Decider* außerdem auf eine Liste von Kausalitätsregeln zu. Diese können mit der Klasse *CausalityRule* definiert werden. Der Konstruktor dieser Klasse nimmt zwei Parameter entgegen. Der Parameter *condition* repräsentiert eine Bedingung, welche eintreten muss sodass die Kausalitätsregel angewandt wird. Die Bedingung wurde innerhalb der Anwendung durch einen Typ umgesetzt. Der Typ prüft dabei auf eine Funktion, die entweder einen oder zwei *indicator* als Parameter entgegennimmt, und als Rückgabewert wahr oder falsch zurückgibt.

```
type Condition = (indicatorA: IndicatorScore , indicatorB?: IndicatorScore
) => boolean;
```

Listing 2: Typ condition

Das bedeutet, dass die Bedingung vom Entwickler frei über die *IndicatorScores* definiert werden kann. Der zweite Parameter der Anwendung repräsentiert die Auswirkungen der

Kausalitätsregel beim Eintreffen der Bedingung. Um diesen Parameter zu definieren wurde ebenfalls ein neuer Typ angelegt. Dieser prüft auf Funktionen, die sowohl eine Liste von *EmotionScores* als auch eine Liste von *IndicatorScores* entgegen nehmen. Außerdem müssen diese Funktionen eine Liste von *EmotionScores* zurückgeben. Das bedeutet, dass in Funktionen dieser Art irgendeine Manipulation der *EmotionScores* stattfinden muss. Die *IndicatorScores* werden benötigt, um eine Veränderung der *EmotionScores* abhängig von ihnen zu ermöglichen. Der zweite Parameter der Kausalitätsregel ist eine Liste solcher Funktionen. Mit diesen beiden Funktionen kann nun in der Klasse *CausalityRule* eine neue Funktion definiert werden, die prüft ob eine Bedingung eingetroffen ist und in diesem Fall, alle Auswirkungen der Regel ausführt.

```

1 public execute(data : Array<EmotionScore>, indicatorScores: IndicatorScore
2     []) {
3     indicatorScores.forEach((indicatorScore) =>{
4         if(this.condition(indicatorScore)){
5             this.effects.forEach((effect) => {
6                 data = effect(data);
7             })
8         }});
9     return data;
10 }
11 }

```

Listing 3: execute Funktion der Klasse CausalityRule

Diese Funktion kann nun von der Klasse *Decider* ausgeführt werden.

#### 4.1.3 Decider-Klasse

LUKAS SEEMANN

Im folgenden Abschnitt wird die Implementierung der wichtigsten Funktionen der *Decider* Klasse aufgezeigt und beschrieben.

```

@Injectable()
2 export class Decider {
3
4     data: Array<{timestamp: number, indicatorScores: IndicatorScore[]}> = [];
5     resultData: Array<{timestamp: number, emotionScores: EmotionScore[]}> = [];
6     causalityRules: Array<CausalityRule> = CRarray;
7
8     ...
9 }

```

Listing 4: Attribute der Decider-Klasse

In Listing 5 sind die Attribute der *Decider*-Klasse, wie sie auch schon in der Architektur eingeführt worden abgebildet. Das Array *data* in Zeile 4 enthält IndicatorScores mit dem Zeitpunkt in Millisekunden, an dem diese Scores hinzugefügt wurden. Das Hinzufügen ist über die Mapper der einzelnen Sensoren geschehen, die die *addIndicatorScore*-Funktion des Deciders aufrufen.

In Zeile 5 wird das *resultData*-Array initialisiert. Anhand der vorhandenen IndicatorScores im *data*-Array werden mithilfe der Kausalitätsregeln für alle 10 Sekunden in EmotionScores umgewandelt. Wie dies umgesetzt ist, ist in der *decide*-Funktion zu sehen.

In Zeile 6 wird das Array der *CausalityRules* über eine Konstante *CRArray* initialisiert. Dieses konstante Array wird in Kapitel 4.3.2 genauer beschrieben.

```

1 public decide() {
    let start: number = this.findRange().start;
    let end: number = this.findRange().end;
    for (var i: number = start; i <= end; i = i + 10000) {
        var j: number = i + 9999;
        this.assumeEmotion(i, j);
    }
}

```

Listing 5: *decide*-Funktion der Decider-Klasse

Die *decide*-Funktion ist in Listing 5 abgebildet. Mithilfe der internen *findRange*-Funktionen werden in den Zeilen 2 und 3 die minimale beziehungsweise maximale *timestamp* des *data*-Array bestimmt. So hat man den ersten und letzten Zeitpunkt (in Millisekunden) gefunden, zu dem IndicatorScores durch die einzelnen Tests hinzugefügt wurden.

Mithilfe einer *for*-Schleife (Zeile 4) wird anschließend in 10 Sekunden-Schritten über den durch *start* und *end* festgelegten Bereich iteriert. Dabei wird in der *for*-Schleife in Zeile 5 eine Hilfsvariable *j* eingeführt, die mit dem Iterator *i* plus 9999 Millisekunden initialisiert wird. Mithilfe von *i* und *j* wird also in jedem Iterationsschritt ein Intervall von 10 Sekunden festgelegt. Auf dieses Intervall wird in Zeile 6 die *assumeEmotion*-Funktion angewendet.

```

private assumeEmotion(start: number, end: number) {
    let currentIndicators = this.data.filter(e => e.timestamp >= start && e
        .timestamp <= end);
    var emotionScores: Array<EmotionScore> = this.executeCausalityRules(
        currentIndicators);
    this.resultData.push({ timestamp: start, emotionScores: emotionScores });
}

```

Listing 6: *assumeEmotion*-Funktion der Decider-Klasse

Listing 6 zeigt den Aufbau der `assumeEmotion`-Funktion, die einen *start* und ein *end* als Parameter erhält. Anhand dieser Zahlen wird anfangs in Zeile 2 das *data*-Array gefiltert. Hierbei werden alle Einträge in eine Variable *currentIndicators* geschrieben, die mit ihrer *timestamp* zwischen *start* und *end* liegen.

In Zeile 3 wird dann eine neue Variable *emotionScores* initialisiert, die als Wert das Ergebnis der `executeCausalityRules`-Funktion angewandt auf die *currentIndicators* erhält. Die Implementierung dieser Funktion, die *EmotionScores* zurückliefert, ist in Listing 7 zu sehen. In Zeile 4 von Listing 6 werden dann diese *EmotionScores* mit der *start*-Timestamp zum *resultData*-Array hinzugefügt.

```

1 private executeCausalityRules(data: Array<{timestamp: number,
   indicatorScores: IndicatorScore[]}>){
2     var emotionScores: EmotionScore[] = [];
3     emotions.forEach(emotion => {
4         var emotionScore : EmotionScore = {emotion, score: 0};
5         emotionScores.push(emotionScore);
6     });
7     data.forEach((e) => {
8         this.causalityRules.forEach((causalityRule) => {
9             emotionScores = causalityRule.execute(emotionScores, e.
            indicatorScores) ;
10        })})
11 )
12 return emotionScores;
13 }

```

Listing 7: `executeCausalityRules`-Funktion der *Decider*-Klasse

Die `executeCausalityRules`-Funktion erhält als Input ein Array von *timestamp-IndicatorScores*-Kombinationen. Zunächst wird in Zeile 2 ein leeres *EmotionScore*-Array initialisiert. In den Zeilen 3 bis 6 wird dieses dann befüllt, indem für jede vorhandene Emotion (*angry*, *happy*, *sad*, *suprised*) ein Score von 0 angelegt wird. So erhält man *EmotionScores*, wobei jede Emotion mit 0 bewertet ist.

Im Anschluss daran wird in Zeile 7 über jeden vorhandenen Eintrag des Input iteriert. Für jeden Eintrag, der aus *timestamp* und *indicatorScores* besteht, wird dann jede im konstanten *CausalityRules*-Array enthaltene Kausalitätsregel ausgeführt (Zeile 8 bis 10). Durch jede ausgeführte Kausalitätsregel werden die ursprünglich initialisierten *EmotionScores* modifiziert. Dies wird für alle Einträge im Input der Methode wiederholt. Am Ende werden die final modifizierten *EmotionScores* zurückgegeben (Zeile 12).

## 4.2 Datenerfassung in Form von Emotionstests

### 4.2.1 GSR-Test

LUKAS SEEMANN

Die Erfassungsmöglichkeit mit der höchsten Priorität war die Messung der Hautleitfähigkeit mithilfe von EDA- beziehungsweise GSR-Sensoren. Da dies nicht mit im Smartphone enthaltenen Sensoren möglich war, wurde zusätzlich ein Arduino-Mikrocontroller benötigt, um die Messung durchzuführen. Zunächst wird die Entwicklung auf dem Arduino-Board mit allen zusätzlichen Modulen beschrieben. Im Anschluss daran wird thematisiert, wie das Arduino-Board mit der App verbunden wurde.

Für das Projekt wurde ein Arduino UNO R3 Board verwendet.<sup>120</sup> Dieser kann mit Stromzufuhr über ein Netzteil oder per USB betrieben werden. Als Sensor wurde ein GSR Sensor des Grove-Toolkits verwendet<sup>121</sup>, der bereits in Abbildung 5 gezeigt wird. Am Sensor selbst werden die Elektroden für die Finger angebracht. Da ein handelsübliches Arduino UNO R3 Board nicht über den benötigten Anschluss für den Grove GSR-Sensor verfügt, muss zusätzlich noch ein Grove Base Shield angebracht werden. Dieses kann auf das Arduino Board aufgesteckt werden und erweitert es um viele verschiedene Anschlüsse, unter anderem für Sensoren.

