UNIVERSITÄT SIEGEN

Durchgeführt im Rahmen des

BMBF-Projekt: ELISE

Dokumentation der Projektarbeit

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

WiSe 2017/2018 und SoSe 2018

Projektbetreuer:

Medizinische Informatik und Mikrosystementwurf
Prof. Dr. rer. nat. Rainer Brück
Dr.-Ing. Armin Grünewald
M.Sc. David Krönert
M.Sc. Tanja Eiler

Forschungsgruppe für Mustererkennung
Prof. Dr.-Ing. Marcin Grzegorzek
M.Sc. Frédéric Li

Projektteilnehmer:

Artur Piet (Sprecher der Projektgruppe)
Jonas Pöhler (Stellv. Sprecher der Projektgruppe)
Arnaud Eric Toham Waffo
Boris Kamdem
Kevin Orth
Meryem Dural
Minas Michail

Inhaltsverzeichnis

1 Einleitung

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

1.1 Hintergrund und Motivation

1.2 ELISE Projektbeschreibung

1.3 Gliederung dieser Dokumentation

1.4 Anhang

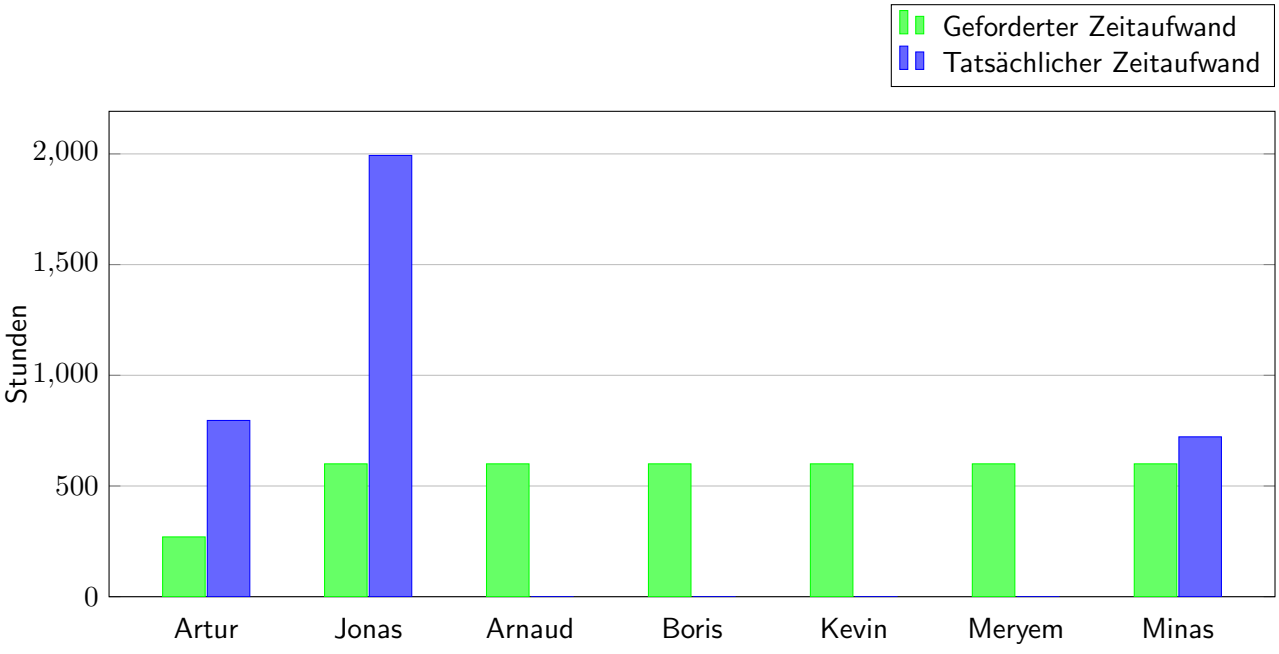
2 Organisation

Verantwortlich: Artur

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

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

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



2.1 Verantwortungsbereiche

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

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

2.2 Gruppentreffen

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

3 Grundlagen

Verantwortlich: Arnaud

3.1 Definition von Emotionen

Verantwortlich: Arnaud

3.2 Virtual Reality (VR)

Verantwortlich: Arnaud, Boris

Virtuelle Realität wird helfen, besser den Emotionen zu erliegen, die wir uns wünschen. Es kann als Variante für die üblichen Modelle gesehen werden, bei denen Testpersonen vor Bildschirmen mit emotionalen Videos platziert werden.

3.3 Sensoren und biophysiologische Signale zur Emotionserkennung

Verantwortlich: Kevin

3.3.1 Körpertemperatur-Sensor

3.3.2 Blood Volume Pulse-Sensor (BVP)

3.3.3 Messen der Sauerstoffsättigung (SpO2)

3.3.4 Galvanic Skin Response (GSR)

3.3.5 Elektroenzephalografie (EEG)

3.3.6 Elektrookulografie (EOG)

3.3.7 Analog/Digital-Wandler

3.4 Kommunikation

Verantwortlich: Kevin, Jonas

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

Die Implementierung erfolgte schlussendlich mit UDP Broadcasts und einem rudimentären Kommunikationsprotokoll. Für die Übertragung war es wichtig, dass die

Implementierung sowohl auf Seiten der Unreal Engine, als auch beim Mikrocontroller und bei der Kontrollanwendung einfach und mit geringem Arbeitsaufwand möglich ist. Zugleich sollte auf eine einfache Wartbarkeit und die Möglichkeit die Verbindung "live" zu debuggen geachtet werden.

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

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

(BILD EINFÜGEN UNTERSCHIEDLICHE DATENPAKETE)

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

3.5 Grundlagen der Mustererkennung

Verantwortlich: Artur
- Bereit zum Korrekturlesen.

Mustererkennung (enlg. "pattern recognition") ist ein Unterthema des machinellen Lernens. Das Ziel besteht darin, automatisierte Systeme zu entwerfen, die hoch abstrakte Muster in Daten erkennen können. Konkret heißt dies, dass man Maschinen beibringen möchte komplexer Aufgaben zu lösen, welche vom Menschen nahzu mühlelos und natürlich erledigt werden können. Typische Beispiele für die zahlreichen Anwendungsbereiche sind die Objekterkennung, Spracherkennung sowie die Erkennung und Verfolgung in Bildern. Die Emotionserkennung ist ein Anwendungsbereich der Mustererkennung. Die Hauptidee hinter der Lösung eines Mustererkennung-

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

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

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

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

Im Rahmen des ELISE-Projekts werden mit Hilfe von Mustererkennungsverfahren eindimensionale Zeitsignale von physiologischen Sensoren in Echtzeit für die Erkennung von drei Emotionen verarbeitet: Glück, Frustation und Langeweile. Um den Emotionsklassifizierer aufzubauen, wird ein standardmäßiger, überwachter Lernansatz namens Emotion Recognition Chain verwendet, der im folgendem Kapitel beschrieben wird.

3.6 Emotion Recognition Chain

Verantwortlich: Artur
- Bereit zum Korrekturlesen.

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

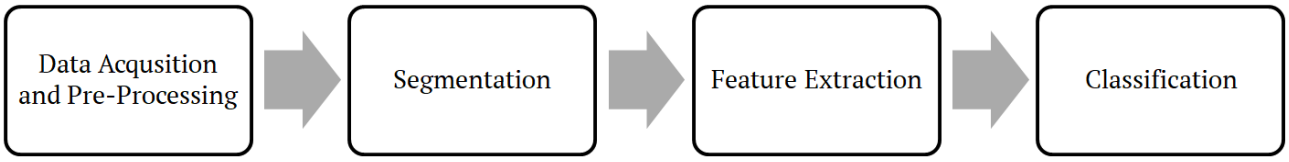


Abbildung 1: Emotion Recognition Chain: Zeitreihen-Datensätze werden von tragbaren Sensoren aufgenommen (Datenerfassung) und vorverarbeitet (Vorverarbeitung). Die Daten werden dann in Segmente unterteilt (Segmentierung), aus denen Merkmale extrahiert werden (Merkmalsextraktion). Mit den gewonnenen Merkmalen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

3.6.1 Datenerfassung

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

3.6.2 Vorverarbeitung

Das Ziel der Vorverarbeitung ist die "Verbesserung" der Daten für die nachfolgenden Schritte der ERC. In der Regel ist es dadurch besser möglich Muster in Daten erkennen zu können. Vorverarbeite Daten erreicht man durch Anwenudng von z.B. Filterung (Rauschunterdruckung), Normierung oder Reduzierung von unerwünschten oder unbedeutenden Datenteilen.

