FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Inhaltsbasierte Suchmaschine für Videos von Parlamentssitzungen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Karin Straka, Bakk.techn.
Matrikelnummer 0225277

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Ao.Univ.Prof. Mag. Dr. Horst Eidenberger

Wien, 15. Februar 2018

Karin Straka	Horst Eidenberger

Erklärung zur Verfassung der Arbeit

Karin Straka, Bakk.techn.
Wilhelminenstraße 230/2/1, 1160 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 15. Februar 2018

Karin Straka

Danksagung

„Der Langsamste, der sein Ziel nicht aus den Augen verliert, geht noch immer geschwinder als jener, der ohne Ziel umherirrt.“ - Gotthold Ephraim Lessing

Ich widme diese Arbeit allen, die die wichtigen Ziele in ihrem Leben hartnäckig verfolgen ohne aufzugeben, auch wenn Schwierigkeiten auftauchen.

Ich bedanke mich bei allen, die daran geglaubt haben, dass ich dieses Ziel erreichen werde und mich auf dem Weg dorthin mit ihrer Freundschaft und ihrem Verständnis unterstützt haben. Ganz besonders möchte ich mich bei meinen Eltern bedanken, die immer alle meine Entscheidungen mittragen, helfend an meiner Seite sind und mein Studium ermöglicht haben. Außerdem möchte ich mich bei meinem Lebensgefährten Christian bedanken: für sein Verständnis, sein Durchhaltevermögen und die vielen Dinge, die er mir im Alltag abgenommen hat. Und natürlich möchte ich Herrn Prof. Eidenberger für seine hervorragende Betreuung während meiner Diplomarbeit meinen Dank aussprechen.

Kurzfassung

Große unstrukturierte Videodatensammlungen automatisiert und damit effizient durchsuchbar zu machen, ist eine Aufgabe an die Wissenschaft, deren Lösung für viele Bereiche z.B. Filmarchive, Online-Mediatheken, Videoüberwachung, Video-Lernplattformen große Bedeutung hat. Ein gängiger Ansatz hierfür sind manuell generierte Inhaltsindizes, die durch Speicherung von Videoframes, versehen mit Schlagwörtern oder textuellen Beschreibungen, produziert werden. Diese sind allerdings extrem zeitaufwändig in der Erstellung, außerdem ungenau und unvollständig. Große Mengen an enthaltener Information gehen dadurch für die Suche und somit für die Wiedergabe verloren.

In dieser Diplomarbeit werden deshalb die Möglichkeiten des aktuelleren Konzeptes der „inhaltsbasierten“ Datensuche am Beispiel des Einsatzes für die Segmentierung und Klassifikation von Videomitschnitten österreichischer Nationalratssitzungen erforscht. Ziel ist es, ohne manuelle Indexierung automatisiert und auf multimodaler Basis Audio- und Bildmerkmale zu extrahieren und damit entsprechende Klassifikatoren zu trainieren, sodass diese für die Klassifizierung von Audioereignissen und Personen eingesetzt werden können. Der Fokus der Klassifizierung liegt dabei auf der Erfassung von Szenen, bei denen die Stimmung im Sitzungssaal von der klassischen Grundstimmung einer Rede abweicht, und damit einen Hinweis auf relevante Ereignisse in den Sitzungen liefert. Die Erkennung der handelnden Personen inklusive ihrer Gesichtsemotion ist der zweite große Schwerpunkt dieser Arbeit.

Gestartet wird mit einem Überblick über die Grundlagen der inhaltsbasierten Videoverarbeitung mit den Teilbereichen Videosegmentierung, Merkmalsextraktion aus Bild- und Audiodaten und Klassifizierung. Außerdem werden die Methoden zur statistischen Evaluierung der Ergebnisse vorgestellt, gefolgt von einer Übersicht verwandter Forschungsarbeiten. Danach folgt die Erklärung des anhand der gewählten Merkmale und Klassifizierungsmethoden implementierten Prototyps. Den Abschluss bildet die statistische Auswertung der Klassifizierungsergebnisse, die zeigt, dass der „inhaltsbasierte“ Ansatz für die Merkmalsextraktion und Klassifizierung durchaus geeignet für eine Detektion von relevanten Ereignissen und Personen in Parlamentsvideos ist und eine aufwändige manuelle Indexierung im Vorfeld nicht benötigt. Es wird dargestellt, dass die Audiomerkmale im vorliegenden Fall aussagekräftiger sind als die Bildmerkmale. Die Fokussierung auf die Erkennung von Audioereignissen zur Detektion von relevanten Szenen hat sich aus diesem Grund als richtig erwiesen. Speziell die Klassifizierung der Gesichtsemotion hat sich als problematisch herausgestellt, da die Gesichtsmimik der Abgeordneten in vielen Fällen für eine korrekte Auswertung nicht ausgeprägt genug ist.

Abstract

Making big, unstructured video data collections searchable fully automated and efficiently is a scientific task whose solution would be of big interest. Many data collections like film archives, online media centres, video surveillance archives and online learning platforms depend on an efficient search structure. It is common to use manually generated content indices for this purpose. These indices are produced by saving video frames including metadata like keywords or textual annotations. This task is extremely time-consuming. The produced indices are mostly inexact and incomplete. Huge amounts of information are lost for search and retrieval by this approach.