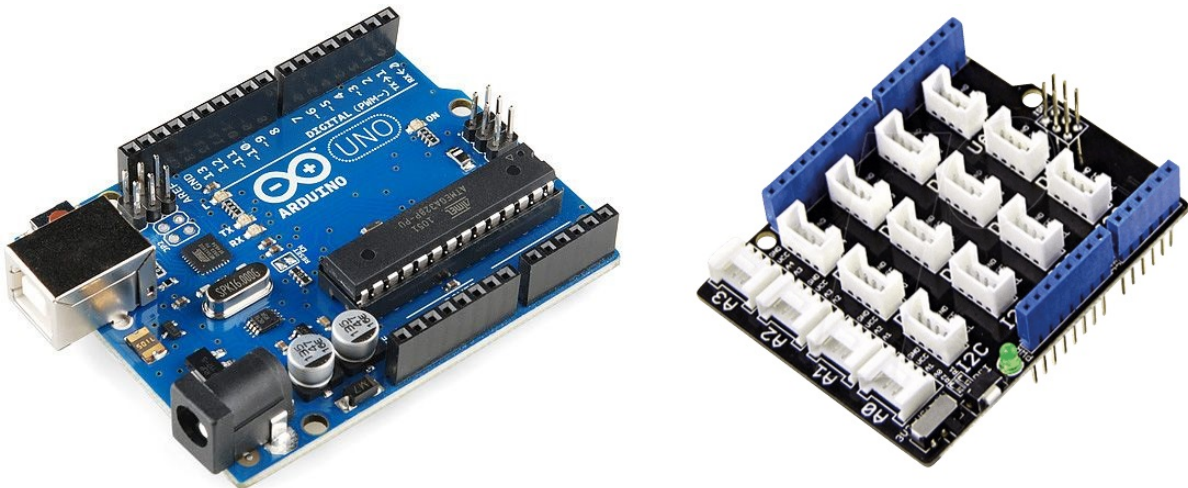


Abbildung 23: Arduino UNO R3 (links) und Grove Base Shield<sup>122</sup>

Mit diesen Komponenten werden die vom Sensor zurückgelieferten Daten an den Arduino geleitet. Von dort aus müssen die Daten, an die mobile Applikation weitergeleitet werden. Aus diesem Grund muss an das Arduino Board ein Bluetooth-Modul angebracht werden, das Daten senden und empfangen kann. Das Empfangen von Daten ist notwendig, um die Messung zu starten, wohingegen das Senden für die Übermittlung der Sensordaten benötigt wird. Heutige Smartphones verfügen meistens immer über eine Bluetooth-Schnittstelle, aus welchem Grund Bluetooth gut für die Übertragung geeignet ist. Eine weitere Möglichkeit wäre die Übertragung über WiFi gewesen. Das Arduino-Board wurde

<sup>120</sup>Vgl. Ard18a.

<sup>121</sup>Vgl. Gro18.

<sup>122</sup>Bilder von: Sou18; Rei18.

mit einem HC05-Bluetooth-Modul erweitert, welches Daten senden und empfangen kann. Dieses ist in Abbildung ? zu sehen.

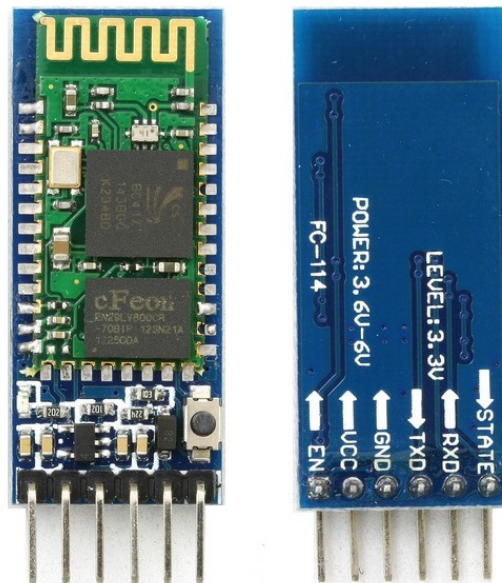


Abbildung 24: HC-05-Bluetooth-Modul für Arduino<sup>123</sup>

Die Beschreibung der Entwicklungsarbeiten wird in zwei Teile aufgespalten. Der erste Teil ist der Quellcode des Arduinos, der zweite Teil die Entwicklung des Emotionstest in der emoTrix-App.

In Listing ? ist der Quellcode des Arduinos abgebildet. In Zeile 1 wird die SoftwareSerial-Bibliothek eingebunden, die eine Verwendung der Pins des Arduinos für verschiedene Module ermöglicht. In Zeile 2 wird dem Arduino mitgeteilt, dass auf den Pins 10 und 11 ein Bluetooth-Modul angeschlossen ist und eine Konstante (GSR) festgelegt, die auf den Anschluss A0 des Grove Shields verweist, an dem der GSR Sensor angeschlossen ist.

Generell besteht der Programmcode des Arduinos immer aus zwei Bestandteilen: einem Setup-Block und einem Loop-Block. Der Setup-Block wird einmalig beim Einschalten des Arduinos ausgeführt. Danach wird der Loop-Block solange wiederholt, bis der Arduino ausgeschaltet wird.

In Zeile 7 innerhalb des Setup-Blocks wird die Geschwindigkeit der seriellen Datenübertragung der Ports des Arduinos, die mit dem Bluetooth-Modul verbunden sind. Hierbei wird die Geschwindigkeit auf 9600 Bits pro Sekunde gesetzt.<sup>124</sup> Dies entspricht der üblich verwendeten Geschwindigkeit und hat in Tests sehr gut funktioniert.

```
1 #include <SoftwareSerial.h>
  SoftwareSerial BTserial (10, 11); const int GSR=A0;
3 int sensorValue=0; int gsr_average=0;
  boolean measuring = false; char BTString;
```

<sup>124</sup>Vgl. Ard18b.



```

5
void setup() {
7   BTserial.begin(9600);
}
9
void loop() {
11  BTString = BTserial.read();
    if(BTString == 'S'){
13      measuring = true;
    }
15  if(BTString == 'F'){
      measuring = false;
17  }
    if(measuring){
19      long sum=0;
      for(int i=0;i<10;i++){
21          sensorValue=analogRead(GSR);
          sum += sensorValue; delay(5);
23      }
      gsr_average = sum/10;
25      BTserial.print(gsr_average); BTserial.println( " ");
    }
27 }

```

Listing 8: Quellcode des Arduinos

Der Loop-Block beginnt in Zeile 11 mit dem Auslesen der Daten, die über das Bluetooth-Modul empfangen werden. Die Variable *BTString* wird mit diesen Daten beschrieben. Die App muss zum Starten der App den String *S* (für Start) per Bluetooth übertragen. Ist dies der Fall, wird die Variable *measuring* auf true gesetzt. Mit dem String *F* (für Finished) kann die App dem Arduino das Stoppsignal für die Messung geben. Demenstprechend wird *measuring* auf false gesetzt. Dies ist in den Zeilen 15 bis 17 umgesetzt. In Zeile 18 wird über die *measuring*-Variable überprüft, ob gemessen werden soll. Wenn ja, wird eine Variable für die Summe von 10 Messdaten initialisiert. Anschließend werden in Abstand von 5 Millisekunden 10 Messungen durchgeführt. In Zeile 21 wird die eigentliche Messung des Sensors durchgeführt. Der hier verwendete GSR-Sensor liefert als Output einen Integer-Wert, der die Stromspannung auf dem seriellen Port des Arduinos in Volt entspricht. Dieser Wert hängt von der gemessenen Hautleitfähigkeit ab. Die Umrechnung in den Hautleitfähigkeit geschieht dann in der App und wird im Anschluss noch betrieben. Im Arduino-Code selbst werden die Daten überliefert, die auch der Sensor übermittelt. Alle 10 Messungen werden nach und nach aufaddiert. Dies geschieht in der for-Schleife in den Zeilen 20 bis 23. Anschließend wird die Summe durch 10 geteilt, sodass man den Durchschnitt aller 10 Werte erhält. Dieses Verfahren wird durchgeführt, da die Messdaten des Sensors Schwankungen aufweisen, die dadurch eliminiert werden

können. In Zeile 25 wird schließlich der Messwert auf den Port des Bluetooth-Moduls geschickt und damit versendet. Als Trennzeichen zum nächsten Wert wird ein Semikolon angehängt.

Wenn die Messung gestartet wurde, erhält die mobile Applikation also alle 50 Millisekunden vom Arduino per Bluetooth einen Integer-Wert übermittelt.

Die Daten, die der Arduino schickt, müssen innerhalb der App verarbeitet werden. Hierfür wurde eine neue TestPage mit dem Namen GSRPage erstellt. Diese Page dient dazu, sich mit dem Arduino Bluetooth-Modul zu verbinden und die Messung zu starten und zu stoppen. Zusätzlich werden die gemessenen Werte in einem Graphen dargestellt, um zu veranschaulichen, wie sich die Hautleitfähigkeit während der Messung ändert.