3.6.3 Segmentation

Ziel dieses Schrittes ist es, Teile von Daten zu identifizieren, welche wichtige Informationen über die zu erkennenden Emotionen enthalten. Dies geschieht durch

Filtern der Daten und Ausschließen von Segmenten, die für das Klassifizierungsproblem nicht relevant sind. Zusätzlich wird die zu verarbeitende Datenmenge reduziert, indem Segmente eines Zeitfensters fester Größe aus den Daten extrahiert werden. Diese Vorgeheisweise ist heute in der Praxis besonders wichtig, da sonst hardwarebedingte Einschränkungen die zu verarbeitende Datenmenge begrenzen könnten.

3.6.4 Merkmalsextraktion

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

3.6.5 Klassifikation

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

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

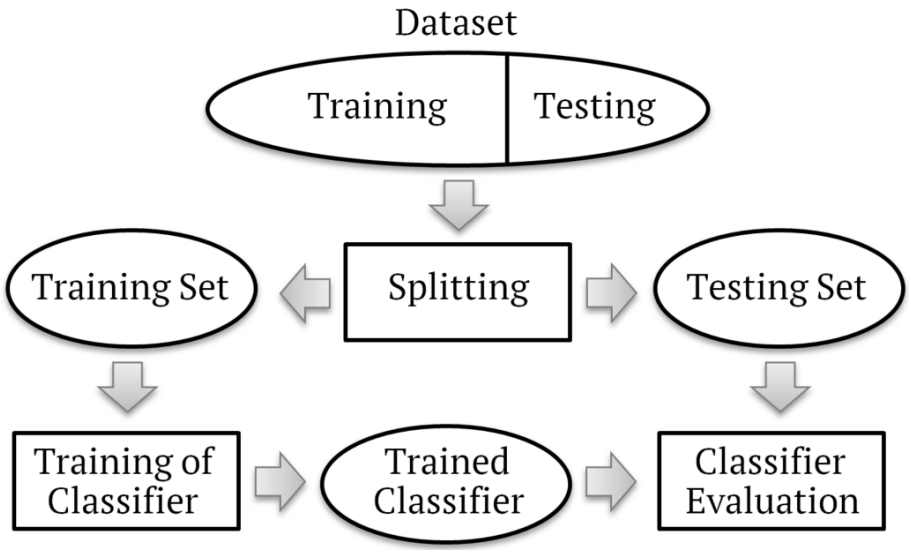


Abbildung 2: Aufteilen eines Datensets in ein Trainings- und ein Testset. Das Trainingsset dient zum Erstellen und Trainieren eines Klassifikators. Die Performance wird mithilfe des Testsets ermittelt und bewertet.

Für die Klassifizierung wurde der state-of-the-art Klassifikator Support-Vector-Machine (SVM) verwendet. SVM ist ein überwachtes Lernansatz, der für binäre Klassifikation oder Regressionszwecke genutzt werden kann. Das Ziel des SVM ist es eine Hyperebene (engl. "hyperplane") zu finden, die zwei Klassen im Raum der Merkmale trennt und gleichzeitig den Abstand (engl. "margin") zwischen der Hyperebene und den nächsten Datenpunkten jeder Klasse maximiert. Für jeden Datenpunkt (dargestellt durch den entsprechenden Vektor von Merkmalen) den wir klassifizieren möchten, wird ein Klassenettiket vergeben, je nachdem, zu welcher Seite der Hyperebene es gehört. Die Methode hat ihren Namen von den "Stützvektoren" (engl. "support vectors"), welche die nächstgelegenen Vektoren beider Klassen zur trennenden Hyperebene sind. Cortes und Vapnik zeigten in [2], dass die Gleichung der optimalen Hyperebene nur von diesen spezifischen Vektoren abhängig ist. Abbildung 3.6.5 zeigt eine optimale Hyperebene in 2D, die beide Klassen perfekt teilt. Datenpunkte werden durch nicht-ausgefüllte blaue Dreiecke bzw. rote Kreise für beide Klassen dargestellt, während Unterstützungsvektoren durch ausgefüllte Punkte und Kreise hervorgehoben werden.

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

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

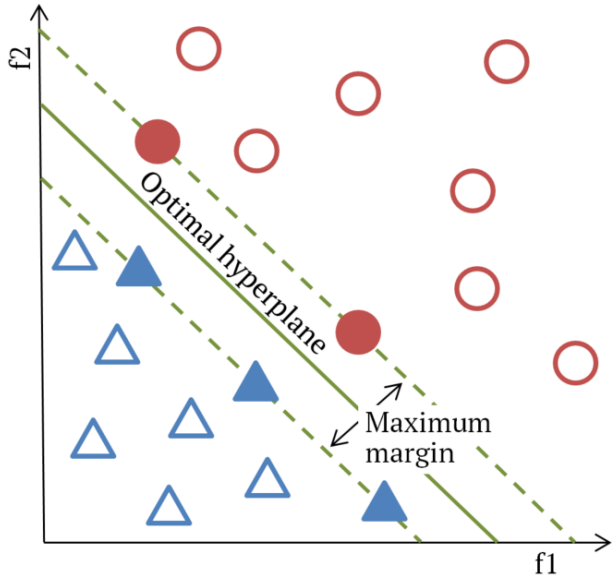


Abbildung 3: Beispiel eines SVM-Klassifikators im zweidimensionalem Merkmalsraum: Die Datenpunkte beider Klassen (dargestellt durch nicht-ausgefüllte blaue Dreiecke und rote Kreise) sind durch eine Hyperebene getrennt, welche die Margin maximiert. Stützvektoren werden als ausgefüllte Dreiecke und Kreise dargestellt.

Dies wird für jede Klasse wiederholt, was zu Z Klassifikatoren führt, d.h. einen für jede Klasse.

In der Praxis sind die Daten durch normales SVM fast nie linear trennbar. Einer der Hauptgründe für die Beliebtheit von SVM ist jedoch die Möglichkeit es so genannten Kernel-Tricks. Es basiert auf der Annahme, dass nicht-linear trennbare Daten linear trennbar werden können, wenn sie in einen Raum höherer Dimension projiziert werden. In [2] zeigten Cortes und Vapnik, dass die SVM-Klassifikationsentscheidungsfunktion als gewichtete Summe von Skalarprodukten zwischen Stützvektoren und dem Vektor der zu klassifizierenden Merkmale ausgedrückt werden kann. Der Kernel-Trick nutzt dies aus, indem er eine Kernelfunktion einführt, die ein skalares Produkt im hochdimensionalen Zielraum repräsentiert. Diese Kernelfunktion erübrigt die eigentliche Zuordnung zwischen dem ursprünglichen und dem hochdimensionalen Feature-Raum. Außerdem ist die Verwendung des Kernel-Tricks oft rechengünstiger als andere Alternativen. Die beiden beliebtesten Kernel sind der lineare und der Radial Basis Function (RBF) Kernel, definiert als:

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

wobei x und x' Vektoren des Merkmalsraums bezeichnen und γ der Parameter ist, der die "Ausbreitung" (engl. "Spread") des Kernels definiert.

Normales SVM verwendet sogenannte hard-margins, die versuchen, eine optimale Hyperebene zu schaffen, die keine Fehlklassifizierungen zulässt. Es ist jedoch oft besser, einige Klassifizierungsfehler zuzulassen, um eine Überanpassung zu verhin-

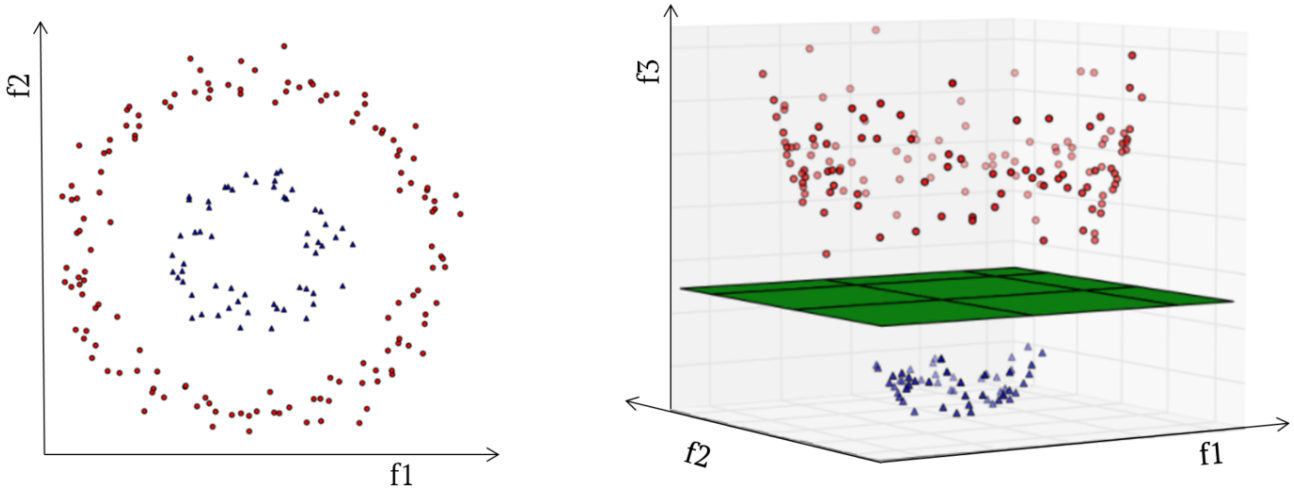


Abbildung 4: Beispiel für die Verwendung des Kernel-Tricks: Nicht-linear trennbare Daten in einem 2D-Merkmalsraum (links) können linear trennbar gemacht werden, wenn sie auf 3D projiziert werden (rechts) [3].