In der `gsr.ts`-Datei sind alle Funktionen hinterlegt, die alle Methoden für die Umsetzung der genannten Funktionalitäten implementiert. In Listing 2 ist ein Ausschnitt aus der Typescript-Datei mit den wichtigsten Funktionen abgebildet.

```
1 startMeasuring() {  
  this.bluetoothSerial.write('S').then((data: any) => { })  
3  .catch((e) => {  
    console.log(e);  
5  });;  
  }  
7  
  stopMeasuring() {  
9  this.bluetoothSerial.write('F').then((data: any) => {})  
    .catch((e) => {  
11 console.log(e);  
    });  
13 }  
}
```

Listing 9: JS Code

Die Funktion *startMeasuring* implementiert das Starten der Messung. Für die GSRPage wurde das Ionic-Package Bluetooth Serial hinzugefügt, das Bluetooth-Optionen des Smartphones für die App verfügbar macht. Das Package wird hier mit *this.bluetoothSerial* referenziert. Das Pairen und Verbinden des Smartphones mit dem Bluetooth-Modul ist selbsterklärend (über eine *connect*-Methode) und deswegen im Listing nicht aufgeführt. Nachdem die Verbindung eingerichtet wurde, kann die *startMeasuring*-Funktion aufgerufen werden. Dabei wird in Zeile 2 mit *write* ein String an das verbundene Bluetooth-Modul gesendet. Wie bereits beschrieben muss zum Start der String *S* gesendet werden. Falls ein Fehler auftritt, wird dieser in Zeile 4 geloggt. Wenn alles ordnungsgemäß funktioniert, empfängt der Arduino den String und startet anschließend die Messung.

Die *stopMeasuring*-Funktion funktioniert analog zur *startMeasuring*-Funktion mit dem



Unterschied, dass hier der String  $F$  zum Stoppen der Messung gesendet wird.

```
1 this.bluetoothSerial.subscribe(";").subscribe(function (data){
    self.value = data.substring(0,data.length - 1);
3    self.value = ((1024+2*self.value)*10000)/(512-self.value);
    if(self.time%5 == 0){
5        if(self.time != 0){
            var data: any = {value: self.value, oldValue: self.oldValue};
7            self.GsrSensor.onSensorData(data);
        }
9        self.oldValue = self.value;
    }
11    if(self.time%20 == 0){
        self.addData(self.lineChart, self.time, self.value);
13    }
    self.time++;
15 }, function (error){
    console.log(error);
17 });
```

Listing 10: JS Code

In Listing 6 ist implementiert, dass das Bluetooth des Smartphones auf einen neu eintreffenden Integer-Wert des Arduinos reagieren kann. Das BluetoothSerial dient dabei als Observable, den die App abonnieren kann. Dies bedeutet, dass die App darauf hingewiesen wird, wenn neue Daten angekommen sind. Mit `subscribe(";")` wird die App jedes Mal informiert, wenn ein Semikolon übertragen wurde. Das Semikolon wurde im Arduino-Code als Trennzeichen eingesetzt und signalisiert, dass ein Integer-Wert abgeschlossen ist. Mit dem nächsten `subscribe` wird bestimmt, was ausgeführt wird, wenn ein Semikolon empfangen wird. Als `data` (Zeile 1, Ende) wird immer alles übertragen, was seit dem letzten Ausführen der Funktion übertragen wurde. Es handelt sich also immer um einen Integer-Wert und das Semikolon. In Zeile 2 wird aus diesem Grund das letzte Zeichen der Übertragung abgeschnitten, sodass der `value`-Variable nur der Integer-Wert zugewiesen wird. In Zeile 3 folgt die Umwandlung des Integer-Wert, der der Stromspannung auf dem seriellen Port entspricht, in den Hautwiderstand.

Nach Herstellerangaben kann mit der Formel  $(1024+2*x)*10000)/(512-x)$  der Hautwiderstand berechnet werden, wobei  $x$  der gemessenen Stromspannung entspricht.<sup>125</sup> Warum diese Formel genau so aussieht, hat mit dem internen Aufbau des Sensors zu tun, der aus Analog-Digital-Umsetzern und Verstärker aufgebaut ist.<sup>126</sup> Da die genaue Erklärung der Formel sehr komplex ist und den Rahmen der Arbeit sprengen würde, wird die Formel hier nicht weiter thematisiert. Wichtig ist, dass man durch diese Umrechnung man den

---

<sup>125</sup>Vgl. Gro18.

<sup>126</sup>Vgl. Com18, 1. Forumsantwort.

Hautwiderstand des Benutzers in Ohm erhält.

Da alle 50 Millisekunden ein Wert übertragen wird, wird kein neuer Timer benötigt. Dieser wurde anfangs mit dem Wert 0 initialisiert und in Zeile 14 nach jedem neuen Wert um 1 erhöht. Da ein Wert und damit eine Verarbeitung alle 50ms für die App einen sehr hohen Workload bedeuten würde, wird nur jeder fünfte übertragene Wert betrachtet. Dies ist in Zeile 4 mit einer if-Bedingung umgesetzt. In Zeile 6 werden immer zwei aufeinander folgende Ohm-Werte zusammengefasst und anschließend in Zeile 7 die `onSensorData`-Methode mit diesen Daten aufgerufen. Zwei Werte sind für den Mapper trivialerweise genug, um zu schauen ob der Wert gestiegen oder gesunken ist. Da am Anfang nur der erste Wert vorhanden ist, wird dieser Fall in Zeile 5 abgefangen. Das Setzen des alten Werts (*oldValue*) geschieht in Zeile 9.

In Zeile 11 bis 13 wird implementiert, dass außerdem jeder zwanzigste Wert in einem Graphen auf der GUI eingetragen wird. Eine häufigere Darstellung ist ebenfalls nicht sehr vorteilhaft für die Benutzung der App, da dies sehr rechenlastig ist. Der Trend des Graphen ändert sich außerdem nicht wesentlich, wenn nicht alle Datenpunkte eingetragen werden.

In Zeile 15 bis 17 wird letztlich noch das Error-Handling implementiert.

#### 4.2.2 Face-Test

TORBEN BRENNER

Eine der am höchsten priorisierten Erfassungsmöglichkeiten für Emotionen, war die Messung der Emotionen anhand der Mimik des Menschen mithilfe der Smartphonekamera. Wie im Mockup<sup>127</sup> beschrieben soll der Nutzer zuerst ein Bild machen können und sich dieses anzeigen lassen können, bevor es ausgewertet werden soll. Wenn das Bild zur Analyse freigegeben wird, soll dem Nutzer ein Ladebildschirm angezeigt werden. Die Analyse selbst wird nicht auf dem Smartphone durchgeführt, sondern mit Hilfe der Gesichtserkennungsschnittstelle von Microsoft<sup>128</sup>.

Zu Beginn der Entwicklung wurde eine Komponente entwickelt, welche das Schießen eines Fotos ermöglicht. Ionic-Komponenten bestehen aus einem HTML-Template, einer SCSS Datei und einem TypeScript Script. Diese Aufteilung in drei Dateien ermöglicht eine Trennung von grafischer Darstellung und der Logik. Die grafische Darstellung der Komponente besteht aus einem Bild, das nur angezeigt wird wenn mit der Komponente auch ein Bild gemacht wurde, und einem *Ionic-Button*, der das Fotografieren auslöst. Interessanter ist die Logik der Komponente, die daraus besteht abhängig von der Auswahl des Entwicklers entweder ein Bild zu machen oder eine Videoaufnahme zu starten. Dies wird mit der Übergabe eines Input Parameters ermöglicht.

1 `@Input() pictureMode: boolean;`

<sup>127</sup>siehe 3.6.3 S.40

<sup>128</sup>siehe Mic18b.

```
@Output() sourceChanged: EventEmitter<String> = new EventEmitter<String>();
```

Listing 11: Input und Output der Kamera Komponente

Da die Komponente aber auch die Bilder, die mit ihr gemacht wurden, herausgeben soll wird ein *Output* definiert. Dieser wird mit Hilfe eines *EventEmitters* realisiert, der ein Event auslöst auf das andere Teile der Anwendung, die eine Kamera Komponente implementieren reagieren können. Das Event gibt hierbei einen String aus, der in diesem Test dem Base64 codierten Bild entspricht. Im folgenden zusehen ist die Einbindung der Kamera Komponente in den Face-Test.

```
<camera [pictureMode]="true" (sourceChanged)="onSourceChanged($event)">/>
</camera>
```

Listing 12: Einbindung Kamera Komponente

Das Fotografieren wird in Ionic mit Hilfe einer *native* Komponente umgesetzt. Diese Komponente erlaubt den plattformunabhängigen Zugriff auf die Kamera<sup>129</sup>. Dazu wird in der App direkt eine Weiterleitung an die ausgewählte Kamera App des Nutzers durchgeführt. Nachdem der Nutzer mit dieser ein Bild gemacht hat, kann entweder der Speicherort oder das Bild selbst zurückgegeben werden. Der zurückgegebene Wert wird dann von der Kamera Komponente per *EventEmitter* ausgegeben.

Wie in der Einbindung der Komponente zu sehen ist, wird beim auslösen des Events *sourceChanged* im Face-Test eine Funktion aufgerufen. Dies geschieht mittels *Event Binding*. Die aufgerufene Funktion speichert den empfangenen Source Wert in einer lokalen Variable im Face-Test. Bei Betätigung des *Send Picture Buttons* wird daraufhin die Analyse gestartet. Die Analyse wird von der *FaceSensor* Klasse durchgeführt. Dazu wird vom Face-Test die Funktion *prepareData* aufgerufen, die eine HTTP POST Anfrage an die Microsoft Gesichtserkennungsschnittstelle durchführt. Die Details der Anfrage sind in folgender Codeausschnitt zu sehen:

```
1 {
  url: 'https://westeurope.api.cognitive.microsoft.com/face/v1.0/detect?
    returnFaceAttributes=emotion',
3  type: 'POST',
  processData: false,
5  headers: {
    'Ocp-Apim-Subscription-Key': '{Api Key}'
7  },
  contentType: 'application/octet-stream',
9  data: this.makeblob(base64Data)
}
```

<sup>129</sup>Ion18a.