dern und eine generalisierte Hyperebene zu erhalten. Diese allgemeinere Hyperebene liefert deutlich bessere und zuverlässigere Ergebnisse, wenn sie auf neue und unbekannte Datensätze angewendet wird. Hard-margin SVM kann zu einem Modell führen, das für die Trainingsdaten perfekt funktioniert, aber bei anderen Datensätzen sehr schlecht, weil es seine Trainingsdaten "zu gut" gelernt hat. Aus diesem Grund haben Cortes und Vapnik in [2] eine Variante des Standard-SVM-Klassifikators namens soft-margin SVM (oder auch C-SVM genannt) eingeführt, die eine Fehlklassifizierung von Beispielen beim Erstellen der trennenden Hyperebene toleriert. Der soft-margin Parameter C wird genutzt um die Anzahl der Fehlklassifikationen festzulegen. Je größer der Wert von C, desto weniger Fehleinstufungen sind zulässig. Umgekehrt erlauben kleine Werte von C mehr Fehlklassifizierungen, um die Verallgemeinerungsfähigkeit des Klassifikators zu verbessern.

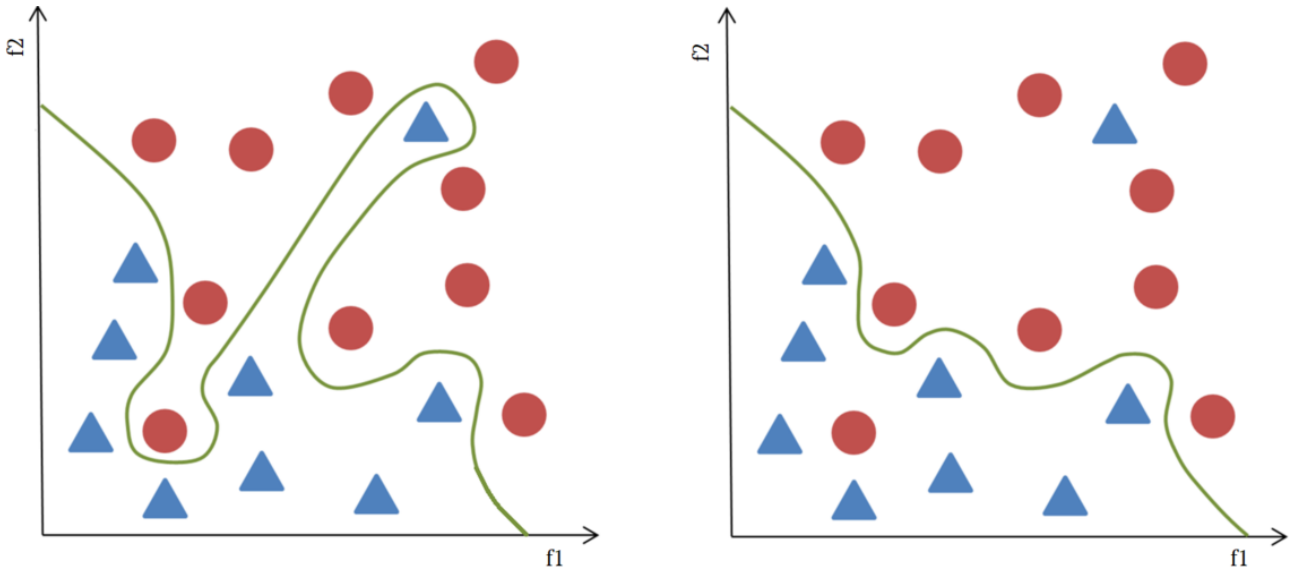


Abbildung 5: Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalsraum: Der hard-margin SVM trennt die beiden Klassen perfekt im Gegensatz zum Soft-Margin SVM, der einige Fehlklassifikationen zulässt.

4 State-of-the-Art Analyse

Verantwortlich: Jonas
- Next Step: Gliederung erstellen

Part I

Erster Prototype

5 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

5.1 Anforderungen

5.2 Konzept

5.3 Hardwareauswahl

5.3.1 Auswahlkriterien

5.3.2 Festlegung der genutzten Hardware

5.4 Hardwarearchitektur

5.4.1 GSR-Sensor

5.4.2 Temperatur-Senosr

5.4.3 Pulsoximeter

5.4.4 EEG

5.4.5 EOG

5.4.6 Datenübertragung

5.5 Programmierung

5.6 Aufnahme der übertragenen Daten

6 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

7 Emotionsinduktion

Verantwortlich: Minas

7.1 Ablauf

Verantwortlich: Minas

7.2 Fragebogen

Verantwortlich: Boris

Ein Fragebogen kann als Formular definiert werden, das einen Satz von Fragen enthält, die für einen bestimmten Zweck definiert wurden [4]. Hier dient der Fragebogen als Befragungsinstrument für die Realisierung unseres Projekts. Hierfür wurden für die Ausarbeitung die Hauptkomponenten definiert, die Auskunft über das Problem der Studie und über die Versuchsperson geben.

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

(Bild für den ersten Fragebogen)

7.3 Szenarien

Verantwortlich: Meryem

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

7.3.1 Glück

Verantwortlich: Minas

7.3.2 Langeweile

Verantwortlich: Boris

Langeweile wird im Allgemeinen als eine Emotion wahrgenommen, die man erlebt, wenn man einen Zustand durchläuft, an dem man kein Interesse hat [5]. Dieses Gefühl charakterisiert Inaktivität, Unproduktivität und kann manchmal zu Melancholie oder Traurigkeit führen. In dieser Studie wird das Erwachen der Langeweile durch eine grüne rotierende Scheibe mit einem schwarzen Strahl verursacht.

Für das zweite Prototyp wird zwei Szenarios benutzt um die Langweile bei Probanden zu erwecken. Als ersten wurde eine 5 minütigen Video über die "Latente Steuer im Jahres Abschluss" von der Tax Universität benutzt um diese Emotion auszulösen. Dabei ging es um eine kurze Einführungsvideo über Auswirkung von Latentesteuer in der Jahresbilanz. Als zweites Szenario würde das "Peg-Turning" Spiel benutzt. Sein Prinzip besteht darin, eine nach alle 15 Sekunden (oder etwas länger) drehende kreisförmige grüne Scheibe zuzuschauen.

(Bild für Langweile (Pec-turning game))

7.3.3 Frustration

Verantwortlich: Meryem

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

Für das Frustrationsexpermiment der Studie soll auf das Auslösen der Misserfolgserlebnisse sowie eine empfundene Ungerechtigkeit zurückgegriffen werden. Dem Probanden wird die Aufgabe gestellt, in eine VR-Umgebung das Spiel "heißer Draht" zu spielen. In diesem Spiel besteht die Aufgabe darin, den Ring, welcher sich in

der Hand des Spieler befindet, von dem Startpunkt zum Endpunkt eines Drahts zu befördern. Der Ring darf währenddessen den Draht nicht berühren. Dieses Spiel ist durch die geschwungene Form des Drahts schwierig und fordert daher viel Ruhe und Geschick. Dem Probanden wird eine Zeit vorgegeben, in welcher dieser von dem Startpunkt zum Endpunkt gelangen muss. Der Faktor des Zeitdrucks kann eine Stressreaktion auslösen. Da das Spiel jedoch den Ehrgeiz erwecken kann, kann es dazu führen, dass es nicht zu der erhofften Frustration kommt. Um diese Emotion trotzdem einleiten zu können, wird das Spiel an einigen Stellen so prepariert, dass der Proband durch Spielfehler den Draht berührt und erneut von dem Startpunkt aus starten muss. Die empfundene Ungerechtigkeit und Ohnmächtigkeit des Probanden soll als Indikator der Frustration dienen.