### Listing 13: Aufbau Anfrage an Gesichtserkennungsschnittstelle

Das einzig nennenswerte an dieser Anfrage ist die Übertragung des Base64 codierten Bildes. Diese geschieht nämlich in Form eines Blob(Binary Large Object). Laut *MDN web docs* ist ein Blob eine “dateiähnliche Menge unveränderlicher Roh-Daten”<sup>130</sup>. Diese wird nicht nur bei der Übertragung großer Datenmengen verwendet, sondern auch beim Abspeichern größerer Dateien in Datenbanken. Um das Base64 codierte Bild in einen Blob zu übertragen wurde die Hilfsmethode *makeblob* genutzt. Diese wurde nicht von mir selbst entwickelt sondern von Kalpesh Singh<sup>131</sup>. Die Funktion trennt die *Content-Type* Informationen von den Roh-Daten und überträgt diese dann in einen neuen Blob. Bei der Base64 Codierung müssen die Roh-Daten erst mit Hilfe der JavaScript Funktion *charCodeAt* in Ganzzahl Werte übertragen werden. Der generierte Blob wird dann im *Body* der HTTP Anfrage mit übertragen.

Nach dem senden der Request wird auf der Oberfläche der App dem Nutzer mit Hilfe eines *Ionic Loaders*<sup>132</sup> signalisiert, dass im Hintergrund gerade eine Verarbeitung der Daten stattfindet. Da die Sensor Klasse ein sogenanntes *Observable* implementiert, können Statusmeldungen zwischen der Face-Test Seite und dem *FaceSensor* ausgetauscht werden. Die Statusmeldungen werden auf Seiten des *FaceSensor* initiiert und von der Seite mit Hilfe einer *Callback* Funktion verarbeitet.

```
1  this.faceSensor.observable.subscribe(data => {
2    if(data === "Sending data to Microsoft"){
3      this.loader.present();
4    }
5    if(data === "empty result"){
6      this.loader.dismiss();
7      alert("Microsoft Server couldn't recognize your face. Please make
8      sure your face is visible");
9    }
10   if(data.startsWith("result:")){
11     console.log("This is a result");
12     this.loader.dismiss();
13     let emotions = JSON.parse(data.replace("result:", ""));
14     console.log(emotions);
15     this.faceSensor.onSensorData(emotions);
16   }
17   if(data.startsWith("error:")){
18     this.loader.dismiss();
19     let errorString = data.replace("error:", "");
20     alert(errorString);
```

<sup>130</sup>Vgl. Mdn18, Abschnitt Übersicht.

<sup>131</sup>Vgl. Sin16, Antwort von Kalpesh Singh.

<sup>132</sup>Ion18e.

```

20     }
    });

```

Listing 14: Callback Methode auf FaceSensor Observable

Die Daten des *FaceSensor* werden mit Strings übertragen. Durch das Schlüsselwort “result” wird gekennzeichnet, dass es sich beim Rest des Strings um Ergebnis Daten handelt, die mit der *JSON.parse* Methode in ein JavaScript Objekt übertragen werden können. Durch “error” wird gekennzeichnet, dass ein Fehler aufgetreten ist der ausgegeben werden soll. Dies wird mit Hilfe eines *alert* umgesetzt. Es kann außerdem passieren, dass die Microsoft Gesichtserkennungsschnittstelle kein Gesicht erkennt. Das ist in unseren Testläufen insbesondere dann aufgetreten, wenn das Bild um 90 Grad gedreht war. Für diesen Fall haben wir eine eigene Fehlernachricht eingebaut.

Wenn aber alles ohne Fehler abgelaufen ist, erhalten wir vom Microsoft Server ein JSON Objekt der Form:

```

1  [
2  {
3    "faceId": "579838a9-fb50-4333-8770-32858b2d0fc6",
4    "faceRectangle": {
5      "top": 131,
6      "left": 177,
7      "width": 162,
8      "height": 162
9    },
10   "faceAttributes": {
11     "emotion": {
12       "anger": 0,
13       "contempt": 0,
14       "disgust": 0,
15       "fear": 0,
16       "happiness": 0.001,
17       "neutral": 0.987,
18       "sadness": 0.001,
19       "surprise": 0.01
20     }
21   }
22 }
23 ]

```

Listing 15: Ergebnis Gesichtserkennungsschnittstelle, Beispiel Microsoft

Aus diesem Objekt interessiert uns der Datensatz *emotion*. Diesen kann man mittels JavaScript einfach auslesen. Eine Problematik die derzeit nicht im Face-Test behandelt wird, ist die Verarbeitung von mehreren Gesichtern in einem Bild. In diesem Fall würde ein weiteres Objekt mit einer anderen *faceId* zurückgegeben werden. Für die Zukunft

wäre es denkbar die *faceId* in der Ausgabe zu kontrollieren und den Nutzer zur Identifikation ein erstes Bild machen zu lassen in dem die *faceId* ermittelt wird. Dies wäre zum Beispiel in den Settings möglich. Problematisch an diesem Ansatz ist aber, dass die *faceId* nur temporär ist und nach 24 Stunden abläuft. Also müsste der Nutzer nach einem Tag Inaktivität diese neu setzen.

Zum Abschluss des Face-Tests fehlt nur noch die Implementierung der *mapper* Funktion im *FaceSensor*. Diese ist relativ einfach, da der Funktion einfach das *emotion* Objekt aus der HTTP Antwort weitergegeben werden kann. Dort werden die Werte dann in *IndicatorScores* übertragen, was in diesem Fall trivial ist, da Sie bereits Prozente darstellen und somit einfach übernommen werden können. Das senden der Daten an den Decider wird durch die Funktion *onSensorData* der Klasse *Sensor*, von der der *FaceSensor* erbt, bereits übernommen, womit die Implementation abgeschlossen ist.

## 4.3 Beschreibung der implementierten Kausalitätsregeln

TORBEN BRENNER

Alle von uns implementierten Kausalitätsregeln befinden sich aus Übersichtsgründen in der Datei *CausalityRulesArray.ts*. Hier werden zum aktuellen Zeitpunkt sechs verschiedene Kausalitätsregeln implementiert (zwei davon für den GSR-Test und vier für den Face-Test). Diese Regeln werden zusammen in einem Array exportiert, welches von der *Decider* Klasse eingebunden wird. Im folgenden sollen die verschiedenen Regeln genauer erläutert werden.

### 4.3.1 Regeln des GSR-Test

TORBEN BRENNER & LUKAS SEEMANN

Da der *Activation* Indikator entweder 0, 0.5 oder 1 sein kann, ist es nur notwendig zwei Regeln für diesen zu implementieren. Wenn der *Activation* Indikator einen Wert von 0.5 hat, ist dieser als neutral anzusehen, d.h. die vom GSR-Sensor erfassten Werte zeigen keine Tendenz für den Stress. In diesem Fall finden keine Veränderungen an den *EmotionScores* statt, weshalb hierfür auch keine Kausalitätsregel zu implementieren ist. Anders sieht das bei den Werten 0 und 1 aus. In diesen Fällen zeigt sich eine Tendenz der erfassten Werte für gestresst(1) oder nicht gestresst(0). Deshalb unterscheidet sich die Bedingung der beiden Regeln gerade mal durch die Prüfung auf den bestimmten *score*.

```
1 (indicatorScore: IndicatorScore) => {  
    return indicatorScore.indicator == <indicator>"activation" &&  
    indicatorScore.score == 1;  
3 }
```

Listing 16: Bedingung der ersten Kausalitätsregel(gestresst)

Auch die Auswirkungen der beiden Regeln sind sich ähnlich. So werden bei Eintritt der ersten Regel zu den *EmotionScores* für die Emotionen *wütend*, *fröhlich* und *überrascht* jeweils eins zu dem Score hinzu addiert. Bei eintritt der zweiten Regel wird stattdessen eins subtrahiert. Diese Änderungen wirken zwar sehr klein im Vergleich zu den Änderungen der Kausalitätsregeln des Face-Test, werden sich aber aufgrund der Menge von Daten in einem Zeitraum dennoch bemerkbar machen. So werden in einer Sekunde circa 20 Werte erfasst, während dem der Face-Test deutlich länger für die Durchführung benötigt.

#### 4.3.2 Regeln des Face-Test

TORBEN BRENNER

Für den Face-Test wurden vier Kausalitätsregeln definiert. Von diesen vier Regeln ist jeweils eine für einen bestimmten *EmotionScore*. Da bei der Eignung des Gesichts als Emotionsindiz festgestellt wurde, dass dieses äußert aussagekräftig ist werden die durch den Face-Test erfassten Werte sehr stark gewichtet. Die Werte die vom Test ermittelt werden sind Prozentzahlen, die aussagen wie wahrscheinlich eine bestimmte Emotion ist. Deshalb wird bei der Erfassung eines solchen Wertes dieser mit 100 multipliziert und dann auf den jeweiligen *EmotionScore* addiert.

```

1  (indicatorScore: IndicatorScore) => {
      return indicatorScore.indicator == <indicator>"angryIndicator" &&
      indicatorScore.score != undefined;
3  },[
      (data: EmotionScore[], indicatorScores :IndicatorScore[]) => {
5          let angry = data.find((emotionScore) => {return emotionScore.emotion
              == <emotion>"angry"});
              let angryIndicator = indicatorScores.find((indicatorScore) => {return
              indicatorScore.indicator == <indicator>"angryIndicator"});
7          angry.score = angry.score + (100 * angryIndicator.score);
              return data;
9      }
      ]

```

Listing 17: Kausalitätsregel basierend auf Face-Test für Emotion Wut

Als Bedingung das eine Kausalitätsregel des Face-Test eintritt, ist gewählt das der jeweilige Indikator definiert ist. Diese sind im Normalfall nicht definiert, werden aber bei Erfassung eines Bildes gesetzt. Im Falle der Wut ist der Indikator der sogenannte *angry-Indicator*. Bei Eintritt der Bedingung werden die Auswirkungen wie oben beschrieben ausgeführt. Die andern drei Regeln für den Face-Test unterscheiden sich nur durch die Auswahl des *Indicator*- und *EmotionScores*.

## 4.4 Benutzeroberfläche der App

LUKAS SEEMANN

In diesem Kapitel wird die Benutzeroberfläche der mobilen Applikation beschrieben. Das Kapitel soll auch als Benutzungsanleitung für die einzelnen Funktionen der App dienen.

### 4.4.1 HomePage

LUKAS SEEMANN

In Abbildung ? ist der StartScreen der App zu sehen. Dieser wird aufgerufen, wenn der User die App öffnet. Hier werden dem User alle Emotionstests aufgelistet, die er durchführen kann. Durch ein Klicken auf das Label oder den Pfeil gelangt der User auf den Screen des jeweiligen Tests. Von den aufgelisteten Tests wurde der Keyboard Scanner nicht umgesetzt. Ein Klicken auf dieses Label führt zu einer Meldung, dass die Seite noch nicht implementiert wurde.

Führt der User einen Emotionstest erfolgreich durch, verwandelt sich bei der Rückkehr auf die HomePage der Pfeil am Ende des Test in einen Haken. So erhält der User einen Überblick darüber, welche Test er bereits durchgeführt und welche noch offen sind.

Durch ein Klicken auf den *Auswertung*-Button wird eine Auswertung der Testergebnisse durchgeführt, die zum Zeitpunkt des Klickens im Decider hinterlegt sind. Der User wird anschließend zur DecisionPage navigiert, welche ihm seine Ergebnisse visualisiert.

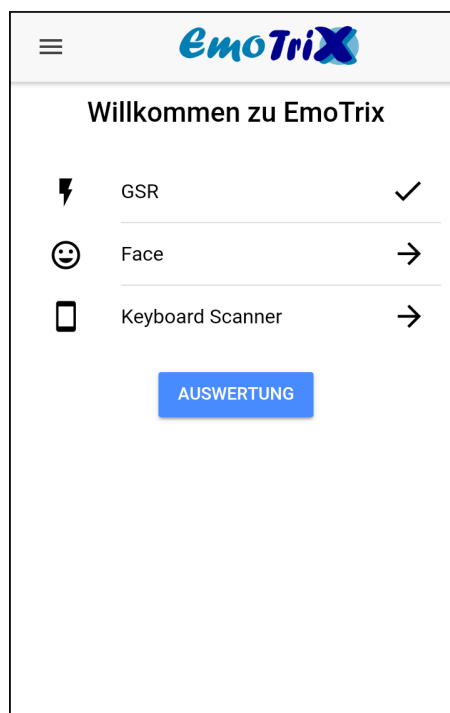


Abbildung 25: GUI der HomePage

Über das Klicken der drei Striche in der Toolbar oder ein Swipen nach rechts, wird seitlich ein Menü angezeigt, über welches der User zu weiteren Screens der App gelangen



kann. Wie bereits im Mockup beschrieben, gibt es hier die Möglichkeiten auf eine Anleitungssseite, eine Übersicht der durchgeführten Messungen oder eine Einstellungsseite zu gelangen. Aus Zeitgründen konnte hiervon nur die Anleitungssseite umgesetzt werden. Über dieses seitliche Menü kann der User außerdem auch auf jeder beliebigen Seite über die erste Auswahlmöglichkeit (*Home*) zurück zum Startscreen gelangen.

#### 4.4.2 GSRPage

LUKAS SEEMANN

Die GSRPage ist im Wesentlichen genau so umgesetzt worden, wie das Mockup den Screen vorgesehen hat. In Abbildung ? sind zwei Beispiele gezeigt, was der User auf der GSRPage sehen kann.

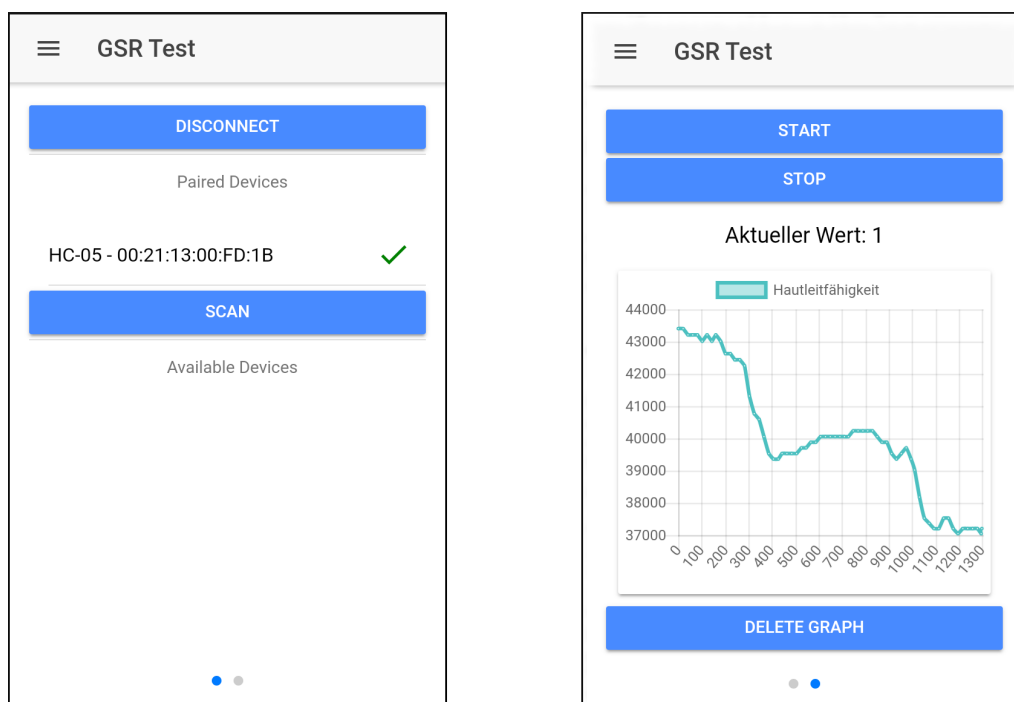


Abbildung 26: GUI der GSRPage

Die GSRPage besteht aus 2 Slides, zwischen denen per Swipen hin und her navigiert werden kann. Der erste Slide befasst sich mit dem Verbinden mit dem Bluetooth-Modul des Arduinos. Wie bereits im Mockup beschrieben, kann der User mit dem Button *Scan* nach nicht verbundenen HC-05-Modulen suchen. Alle Ergebnisse werden dann unter *Available Devices* angezeigt.

Unter *Paired Devices* werden alle Bluetooth-Module gezeigt mit denen der User sich bereits schon vorher verbunden hat. Durch Klicken auf einen Eintrag in *Paired Devices* oder nach dem *Scan* in *Available Devices* erscheint eine Anzeige, ob der User sich gerne mit dem Modul verbinden möchte. Bestätigt er dies und die Verbindung ist erfolgreich, wird dies durch einen grünen Haken neben dem Eintrag angezeigt. Damit ist das Verbinden

mit dem Bluetooth-Modul abgeschlossen und der Test kann gestartet werden. Durch ein Swipen nach rechts gelangt man auf den Screen, um den Test zu starten. Der Benutzer muss nun die Elektroden des GSR-Sensors an die mittleren Fingerkuppen des Zeige- und Mittelfinger der schwachen Hand anbringen. Durch Drücken des *START*-Buttons wird die Messung gestartet. Danach wird der Hautwiderstand des Benutzers in Echtzeit in den Graphen eingetragen, der in der Mitte des Screens zu sehen ist. Wenn der Graph sinkt, bedeutet dies, dass die Hautwiderstand sinkt und somit der User emotional erregt wird. Eine Steigung des Graphen hingegen bedeutet einen gesteigerten Hautwiderstand und somit Entspannung. Hierbei sei nochmals angemerkt, dass der Körper ungefähr 4 Sekunden benötigt, dass die Aktivierung sich anhand des Schweißausstoßes feststellen lässt. Ein schnelles Ein- und Ausatmen führt bei den meisten Menschen beispielsweise dazu, dass der Graph nach 4 Sekunden anfängt zu sinken.