8 Messreihe

Verantwortlich: Kevin, Artur

9 Mustererkennung

Verantwortlich: Artur

In diesem Kapitel werden alle Schritte entlang der ERC (vgl. Kapitel 3.6) konkretisiert und es wird detailreich beschrieben, wie wir vorgegangen sind.

9.1 Datenerfassung

In Kapitel 3.6.1 wurde das verwendete Datenset bereits detailiert beschrieben, sodass hier darauf verzichtet wird.

9.2 Vorverarbeitung

Wie bereits in Kapitel 3.6.2 beschrieben, ist das Ziel der Vorverarbeitung die "Verbesserung" der Daten für die nachfolgenden Schritte der ERC. Im Rahmen des ELISE Projektes wurden Normalisierungstechniken auf dem gesamten Datensatz angewendet. Wir haben insbesondere die Standardnormalisierung verwendet, welche den Mittelwert der Daten auf Null setzt und die Einheitsvarianz ergibt [6]. Die Formel für die Standardnormierung lautet:

$$x' = \frac{x - \overline{x}}{\sigma} \tag{2}$$

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

9.3 Segmentation

Wie bereits in Kapitel 3.6.3 beschrieben, ist das Ziel der Segmentation Teile von Daten zu identifizieren, welche wichtige Informationen über die zu erkennenden Emotionen enthalten. In dieser Projektarbeit wurde ein Schiebefensteransatz (engl. "sliding window approach") verwendet. Ziel der Methode ist die Segmentierung der vorhandenen Daten in kleinere Einheiten, um die Merkmalsextraktion sowie die anschließende Klassifizierung zu vereinfachen oder gar erst zu ermöglichen. Die Länge des Zeitfensters (engl. "time window") und des Gleitschritts (engl. "sliding stride") sind zu bestimmende Parameter (und werden auch als "'Hyperparameter" bezeichnet), wobei sich das Zeitfenster auf die feste Größe pro extrahiertem Segment und der Gleitschritt auf den Abstand zu dem Beginn des darauf folgenden Zeitfensters bezieht. Es ist zu beachten, dass sich aufeinanderfolgende Zeitfenster überlappen können, sobald der definierte Gleitschritt kleiner als das Zeitfenster ist.

Die Daten werden auf Zeitstempel-Ebene mit Etiketten beschriftet, basierend auf den von der jeweiligen Versuchsperson ausgefüllten Fragebögen. Jedem Zeitfenster wird ein Etikett zugeordnet, welches das dominate (d.h. am meisten vohandene) Etikett der im entsprechendem Fenster enthaltenen Zeitstempel basiert. Es wird davon ausgegangen, dass jedes Zeitfenster nur von einer Emotion belegt ist.

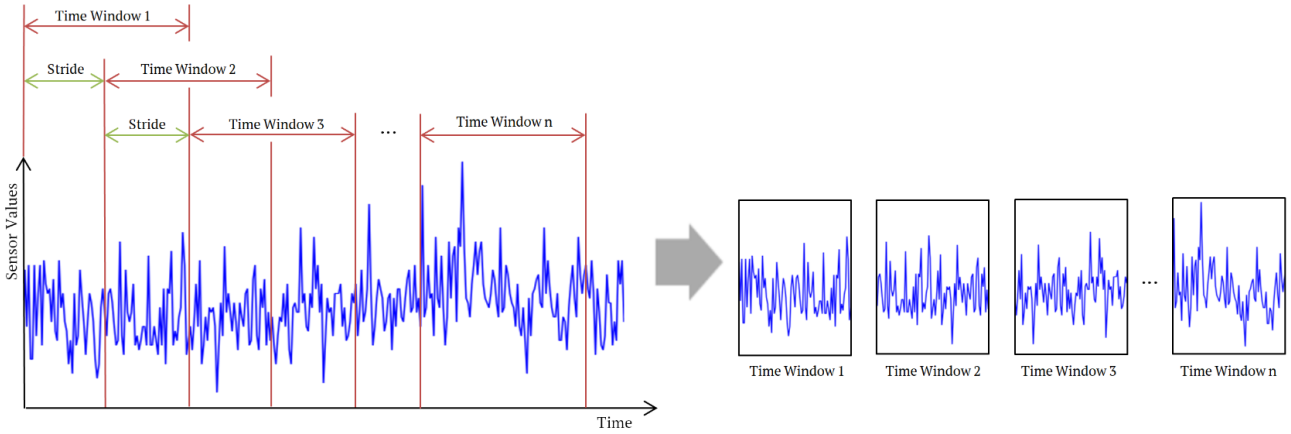


Abbildung 6: Schiebefenster-Segmentierung: Die Daten werden durch ein Zeitfenster fester Größe in kleinere Segmente aufgeteilt. Das Fenster wird mit einem festen Gleichschritt geschoben, um den aufeinanderfolgend Daten-Zeitfenster zu erhalten.

9.4 Merkmalsextraktion

Wie bereits in Kapitel 3.6.4 beschrieben, ist das Ziel der Merkmalsextraktion Charakteristiken und Merkmale in den Daten zu finden, die für das Klassifizierungsproblem von möglichst hoher Relevanz sind. Im Rahmen des ELISE Projektes haben wir verschiedene Vorgehensweisen angewendet. Im den folgenden Unterkapiteln werden diese vorgestellt.

9.4.1 Handgefertigte Merkmale

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

Obwohl frühere Forschungsarbeiten schon handgefertigte Merkmale zur Emotionserkennung unter mithilfe physiologischer Signale getestet haben (vgl. [7]), wurde dieser Ansatz noch nie für die Erkennung dieser spezifischen Emotionen unter Verwendung dieser Kombination von Sensoren getestet. Zusätzlich haben wir zuerst

handgefertigte Merkmale getestet, um ein Basisergebnis zu liefern, mit der die Ergebnissen der anderen Ansätze vergleichen werden können. Handgefertigte Merkmale sind in der Regel entweder einfache statistische Werte, Fourier-basierte oder selbstentwickelte Merkmale sein, die aufgrund von Vorkenntnissen der Daten verwendet werden. Diese Arbeit wurden statistische, Fourier-basierte und selbstentwickelte Merkmale getestet.

Statistische Merkmale

Die Tabelle 1 fasst die elf verschiedenen und in der Studie verwendeten statistischen Merkmale zusammen [8]. Wir bezeichnen $\mathbf{x} = (x_1, x_2, ..., x_T)$ als Vektor, der die in einem Datenzeitfenster der Länge T enthaltenen Sensorwerte für einen Sensorkanal darstellt.

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

Tabelle 1: Statistische Merkmale, die im Rahmen des ELISE-Projektes verwendet wurden.

Fourier-basierte Merkmale

Die in diesem Kapitel präsentierten Ergebnisse wurden in einer Bachelorarbeit [9] ermittelt, die auch ihm Rahmen des ELISE-Projektes geschrieben wurde.

Die Transformation von Zeitsignalen im Frequenzbereich bei Studien zu einem gängigen Werkzeug für die Analyse von Zeitreihendaten geworden. Es basiert auf einem mathematischen Prozess namens Fouriertransform, wo ein Signal in einer unendlichen gewichteten Summe von Sinus- und Coisnewellen unterschiedlicher

Frequenz zu zerlegen. Bei einer kontinuierlichen Funktion $f : t \rightarrow f(t)$ ist der Fouriertransformation $F(w)$ eines Signals f:

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

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

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

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

$$p_k(x) = \Re(X_k^2) + \Im(X_k^2) \tag{5}$$

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

Um frequenzbezogene Merkmale zu extrahieren, wird das Leistungsspektrum der Signale in vier Frequenzbänder gleicher Länge unterteilt. Der Mittelwert, die Standard-Abweichung, das Maximum und das Minimum wurde dann auf jedem Frequenzband für jeden Sensorkanal extrahiert, was zu einem Prozess der Merkmalsextraktion aus normalisierten Datenrahmen führte, wie in Abbildung 7 dargestellt.