Während die Messung läuft kann der User beliebig weiter durch die App navigieren, da der Sensor weiterhin im Hintergrund Messwerte einsammelt. In dieser Zeit der Messung ist sogar für interessantere Ergebnisse angedacht, dass andere Emotionstest durchgeführt werden. Wenn auf die GSRPage zum Graphen zurückgekehrt wird, findet der User dieser wieder im aktuellen Stand und der Graph wird weiterhin aktualisiert. Durch Klicken auf den *STOP*-Button wird die Messung beendet. Durch diesen Button wird auch der Haken für den GSR Test auf der HomePage gesetzt. Der User kann theoretisch zu jeder Zeit eine neue Messung starten und beenden. Jede Messung wird am Ende in der Auswertung berücksichtigt.

Durch Klicken des *DELETE GRAPH*-Buttons kann der User außerdem den Graph zurücksetzen. Dies kann aus Übersichtlichkeitsgründen sinnvoll sein, wenn der User gerne den Graphen neu starten lassen möchte. Das Drücken des Buttons ändert allerdings nicht, dass alle gemessenen Daten an den Decider gesendet werden.

#### 4.4.3 CameraPage

LUKAS SEEMANN

Die CameraPage dient dazu, den Emotionstest der Gesichtserkennung auszuführen. Auch diese Page konnte so umgesetzt werden, wie es in den Mockups geplant wurde. Die GUI der CameraPage ist in Abbildung ? gezeigt.

Mithilfe des *TAKE PICTURE*-Buttons kann der User ein Bild von sich selbst aufnehmen. Beim ersten Benutzen der App wird beim Drücken dieses Buttons ein Hinweis angezeigt, dass der User bestätigen muss, dass die App emoTriX Zugriff auf die Kamera benötigt. Ansonsten wird die Kamera des Smartphone aufgerufen. Anschließend muss ein Bild vom Gesicht des Nutzers aufgenommen, welches im Anschluss von der API analysiert werden soll. Das aufgenommene Bild wird dann, wie rechts abgebildet, auf dem Screen der App eingefügt. Hierbei muss darauf geachtet werden, dass das Gesicht des Users auf dem Bild nach oben zeigt. Ist dies nicht der Fall muss ein neues Bild durch erneutes Drücken

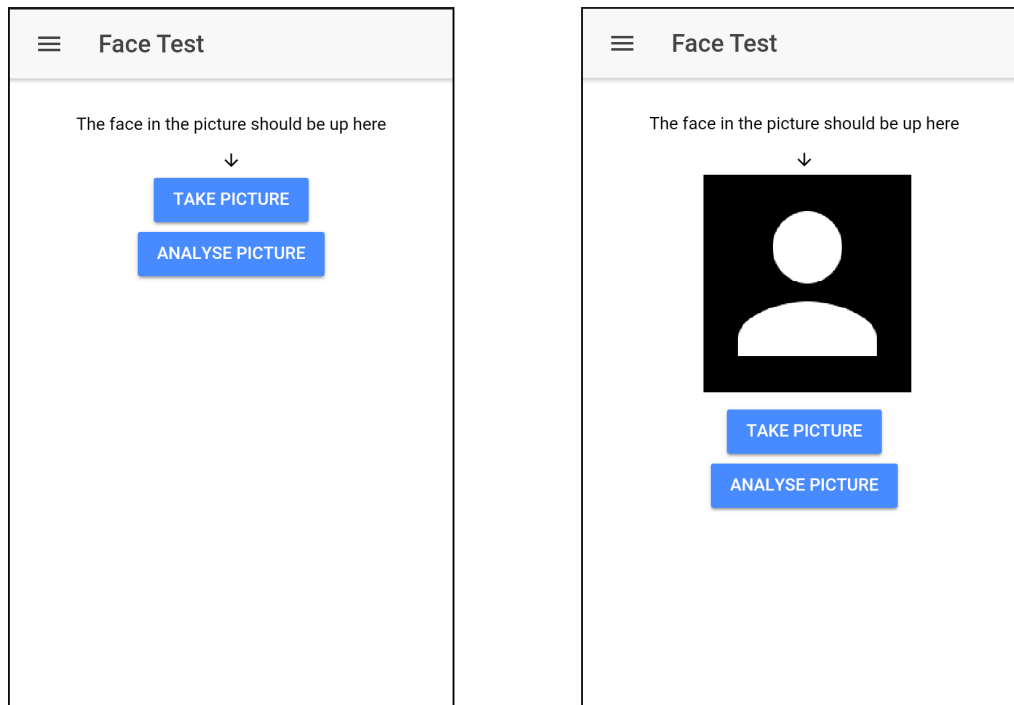


Abbildung 27: GUI der CameraPage

des *TAKE PICTURE*-Buttons aufgenommen und das Smartphone entsprechend gedreht werden, dass das Bild in der richtigen Anordnung in der App angezeigt wird. Die API ist ansonsten nicht in der Lage, das Foto zu analysieren.

Wird das gewünschte Foto in richtiger Anordnung auf dem Screen angezeigt, kann der User mit dem *ANALYSE PICTURE*-Button das Bild an die Microsoft API senden. Anschließend erscheint ein Loading-Anzeige, die anzeigt, dass das Bild übertragen wird. Verschwindet diese Lade-Anzeige ohne Fehlermeldung, wurde das Bild übertragen und das Bild analysiert. Die Ergebnisse sind dann bereits in den Decider eingetragen worden und der Test ist abgeschlossen. Nach einem analysierten Bild wird auch der Teststatus auf der HomePage auf erledigt gesetzt. Natürlich ist es jedoch möglich und auch gewünscht, dass der User mehrere Bilder analysieren lässt. Dadurch wird die Auswertung am Ende umfangreicher und realitätsgetreuer.

#### 4.4.4 DecisionPage

LUKAS SEEMANN

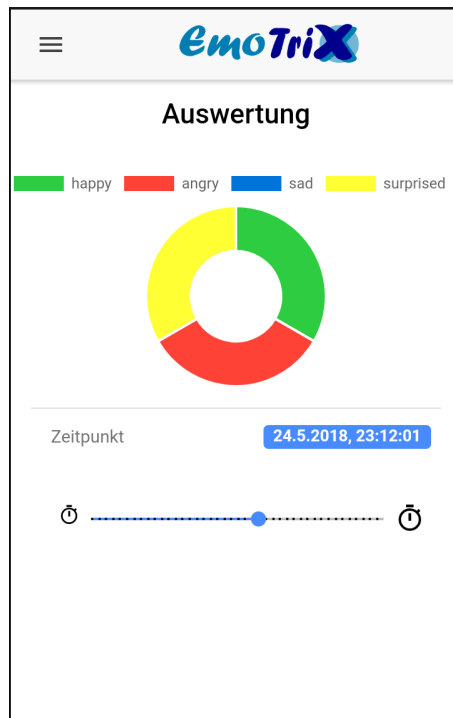


Abbildung 28: GUI der DecisionPage

## 5 Schluss

Hier werden wir darauf eingehen was erreicht wurde was nicht und weshalb nicht.

### 5.1 Anwendungsszenarien

### 5.2 Fazit

### 5.3 Ausblick

# Literatur

- [Ada18] *How are Cellphones Different From Smartphones*. 11. Mai 2018. URL: <https://www.lifewire.com/cell-phones-vs-smartphones-577507>.
- [Akj04] A.K. Jain ; A. Ross ; S. Prabhakar. “An Introduction to Biometric Recognition”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (30. Jan. 2004): *Image- and Video-Based Biometrics*.
- [Ard18a] Arduino. *ARDUINO UNO REV3*. 2018. URL: <https://store.arduino.cc/arduino-uno-rev3> (besucht am 13.05.2018).
- [Ard18b] Arduino. *begin()*. 2018. URL: <https://www.arduino.cc/en/serial/begin> (besucht am 14.05.2018).
- [Bie14] Kai Biermann. “Smartphone - Mächtige Sensoren”. In: *Zeit Online* (28. Mai 2014). URL: <http://www.zeit.de/digital/mobil/2014-05/smartphone-sensoren-iphone-samsung> (besucht am 15.04.2018).
- [Bio18a] *Gangerkennung*. deutsch. bio-metrie.info. 22. Mai 2018. URL: <http://bio-metrie.info/merkmale/gangverhalten.html> (besucht am 22.05.2018).
- [Bio18b] *Tippverhalten*. deutsch. bio-metrie.info. 22. Mai 2018. URL: <http://www.bio-metrie.info/merkmale/tippverhalten.html> (besucht am 22.05.2018).
- [Bou11] Maged N. Kamel Boulos u. a. “How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX”. In: *BioMedical Engineering OnLine* 10.1 (5. Apr. 2011), S. 24. ISSN: 1475-925X. DOI: 10.1186/1475-925X-10-24. URL: <https://doi.org/10.1186/1475-925X-10-24> (besucht am 15.05.2018).
- [Bou88] Wolfram Boucsein. *Elektrodermale Aktivität: Grundlagen, Methoden und Anwendungen*. Springer-Verlag Berlin Heidelberg GmbH, 1988. ISBN: 978-3-662-06969-1.
- [Can01] Ulrich Canzler. “Automatische Erfassung und Analyse der menschlichen Mimik”. In: *Bildverarbeitung für die Medizin 2001*. Hrsg. von Heinz Handels u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, S. 382–386. ISBN: 978-3-642-56714-8.
- [Cla09] Claudia Nickel. “Authentifizierung an mobilen Geräten mittels Gangerkennung”. In: *Datenschutz und Datensicherheit - DuD* 33 (5 17. Juli 2009), S. 280–283. ISSN: 1862-2607. DOI: <https://doi.org/10.1007/s11623-009-0085-4>. URL: <https://link.springer.com/article/10.1007/s11623-009-0085-4> (besucht am 15.05.2018).

- [Com17] *Animojis für das iPhone X: So sehen die bewegten Apple-Emojis aus.* 16. Nov. 2017. URL: <http://www.computerbild.de/artikel/cb-News-Handy-Animoji-iPhoneX-Apple-Karaoke-19268547.html> (besucht am 15.05.2018).
- [Com18] User: Renney77. *Grove GSR Sensor Technical Question (Resistance Equation)*. Community Seedstudio. 14. Feb. 2018. URL: <https://pulsesensor.com/pages/pulsesensor-playground-toolbox> (besucht am 23.05.2018).
- [Dam12] Holger Dambeck. *Herzschlag-Messung - Videosoftware lässt Gesichtsfarbe pulsieren*. SPIEGEL Online. 27. Juni 2012. URL: <http://www.spiegel.de/wissenschaft/technik/video-analyse-software-macht-puls-und-atmung-sichtbar-a-840836.html> (besucht am 19.05.2018).
- [Deu09] Johann Deutsch. *Pädiatrie und Kinderchirurgie*. Stuttgart: Georg Thieme Verlag, 2009. ISBN: 9783131428110.
- [Die06] Franziska Dietz. *Psychologie 1: Methodische Grundlagen und biopsychologische Modelle*. Marburg: MEDI-LEARN Verlag, 2006. ISBN: 978-3-938802-02-1.
- [Dil13] Deborah Bacon Dilts Robert Dilts Judith DeLozier. *NLP II - die neue Generation: Strukturen subjektiver Erfahrung - die Erforschung geht weiter*. Paderborn: Junfermann Verlag GmbH, 2013. ISBN: 9783955712433.
- [Dud18] *Indiz, das*. Deutsch. Duden. 13. Apr. 2018. URL: <https://www.duden.de/rechtschreibung/Indiz> (besucht am 13.04.2018).
- [Ekm92] Paul Ekman. "An argument for basic emotions". In: *Cognition and Emotion* 6.3-4 (1992), S. 169–200. DOI: 10.1080/02699939208411068. eprint: <https://doi.org/10.1080/02699939208411068>. URL: <https://doi.org/10.1080/02699939208411068>.
- [Fer17] Jan-Peter Ferdinand; Tobias Jetzke. "Voice Computing – allgegenwärtige Spracherkennung". In: (Mai 2017).
- [Fraa] *Download der Gesichtserkennungssoftware SHORE®*. 23. Mai 2018. URL: <https://www.iis.fraunhofer.de/de/ff/sse/ils/dl/shore.html>.
- [Frab] *Gesichtserkennung: Schnell, robust und in Echtzeit*. 23. Mai 2018. URL: <https://www.iis.fraunhofer.de/de/ff/sse/ils/tech/shore-facedetection.html>.
- [Gav18] Gavin Phillips. *Smartphones vs. Desktops: Why Is My Phone Slower Than My PC?* Englisch. makeuseof.com. 24. Feb. 2018. URL: <https://www.makeuseof.com/tag/smartphone-desktop-processor-differences/> (besucht am 15.05.2018).

- [Gil14] Andreas Gill. *Runtastic Heart Rate im Praxischeck*. Lifeline. 3. Juni 2014. URL: <https://www.lifeline.de/ernaehrung-fitness/sport/app-test-runtastic-heart-rate-id132003.html> (besucht am 19.05.2018).
- [Gro18] Grove. *Grove - GSR Sensor*. 2018. URL: [http://wiki.seeedstudio.com/Grove-GSR\\_Sensor/](http://wiki.seeedstudio.com/Grove-GSR_Sensor/) (besucht am 07.05.2018).
- [Hen17] Lucia Hennerici. *Video: Pulsmessen - mobil, smart, richtig?* rbb - Rundfunk Berlin-Brandenburg. 30. Aug. 2017. URL: [https://www.rbb-online.de/rbbpraxis/archiv/20170830\\_2015/puls-messen-mobil--smart--richtig-sdg.html](https://www.rbb-online.de/rbbpraxis/archiv/20170830_2015/puls-messen-mobil--smart--richtig-sdg.html) (besucht am 19.05.2018).
- [Hok17] Hokuma. *Plutchik's Wheel of Emotions: What is it and How to Use it in Counseling? The Wheel of Emotions*. English. 14. Dez. 2017. URL: <https://positivepsychologyprogram.com/emotion-wheel/#what> (besucht am 13.04.2018).
- [Hub14] Winke, winke, Passwort. 21. Aug. 2014. URL: <http://www.sueddeutsche.de/digital/sicherheitssoftware-behaviosecc-winke-winke-passwort-1.2096367>.
- [Ion17a] Dan Bucholtz. *Ionic Release Notes*. 8. Nov. 2017. URL: <https://github.com/ionic-team/ionic/releases/tag/v3.9.2> (besucht am 22.04.2018).
- [Ion17b] Jens Stark. *KI-Algorithmus erkennt Emotionen in der Stimme*. Computer-World. 14. Nov. 2017. URL: <https://www.computerworld.ch/technik/forschung/ki-algorithmus-erkennt-emotionen-in-stimme-1444360.html> (besucht am 25.05.2018).
- [Ion18a] Camera. Ionic. 23. Mai 2018. URL: <https://ionicframework.com/docs/native/camera/> (besucht am 23.05.2018).
- [Ion18b] Ionic Framework. *Core Concepts*. 2018. URL: <https://ionicframework.com/docs/intro/concepts/> (besucht am 22.04.2018).
- [Ion18c] Ionic Framework. *Ionic Docs - CLI*. 2018. URL: <https://ionicframework.com/docs/cli/> (besucht am 22.04.2018).
- [Ion18d] Ionic Framework. *Ionic Framework*. 2018. URL: <https://ionicframework.com/framework> (besucht am 22.04.2018).
- [Ion18e] *LoadingController*. Ionic. 24. Mai 2018. URL: <https://ionicframework.com/docs/api/components/loading/LoadingController/> (besucht am 24.05.2018).

- [Jan06] JESSICA LUBIENETZKI DANIEL JANSSEN WOLFGANG I. SCHÖLLHORN. *Diagnose emotionaler Zustände beim Gang mittels neuronaler Netze*. deutsch. Westfälische Wilhelms-Universität Münster, Institut für Sportwissenschaften, 2006. URL: [https://sport.uni-mainz.de/files/2014/05/janssen\\_2006\\_diagnose.pdf](https://sport.uni-mainz.de/files/2014/05/janssen_2006_diagnose.pdf) (besucht am 22.05.2018).
- [Kai18] Kai Schmerer. *Samsung DeX: Galaxy S9+ als PC-Ersatz*. ZDNet. 15. März 2018. URL: <https://www.zdnet.de/88328617/samsung-dex-galaxy-s9-als-pc-ersatz/> (besucht am 15.05.2018).
- [Kar16] Nils Warkentin. *Mimik: Kommunikation ohne Worte*. karrierebibel. 14. Nov. 2016. URL: <https://karrierebibel.de/mimik/> (besucht am 15.05.2018).
- [Kes02] Henrik Kessler u. a. “Facially Expressed Emotion Labeling (FEEL): PC-gestützter Test zur Emotionserkennung”. In: 23 (Jan. 2002). URL: [https://www.researchgate.net/profile/Harald\\_Traue/publication/255638250\\_Facially\\_Expressed\\_Emotion\\_Labeling\\_FEEL\\_PC-gestuetzter\\_Test\\_zur\\_Emotionserkennung/links/0f31753b1190c0350e000000/Facially-Expressed-Emotion-Labeling-FEEL-PC-gestuetzter-Test-zur-Emotionserkennung.pdf](https://www.researchgate.net/profile/Harald_Traue/publication/255638250_Facially_Expressed_Emotion_Labeling_FEEL_PC-gestuetzter_Test_zur_Emotionserkennung/links/0f31753b1190c0350e000000/Facially-Expressed-Emotion-Labeling-FEEL-PC-gestuetzter-Test-zur-Emotionserkennung.pdf) (besucht am 15.05.2018).
- [Kla02] Klaus Scherer. *Sozialpsychologie*. Springer, Berlin, Heidelberg, 2002. Kap. Emotion, S. 165–213. ISBN: 978-3-540-42063-7. DOI: [https://doi.org/10.1007/978-3-662-08008-5\\_6](https://doi.org/10.1007/978-3-662-08008-5_6). URL: [https://link.springer.com/chapter/10.1007/978-3-662-08008-5\\_6](https://link.springer.com/chapter/10.1007/978-3-662-08008-5_6) (besucht am 15.05.2018).
- [Kla05] Klaus R. Scherer. “What are emotions? And how can they be measured?” Englisch. In: *Social Science Information* 44.4 (2005), S. 695–729. DOI: 10.1177/0539018405058216. eprint: <https://doi.org/10.1177/0539018405058216>. URL: <https://doi.org/10.1177/0539018405058216>.
- [Lie14] Julia Müller Alexandra Liedl Christine Knaevelsrud. *Trauma und Schmerz: Manual zur Behandlung traumatisierter Schmerzpatienten*. Stuttgart: Klett-Cotta, 2014. ISBN: 9783608267266.
- [Mdn18] Blob. MDN web docs. 24. Mai 2018. URL: <https://developer.mozilla.org/de/docs/Web/API/Blob> (besucht am 24.05.2018).
- [Mic18a] *Cognitive Services – Gesichtserkennungs-API*. 23. Mai 2018. URL: <https://azure.microsoft.com/de-de/pricing/details/cognitive-services/face-api/>.
- [Mic18b] *Gesichtserkennung*. 23. Mai 2018. URL: <https://azure.microsoft.com/de-de/services/cognitive-services/face/>.