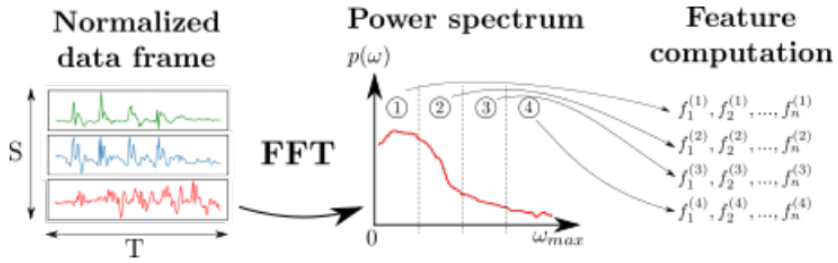


Abbildung 7: Merkmalsextraktion aus frequenzbezogener Domain.

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

im Frequenzbereich pro Zeitfenster.

Selbstentwickelte Merkmale
Es wurden zwei eigene Merkmale definiert [8]. Nulldurchgang (engl. "zero crossing") und Anzahl der Spitzen (engl. "number of peaks"). Im Folgendem werden diese beiden Merkmale detailiert beschrieben.

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

Das Spitzenzähler-Merkmal bestimmt die Anzahl von loklanen Hochpunkten im Zeitsignal. Alle lokalen Maximen sind durch einen Onset (Startpunkt), eine Spitze und einen Offset (Endpunkt) gekennzeichnet (vgl. Abbildung 8). Jedes Vorkommen einer Onset/Offset-Paarung wird hierbei als Spitze gezählt. Onsets, Spitzen und Offsets werden durch die folgenden Operationen identifiziert (vgl. [10]):

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

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

9.4.2 Codebook Approach

Die bisherige Methodik mit handgefertigten Merkmalen ist klassisch für den überwachten Lernansatz. Es existieren aber auch einige Nachteile. Das Hauptproblem besteht darin, dass nicht sichergestellt werden kann, dass die gewählten Merkmale die besten Klassifizierungsergebnisse erzielen. Damit besteht immer die Gefahr, dass

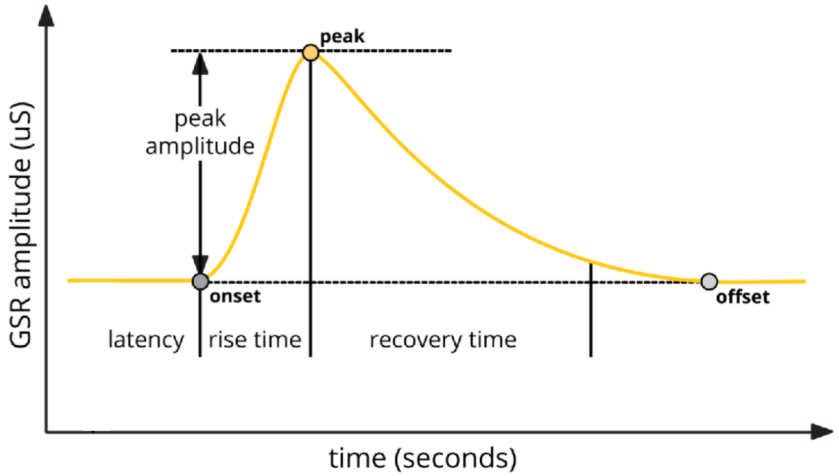


Abbildung 8: Spitzenzähler-Merkmal: Onset (Startpunkt), Spitze und Offset (Endpunkt). Jedes Paar von Onset/Offset erhöht die Anzahl der Spitzen um eins.

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

K. Shirahama et al. [11] schlugen eine unüberwachte Merkmalsextraktionsmethode namens Codebook Approach (CA) vor, um Merkmale aus 1D-Zeitreihensignalen zu erzeugen. Der CA hat den Vorteil, dass formbasierte Merkmale gefunden werden können, die für das Problem der Emotionserkennung relevant sind, aber weder offensichtlich noch leicht als Mensch zu interpretieren sind. Der CA besteht aus drei Schritten, die in den folgenden Abschnitten erläutert werden: Codebuchkonstruktion (engl. "codebook construction"), Codewortzuordnung (engl. "codeword assignment") und der anschließenden Klassifizierung.

Codebuchkonstruktion

Ziel dieses Schrittes ist es, Teilsequenzen (sogenannte "Codewörter") zu bestimmen, die für die 1D-Eingangssensorik charakteristisch sind. Dies wird erreicht, indem Zeitfenster aus dem ursprünglichen Datensatz für jeden Sensorkanal unabhängig voneinander nach dem im Kapitel 9.3 definierten Segmentierungsansatz extrahiert werden. Aus jedem so erhaltenen Zeitfenster der Größe T werden kleinere Segmente der Größe α unterteilt. Ein Clustering-Algorithmus wird dann auf die Menge der Segmente α angewendet, um Clusterzentren zu finden. Nach der Kon-

vergenz werden die Clusterzentren als Codewörter betrachtet und zum Aufbau einer Sammlung von Codewörtern mit dem Namen "Codebuch" verwendet, wie in Abbildung 9 aus [11] dargestellt. Die Anzahl der Codewörter (d.h. die Größe des Codebuchs oder die Anzahl der Cluster) ist ein Hyperparameter des Verfahrens. Im Rahmen dieser Arbeit wurde ein k-means Clustering-Algorithmus verwendet, um die Codewörter auf den ELISE-Daten zu erhalten.

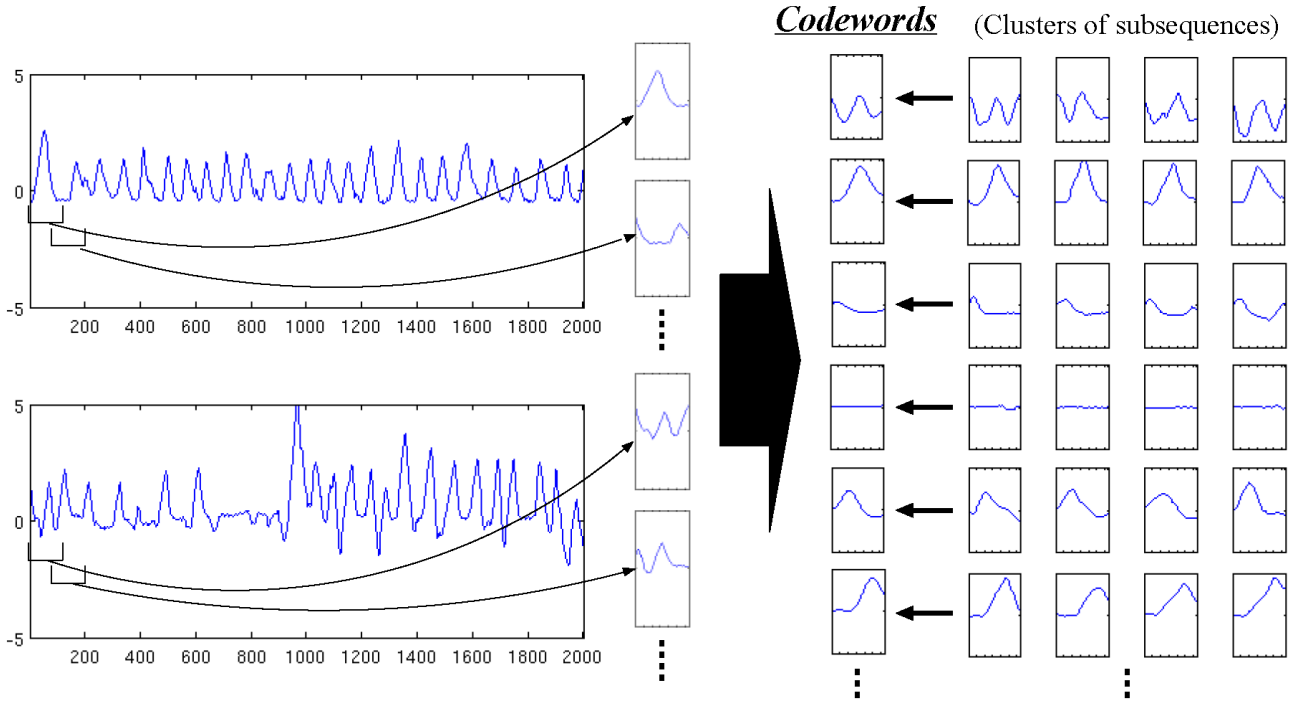


Abbildung 9: Codebuchkonstruktion: Zeitfenster von Daten werden zunächst extrahiert. Ein Clustering-Algorithmus wird dann auf das alle Zeitfenster angewendet, um Clusterzentren (und damit Codewörter) zu finden, die zum Aufbau des Codebuchs verwendet werden.