- [MIT15] Holger Dambeck. *Video Magnification*. Massachusetts Institute of Technology (MIT). Juni 2015. URL: <http://people.csail.mit.edu/mrub/vidmag/> (besucht am 19.05.2018).
- [Moe07] Thorsten Möll. *Messung und Wirkung von Markenemotionen: Neuromarketing als neuer verhaltenswissenschaftlicher Ansatz*. Wiesbaden: Deutscher Universitäts-Verlag, Aug. 2007. ISBN: 978-3-8350-0897-7.
- [Pat] Sona Patel u. a. *Acoustic Markers of Emotions Based on Voice Physiology*. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.302.3069#?> (besucht am 16.05.2018).
- [Pet17] Jutta Petters. *Puls messen: Diese 5 Apps haben Deine Herzfrequenz im Blick*. Turn On - Das Saturn Magazin. 16. Dez. 2017. URL: <https://www.turn-on.de/lifestyle/topliste/puls-messen-diese-5-apps-haben-deine-herzfrequenz-im-blick-333327> (besucht am 19.05.2018).
- [Pre16] Preventius GmbH. *Weltweit erste medizinisch zugelassene App erkennt zuverlässig Herzrhythmusstörungen - einfach per Smartphone ohne Zubehör*. Preventius GmbH. 25. Aug. 2016. URL: <https://www.presseportal.de/pm/121571/3412209> (besucht am 19.05.2018).
- [Rab06] Eberhard Rabe. *Praktische Phlebologie*. Stuttgart: Thieme Verlagsgruppe, 2006. Kap. Photoplethysmographie. ISBN: 9783131192325.
- [Rei18] Reichelt Elektronik. *ARD SHD BASE V2*. 2018. URL: <https://www.reichelt.de/Entwicklerboard-Zubehoer/ARD-SHD-BASE-V2/3/index.html?ACTION=3&GROUPID=8244&ARTICLE=191204> (besucht am 13.05.2018).
- [Ros13] Sandro Wefel Manfred Rost. *Elektronik für Informatiker: Von den Grundlagen bis zur Mikrocontroller-Applikation*. München: Oldenbourg Wissenschaftsverlag GmbH, 2013. ISBN: 9783486706925.
- [Sam06] Sascha Müller. *Bewertung der Qualität von Unterschriften für die biometrische Erkennung*. deutsch. Diplomarbeit. TU-Darmstadt, 26. Sep. 2006. Kap. 1.1.1 Biometrische Merkmale, S. 1+2. 105 S. URL: <https://www.seceng.informatik.tu-darmstadt.de/assets/mueller/muellerdiplom.pdf> (besucht am 15.05.2018).
- [Sav18] *Kausalität*. Deutsch. Spektrum akademischer Verlag, Heidelberg. 15. Apr. 2018. URL: <https://www.spektrum.de/lexikon/physik/kausalitaet/7841> (besucht am 15.04.2018).
- [Sch12] Dipl. Math. Michael Wand Prof. Dr. Tanja Schultz. *Biosignale und Benutzerschnittstellen - Elektrodermale Aktivität (EDA)*. Universität Bremen, WS Vorlesung 2012/2013: Cognitive Systems Lab, 2012.

- [Sch14] Ralf Stürmer Jennifer Schmidt. *Erfolgreiches Marketing durch Emotionsforschung: Messung, Analyse, Best Practice*. Freiburg: Haufe-Lexware GmbH & Co. KG, 2014. ISBN: 978-3-648-04894-8.
- [Sea18] SearchSecurity.de. *Definition Biometrie*. 17. Apr. 2018. URL: <https://www.searchsecurity.de/definition/Biometrie> (besucht am 17.04.2018).
- [Sin16] *Testing Project Oxford OCR : How to use a local file ? (in base64 for example)*. social.msdn.microsoft.com. 9. Jan. 2016. URL: <https://social.msdn.microsoft.com/Forums/sqlserver/en-US/807ee18d-45e5-410b-a339-c8dcb3bfa25b/testing-project-oxford-ocr-how-to-use-a-local-file-in-base64-for-example> (besucht am 24.05.2018).
- [SK98] Klaus R. Scherr Susanne Kaiser. "Models of "Normal" Emotions Applied to Facial and Vocal Expression in Clinical Disorders". Englisch. In: *Emotions in Psychopathology*. Oxford University Press., 1998, S. 81–98. ISBN: isbn. DOI: doi. URL: <http://www.unige.ch/fapse/emotion/members/kaiser/article98.pdf> (besucht am 17.05.2018).
- [Sou18] Souq.com. *Arduino Uno - R3*. 2018. URL: <https://uae.souq.com/ae-en/arduino-uno-r3-6186780/i/> (besucht am 13.05.2018).
- [Spe18] Spektrum.de. *Puls*. Spektrum.de. 15. Apr. 2018. URL: <https://www.spektrum.de/lexikon/biologie/puls/54932> (besucht am 15.04.2018).
- [Sta18a] Statista GmbH. *Anzahl der Smartphone-Nutzer in Deutschland in den Jahren 2009 bis 2018 (in Millionen)*. Bitkom Research; comScore. 2018. URL: <https://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonennutzer-in-deutschland-seit-2010/> (besucht am 15.04.2018).
- [Sta18b] Statista GmbH. *Prognose zum Anteil der Smartphone-Nutzer an den Mobiltelefonnutzern weltweit von 2014 bis 2020*. eMarketer. 2018. URL: <https://de.statista.com/statistik/daten/studie/556616/umfrage/prognose-zum-anteil-der-smartphone-nutzer-an-den-mobiltelefonnutzern-weltweit/> (besucht am 15.04.2018).
- [Sta18c] Statista GmbH. *Prognose zur Anzahl der Smartphone-Nutzer weltweit von 2012 bis 2020 (in Milliarden)*. eMarketer. 2018. URL: <https://de.statista.com/statistik/daten/studie/309656/umfrage/prognose-zur-anzahl-der-smartphone-nutzer-weltweit/> (besucht am 15.04.2018).
- [Tho11] F. Thoemmes. *Ausgewählte Kausalitätstheorien im Vergleich*. English. Institut für Erziehungswissenschaft Abteilung Empirische Bildungsforschung und Pädagogische Psychologie, 2011. URL: [https://www.human.cornell.edu/sites/default/files/HD/qml/Thoemmes\\_2011.pdf](https://www.human.cornell.edu/sites/default/files/HD/qml/Thoemmes_2011.pdf) (besucht am 15.04.2018).

- [Til11] Gereon Tillenburger. “Stimmt die Stimme?” In: *Datenschutz und Datensicherheit - DuD* 35.3 (2011), S. 197–199. ISSN: 1862-2607. DOI: 10.1007/s11623-011-0049-3. URL: <https://doi.org/10.1007/s11623-011-0049-3> (besucht am 16.05.2018).
- [Til17] Janna Tillmann. *Was ist eigentlich Typescript?* t3n News. 15. Sep. 2017. URL: <https://t3n.de/news/eigentlich-typescript-859869/> (besucht am 22.04.2018).
- [Tro13] Matthias Trojahn; Florian Arndt; Markus Weinmann; Frank Ortmeier. “Emotion Recognition through Keystroke Dynamics on Touchscreen Keyboards”. englisch. In: *Proceedings of the 15th International Conference on Enterprise Information Systems*. Bd. 3. 2013, S. 31–37. ISBN: 978-989-8565-61-7. DOI: 10.5220/0004415500310037. URL: <http://www.scitepress.org/PublicationsDetail.aspx?ID=P4v2efGXqzQ%3d&t=1> (besucht am 22.05.2018).
- [Wal84] Richard D. Walk und Carolyn P. Homan. “Emotion and dance in dynamic light displays”. In: *Bulletin of the Psychonomic Society* 22.5 (1984), S. 437–440. ISSN: 0090-5054. DOI: 10.3758/BF03333870. URL: <https://doi.org/10.3758/BF03333870>.
- [Web18] Merriam-Webster. *sensor*. Merriam-Webster. 4. Apr. 2018. URL: <https://www.merriam-webster.com/dictionary/sensor> (besucht am 15.04.2018).
- [Wed12] Klaus Wedekind. *App analysiert Stimme: Smartphone erkennt Emotionen*. ANDROIDPIT. 5. Dez. 2012. URL: <https://www.androidpit.de/smartphones-erkennen-emotionen> (besucht am 25.05.2018).
- [Wik18] Wikipedia. *Ionic Logo*. 2018. URL: [https://de.wikipedia.org/wiki/Ionic\\_\(Framework\)#/media/File:Ionic\\_Logo.svg](https://de.wikipedia.org/wiki/Ionic_(Framework)#/media/File:Ionic_Logo.svg) (besucht am 22.04.2018).
- [Wor18a] World Famous Electronics llc. *Pulse Sensor*. World Famous Electronics llc. 2018. URL: <https://pulsesensor.com/> (besucht am 19.05.2018).
- [Wor18b] World Famous Electronics llc. *PulseSensor Playground Toolbox*. World Famous Electronics llc. 2018. URL: <https://pulsesensor.com/pages/pulsesensor-playground-toolbox> (besucht am 19.05.2018).

## Anhänge