Codewortzuordnung

Nach der Konstruktion der Codewörter wird für jedes Zeitfenster T ein histogrammbasierter Merkmalsvektor erstellt (vgl. Abbildung 10, entnommen aus [11]). Der zu klassifizierende Datensatz wird zunächst in Zeitfenster der Größe T segmentiert, aus denen Segmente der Größe α nach dem gleichen Verfahren wie beim Aufbau des Codebuchs extrahiert werden. Jedes Segment α wird dann mit den Codewörtern verglichen, so dass das "ähnlichste" Codewort gefunden werden kann. Ein K-Bin-Histogramm (mit K Anzahl der Codewörter) mit Informationen über die Anzahl der Male enthält, die jedes Codewort als am ähnlichsten zu den Segmenten α im Zeitfenster T betrachtet wurde, wird dann erstellt und als Merkmalsvektor verwendet, um das zu klassifizierende Zeitfenster T darzustellen. Das Maß für die Ähnlichkeit von Codewörtern und Datensegmenten α basiert auf der euklidischen Entfernung.

Der zuvor beschriebene Ansatz wird als "Hard-Zuordnung" (engl. "hard assign-

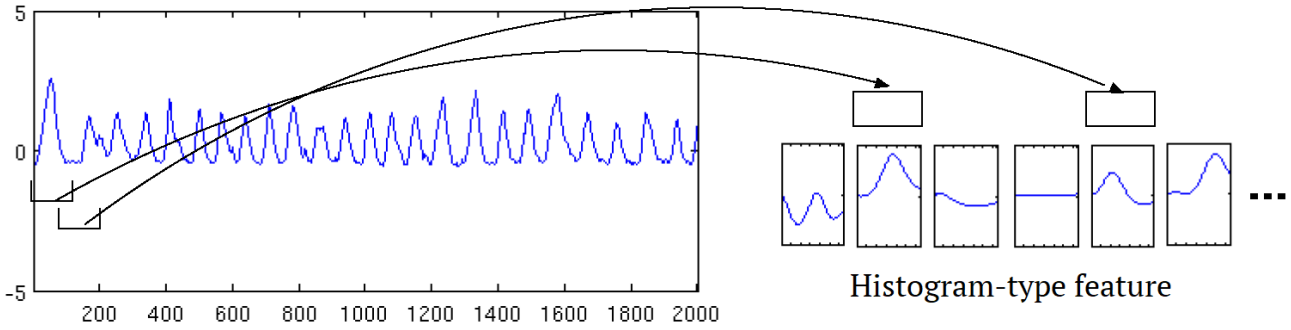


Abbildung 10: Codewortzuweisung: Jedes Datensegment wird mit den Codewörtern verglichen und ein Histogramm mit Informationen darüber, wie oft jedes Codewort als "am ähnlichsten" betrachtet wurde, wird erstellt.

ment") bezeichnet, da Datensegmente einem einzigen Codewort zuzuordnen werden. Ein Nachteil dieser Vorgehensweise ist die mangelnde Flexibilität im Umgang mit potenzieller Unsicherheit bei der Codewortzuweisung, die z.B. auftreten kann, wenn ein Zeitfenster von Daten zwei oder mehr Codewörtern sehr ähnlich ist, da es nur einem zugeordnet werden kann. Ein alternativer Ansatz dieses Problem zu umgehen ist die so genannte "Soft-Zuordnung" (engl. "soft assignment"). Hierbei werden auch alle Codewörter des Codebuchs Datensegmenten zuzuordnen, wobei die Ähnlichkeit jeweils als ein Bin im Histogramms dargestellt wird. Sehr ähnlich entspricht hierbei einem hohen Wert und nicht ähnlich einem kleinen Wert (anstatt 0 oder 1 wie in der Variante der Hard-Zuordnung). Eine Kerneldichtefunktion (vgl. [12]) wird verwedent, um die Histogramm-Bins zu berechnen. Es wurde die folgende Funktion verwendet:

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

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

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

Um numerischen Unterlauf zu vermeiden, wird die Gleichung (7) zunächst mit dem Log-Sum-Exp-Trick (vgl. [13]) berechnet.

Fusion mehrerer Sensoren

Oftmals werden mehrere Sensoren verwendet, die gleichzeitig mehrere verschiedene Signale derselben Emotion erzeugen. Die Fusion dieser Signale ist wichtig, da sie die Genauigkeit der Emotionserkennung verbessern kann. Es gibt zwei verschiedene Ansätze (vgl. [14]): die frühe Fusion (engl. "early fusion") und die späte Fusion (engl. "late fusion"):

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

In dem ELISE Projekt verwenden wir den späten Fusionsansatz, weil er rechnerisch günstiger ist und von K. Shirahama (vgl. [11]) empfohlen wird.

9.4.3 Merkmals Auswahl

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

Der folgende Pseudocode beschreibt unseren Algorithmus zur Merkmalsauswahl:

Algorithm 1: Merkmalsauswahl-Algorithmus.

Input parameters:
- C, γ = C-SVM params
- $candidates = [f1, f2, ..., fn]$ list of n features to test
- $training\_set$ = set of all features computed on the training data
- $testing\_set$ = set of all features computed on the testing data

Output parameters:
- $best\_feature\_combination$ = list of the best features
- $best\_accuracy$ = classification accuracy obtained by the best features
- $feature\_ranking$ = list ranking features in decreasing order of relevance

---

Begin

$candidates\_to\_test = candidates$
$current\_best\_features = \emptyset$
$current\_best\_accuracy$ = -1
$all\_time\_best\_features = \emptyset$
$all\_time\_best\_accuracy$ = -1
$feature\_ranking = \emptyset$
while $candidates\_to\_test \neq \emptyset$ **do**
  for $featuref$ **in** $candidates\_to\_test$ **do**
    $trained\_svm = train\_svm(C, \gamma, training\_set, current\_best\_features \cup \{f\})$
    $accuracy = evaluate\_svm(trained\_svm, testing\_set, current\_best\_features \cup \{f\})$
    if $accuracy > current\_best\_accuracy$ **then**
      $best\_feature\_of\_iteration = f$
      $current\_best\_accuracy = accuracy$
    if $accuracy > all\_time\_best\_accuracy$ **then**
      $all\_time\_best\_features = current\_best\_features \cup \{f\}$
      $all\_time\_best\_accuracy = accuracy$
  $feature\_ranking = feature\_ranking \cup [best\_feature\_of\_iteration]$
  $current\_best\_features = current\_best\_features \cup [best\_feature\_of\_iteration]$
  $candidates\_to\_test = candidates\_to\_test \setminus [best\_feature\_of\_iteration]$
return $all\_time\_best\_features, all\_time\_best\_accuracy, feature\_ranking$

9.5 Klassifikation

Wie bereits in Kapitel 3.6.5 beschrieben ist das Ziel der Klassifizierung ein Klassifizierungsmodell zu trainieren, das in der Lage ist, Objekte in den Daten in die entsprechende Klasse zuzuordnen. Die Klassen entsprechen hierbei den Emotionen, die erkannt werden sollen: Glück, Langeweile, Frustation und andere (d.h. alle Emotionen, die nicht Glück, Langeweile oder Furstation entsprechen).

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

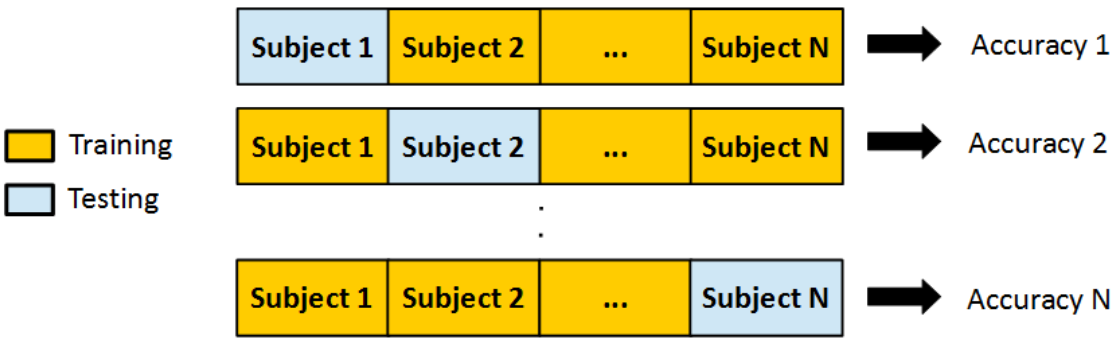


Abbildung 11: Leave-One-Subjekt-Out-Cross-Validation (LOSOCV): N entspricht der Anzahl der Probanden. Für jeden Split wird ein Testset aus den Daten eines Probanden aufgebaut, während die Daten der anderen Probanden einen Trainingsset bilden. Dieser Vorgang wird für die Daten jedes Probanden durchgeführt.

10 Ergebnisse

Verantwortlich: Artur

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

10.1 Ergebnisse der hand-gefertigten Merkmale

Zunächst haben wir die Daten mit einer Normalisierungstechnik vorverarbeitet, die bereits im Kapitel 3.6.1 beschrieben wurde. Entsprechend der ERC haben wir dann den gleitenden Zeitfenster-Segmentierungsansatz (vgl. Kapitel 9.3) mit den Parametern $T = 120$ und $\sigma = 30$ verwendet. Um die optimalen Parameter des SVM-Klassifikators (d.h. Soft-Margin C und Kernelparameter γ) zu bestimmen, wurde ein Gitter-Suchansatz verwendet. Es besteht darin, einen Satz möglicher Werte für jeden Parameter zu definieren und die Leistungen des Klassifikators für jedes Paar möglicher Werte zu testen (d.h. an jedem "Knoten des Gitters").

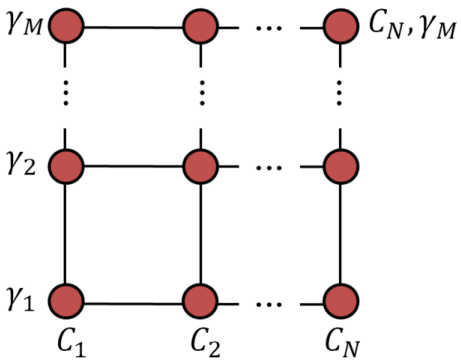


Abbildung 12: Gitter-Suche: Ein definierter Satz möglicher Werte für jeden Parameter wird an jedem "Knoten des Gitters" getestet. Diese Knoten sind in der Abbildung als rote Punkte markiert.

Die folgenden Werte für γ und C wurden getestet:

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

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

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

Tabelle 2: Durchschnittlicher F1-Wert in % für die fünf Falten des Datensatzes.

Die Ergebnisse unserer Experimente zeigen, dass ein solcher Aufbau die Erzielung einer relativ hohen Genauigkeit für die Erkennung der vier Klassen ermöglicht. Die Confusion-Matrizen für den handgefertigten Merkmalsansatz unter Verwendung der SVM-Klassifikation ist in der folgenden Tabelle zu finden.

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

Tabelle 3: Confusion Matrix in % für die fünf Falten des Datensatzes.

10.2 Ergebnisse des Codebook Approach

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

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

10.3 Analyse der Ergebnisse

Es sei darauf hingewiesen, dass Lösungen, die auf der Verwendung von Deep Neural Networks (DNNs) zur Extraktion von Merkmalen basieren (z.B. Multi-Layer-Perceptron, Convolutional Neural oder Long-Short-Term-Memory Networks) ebenfalls getestet wurden. Diese Ansätze konnten aber nicht so gute Ergebnisse erzielen, wie die handgefertigten Merkmale. Erkennungsraten für die am wenigsten vertretenen Klassen (insbesondere "Frustration") scheinen der Grund für die schwache Performance der DNNs zu sein. Wir gehen davon aus, dass dieses Phänomen durch die relativ geringe Größe unseres Datensatzes verursacht wird.

Entgegen unserern Erwartungen liefert der CA schächere Performance als die handgefertigten Merkmale. Der Grund hierfür ist aber sehr wahrscheinlich der selbe wie

bei den DNNs, und zwar der relativ kleine Datensatz. Eine kleine qualitative Analyse der in der Studie verwendeten Daten kann die Gründe für diese Beobachtungen begründen. CA-Merkmale (sowohl für weiche als auch für harte Zuweisungen) sind per Definition sehr empfindlich gegenüber Variationen der Formen der Originalsignale: Die Codewörter werden durch Clustering auf Sätzen von Segmenten bestimmt, die aus den Originalsignalen extrahiert wurden, und die histogrammbasierten Merkmale selbst basieren auf direkten Vergleichen zwischen den Codewörtern und dem Segment der zu klassifizierenden Daten. Daher kann jede Quelle von Rauschen oder Unregelmäßigkeiten in den Originaldaten die Effektivität der CA-Funktionen stark beeinträchtigen. Ein Blick auf die in unserer Studie verwendeten Signale ergab zwei Hauptprobleme: falsche Datenwerte, die durch Hardwareprobleme bei einigen Sensoren verursacht wurden (wie in Abbildung 13), und das Vorhandensein von Rauschen, das Unregelmäßigkeiten in den Signalformen verursacht (siehe Abbildung 14). Mögliche Lösungen zur Behebung dieses Problems könnten sein, zusätzliche Vorverarbeitungstechniken zur Rauschunterdrückung einzusetzen, wie z.B. Tiefpassfilterung.

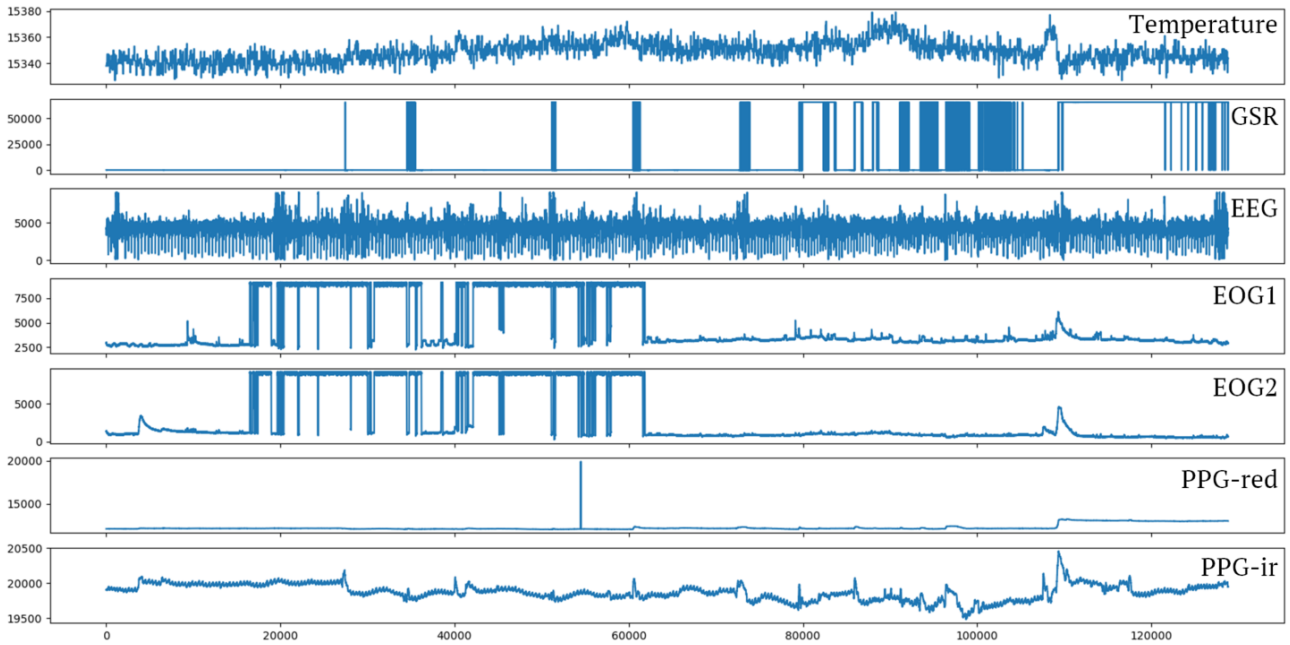


Abbildung 13: Sensoraufzeichnungen von Daten, die im Rahmen des ELISE-Projekts von einem der drei getesteten Probanden erfasst wurden. Die x-Achse repräsentiert die Zeit und die y-Achse die Sensorwerte. Die in den Daten von GSR und den beiden EOG-Kanälen sichtbaren Unregelmäßigkeiten deuten auf Probleme mit der Hardware hin.

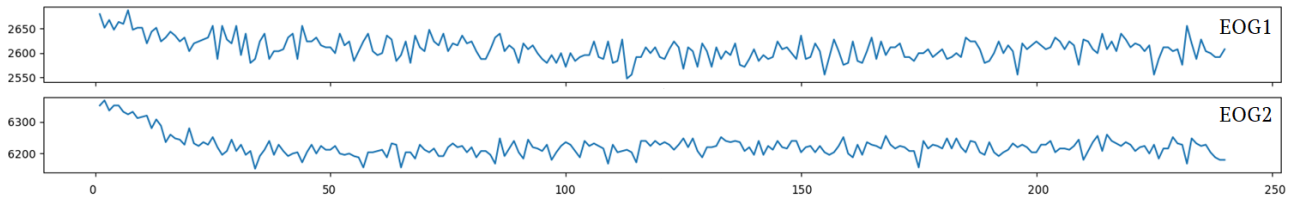


Abbildung 14: Nahaufnahme der im Rahmen des ELISE-Projekts erworbenen EOG-Kanäle von einem der drei getesteten Probanden. Das Vorhandensein von Rauschen in den Daten ist sichtbar, das zu Unregelmäßigkeiten in den Signalformen führt.

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

Part II

Zweiter Prototype

11 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

11.1 Anforderungen

11.2 Konzept

11.3 Hardwareauswahl

11.3.1 Auswahlkriterien

11.3.2 Festlegung der genutzten Hardware

11.4 Hardwarearchitektur

11.4.1 GSR-Sensor

11.4.2 Temperatur-Senosr

11.4.3 Pulsoximeter

11.4.4 EEG

11.4.5 EOG

11.4.6 Datenübertragung

11.5 Programmierung

11.6 Aufnahme der übertragenen Daten

12 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

Part III

Dritter Prototype

13 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

13.1 Anforderungen

13.2 Konzept

13.3 Hardwareauswahl

13.3.1 Auswahlkriterien

13.3.2 Festlegung der genutzten Hardware

13.4 Hardwarearchitektur

13.4.1 GSR-Sensor

13.4.2 Temperatur-Senosr

13.4.3 Pulsoximeter

13.4.4 EEG

13.4.5 EOG

13.4.6 Datenübertragung

13.5 Programmierung

13.6 Aufnahme der übertragenen Daten

14 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

15 Emotionsinduktion

Verantwortlich: Minas

15.1 Ablauf

Verantwortlich: Minas

15.2 Fragebogen

Verantwortlich: Boris

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

(Bild für Fragebogen)

Der zweite Typ ähnelt dem berühmten Modell von James Russels "circumplex" [15]. Es ist ein klassisches Modell mit einer kreisförmigen Struktur, die auf zwei senkrechten Diagonalen ruht. Die vertikale Achse, die die Erregung darstellt, und die horizontale Achse, die die Valenz darstellt. Das Zentrum des Kreises stellt eine neutrale Valenz und ein mittleres Erregungsniveau dar. Andere Emotionen werden auf jeder Ebene des Kreises dargestellt. Hier wird ein weniger bekanntes Modell

verwendet, das "Self Assessment Manikin" (SAM). Es besteht aus drei Reihen mit je fünf Piktogrammen. Diese Piktogramme stellen den Zustand eines Gesichts nach verschiedenen Arten von Emotionen dar. So repräsentiert der erste Bereich die Wertigkeit, der zweite die Erregung und der dritte die Dominanz. Eine Erklärung zu jedem dieser Begriffe ist ebenfalls neben dem Fragebogen enthalten, um die Testpersonen über diese Wörter aufzuklären. Bei jeder Avatar und in der Mitte jeder der beiden Avatar befindet sich ein Checkbox. So muss man für jede Zeile das Checkbox auswählen, das ihrem emotionalen Zustand am besten entspricht. Man kann nur ein Checkbox pro Zeile markieren und man hat auch die Möglichkeit wie bei dem ersten Model seine Wahl zu ändern. Es kann einfach mit Branch-Bedingungen realisieren werden. Diese werden auch in eine Widget Blueprint wie für das erste Modell gemacht. Es gibt auch wieder die zwischen Speicherungsfunktion und das Button "next". Was neues hier kommt ist das Button "back" um wieder zum ersten Fragebogen zu navigieren.
(Bild für circumplex)

15.3 Szenarien

Verantwortlich: Meryem

15.3.1 Glück

Verantwortlich: Minas

15.3.2 Langeweile

Verantwortlich: Boris

In diesem Prototyp wird nur das "Pec-Turning" Spiel als Szenario benutzt, da man die Zeit des Experiments reduzieren wollte. Allerdings gib es ein kleines Unterschied mit dem Spiel von dem zweiten Prototyp. Hier wird die grüne kreisförmige Scheibe durch den Proband selbst in Bewegung gebracht. Nach jeder Bewegung des Kreis muss man mindesten fünften Sekunde warten bis die nächste Bewegung möglich ist. Um dies Szenario in Unreal Engine zu verwirklichen habe ich erstmal ein Widget Blueprint erstellt. Dann habe ich vier mal das Bild von dem Spiel genau an der gleiche Stelle und in der gleiche Große in der Widget hinzugefügt. Das erste Bild ist in der Ausgangsposition. Dann wird jedes der nächsten Bilder eine Drehung des vorherigen Bildes um neunzig Grad sein. Es wird von Anfang an das Ausgangsbild auf sichtbar gesetzt und die anderen auf unsichtbar. Danach habe ich einen transparenten Button immer auf die Bilder der Scheiben eingefügt, um das Wechsel von Bilder zu steuern. Zur Programmierung erstelle ich eine Funktion mit Timeout

und Branch-Bedingung und der Algorithmus wird so rekursiv definiert: Wenn das Button gedrückt wird, wird das nächste Bild auf sichtbar ("visible") gesetzt und die anderen Bilder auf unsichtbar ("unvisible"), dann wird ein Timeout gesetzt und eine Branch-Bedingung soll Prüfen ob das Timeout durch ist. Sollte man vom Ende des Timeouts der Button drücken, so sollte nichts passieren. Wenn das Timeout fertig ist, soll das Szenario sich wiederholen. Es wird auch jede zeit durch einen Text in einem Textfeld gezeigt, wenn man die Scheibe drehen kann und wenn man warten muss. Ein letztes Timeout wird hinzugefügt, um zu prüfen, dass das Szenario ein gewünschtes Zeit dauert.

(Bild für Langweile in VR mit Textbeschreibung)

15.3.3 Frustration

Verantwortlich: Meryem

16 Messreihe

Verantwortlich: Kevin, Artur

17 Mustererkennung

Verantwortlich: Artur

18 Ergebnisse

Verantwortlich: Artur

19 Alternative Lösungen

Verantwortlich: Boris, Arnaud, Minas, Meryem

19.1 Kalibrierung

Verantwortlich: Jonas

19.2 Plan B

Verantwortlich: Boris, Arnaud, Minas, Meryem

20 Zusammenfassung und Ausblick

Machen wir später.

20.1 Zusammenfassung

20.2 Fazit

20.3 Ausblick

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

Bitte alle verwendete Abkürzungen nochmals hier aufführen.

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

Referenzen

[1] X. Zhu. Semi-supervised learning literature survey, July 2008.

[2] C. Cortes and V. Vapnik. *Support-Vector Networks*, volume volume 20. Kluwer Academic Publishers, Boston, September 1995.

[3] E. Kim. Everything you wanted to know about the kernel trick. September 2013.

[4] R. H. Gault. A history of the questionnaire method of research in psychology. `https://www.tandfonline.com/doi/abs/10.1080/08919402.1907.10532551`, accessed on 30 January 2019.

[5] S. J. Vodanovich. Psychometric measures of boredom: A review of the literature. *Journal of Psychology*, volume 137(issue 6):pages 569–595, November 2003.

[6] J. Grus. *Data Science from Scratch*. O'Reilly Media, April 2015.

[7] H.P. Martinez, Y. Bengio, and G.N. Yannakakis. Learning deep physiological models of affect. *IEEE Computational Intelligence Magazine*, volume 8(issue 2):pages 20–33, April 2013.

[8] A. Piet. *Bachelor-thesis: Emotion recognition using 1D physiological signals following a supervised learning approach*. December 2017.

[9] J. Littau. *Bachelor-thesis: Analysis of fourier features for emotion recognition using physiological signals*. June 2018.

[10] P. Gouverneur. *Bachelor-thesis: Classification of physiological data for emotion recognition*. September 2016.

[11] K. Shirahama, L. Koeping, and M. Grzegorzek. Codebook approach for sensor-based human activity recognition. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 197–200, September 2016.

[12] J.C. van Gemert, C.J. Veenman, and A.W.M. Smeulders. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32(issue 7):pages 1271–1283, June 2009.

[13] K.P. Murphy. *Machine Learning and A Probabilistic Perspective*. The MIT Press, London, September 2012.

[14] C.G.M. Snoek, M. Worring, and A.W.M. Smeulders. Early versus late fusion in semantic video analysis. *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402, November 2005.

[15] J. Russel. A circumplex model of affect. *Journal of Personality and Social Psychology*, volume 39(issue 6):pages 1161–1178, 1980.

Anhang