Therefore the possibilities of the more current concept of „content based“ data search are investigated with this master's thesis, as an example of using this approach for segmentation and classification of videos from Austrian parliament sessions. The aim is the automated and multimodal extraction of audio and image features for training appropriate classifiers in order to use them for classification of audio events and persons. The main focus of the classification lies in the detection of scenes where the atmosphere in the parliament chamber is different from the classical speech-atmosphere, which would be an evidence of interesting events during the sessions. The recognition of acting parliamentarians - including their facial expression - is the second big focus of this work.

This paper starts with an overview of the basic principles of “content based” video retrieval including its subsections: video segmentation, feature extraction from image and audio data and classification. Furthermore, methods for the statistical evaluation of the results will be presented, followed by an overview of related research papers. Afterwards, an explanation of the implemented prototype on the basis of the chosen features and classification methods is given. Finally, the statistical evaluation of the classification results is introduced, which show that the „content based“ approach for feature extraction and classification is definitely appropriate for the detection of relevant events and persons in videos of parliament sessions without the need for complex, manual indexing in advance. It is shown that, in the case of parliament session videos, audio features are more significant than visual features. Focussing on the detection of audio events for the identification of relevant scenes has proved to be right for this reason. Especially the classification of facial expression has turned out to be problematic, because in many cases the expression is not distinctive enough for a correct evaluation.

Abkürzungsverzeichnis

AAC	Advanced Audio Coding
acc	Accuracy
AU	Action Unit
BoVW	Bag of Visual Words
BIC	Bayessches Informationskriterium
DCT	Diskrete Kosinustransformation
DFT	Diskrete Fourier-Transformation
DTW	Dynamic Time Warping
ECOC	Error-correcting output codes
ECR	Edge Change Ratio
EE	Energy-Entropy
EM	Expectation Maximation
EmFACS	Emotion FACS System
f1	F1-Score
FACS	Facial Action Coding System
FFT	Schnelle Fourier-Transformation
FN	False negative
FP	False positive
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HTK	Hidden Markov Model Toolkit
kNN	k-Nearest Neighbour
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LFCC	Linear Frequency Cepstral Coefficient
MFCC	Mel Frequency Cepstral Coefficient
NLP	Natural Language Processing
PCA	Principal Component Analysis
p	Precision
r	Recall
RBF	Radiale Basisfunktion
RMS	Root-Mean-Square

SIFT	Scale-invariant Feature Transform
STE	Short-Time-Energy
SURF	Speeded Up Robust Features
SVM	Support Vector Machine
TN	True negative
TP	True positive
ZCR	Zero-Crossing-Rate

Inhaltsverzeichnis

Kurzfassung **vii**

Abstract **ix**

Abkürzungsverzeichnis **xi**

Inhaltsverzeichnis **xiii**

1 Einführung **1**
1.1 Motivation 1
1.2 Zielsetzung und Vorgehensweise 2
1.3 Struktur der Arbeit 4

2 Aktueller Stand der Technik **5**
2.1 Grundlagen einer inhaltsbasierten Videosuche 5
2.2 Videosegmentierung 6
2.2.1 Shot-Erkennung 6
2.2.2 Extraktion von Schlüsselbildern 8
2.3 Bildanalyse 10
2.3.1 Globale Bildmerkmale 10
2.3.2 Lokale Bildmerkmale 11
2.3.3 Gesichtserkennung und -verifizierung 15
2.3.4 Emotionserkennung 17
2.4 Audioanalyse 19
2.4.1 Zeitabhängige Merkmale 21
2.4.2 Frequenzabhängige Merkmale 22
2.4.3 Spracherkennung 26
2.5 Klassifizierung 27
2.5.1 Überblick Klassifikatoren 28
2.5.2 Bag of Visual Words 28
2.5.3 Support Vector Machine 30
2.5.4 Gaussian Mixture Model und Hidden Markov Model 32
2.5.5 Dynamic Time Warping 33
2.6 Methoden zur statistischen Evaluierung 34

2.6.1 Statistische Kennzahlen . . . 34
2.6.2 Kreuzvalidierung . . . 36
2.7 Verwandte Arbeiten . . . 37

3 Implementierung **39**
3.1 Methodische Vorgehensweise . . . 39
3.1.1 Systemspezifikation . . . 39
3.1.2 Aufbau des Prototyps . . . 40
3.1.3 Ground-Truth-Daten für Training und Evaluierung . . . 41
3.1.4 Zeitliche Videosegmentierung und Schlüsselbild-Auswahl . . . 44
3.1.5 Visuelle Merkmale, Cluster und Training . . . 46
3.1.6 Auditive Merkmale, Cluster und Training . . . 47
3.1.7 Suchphase . . . 51
3.2 Entwicklungsumgebung . . . 56
3.3 Graphische Benutzeroberfläche . . . 57

4 Statistische Evaluierung **59**
4.1 Shoterkennung . . . 59
4.2 Gesichtserkennung und -verifizierung . . . 60
4.3 Emotionserkennung . . . 63
4.4 Erkennung von Audioereignissen . . . 65

5 Zusammenfassung und Ausblick **71**

Abbildungsverzeichnis **73**

Tabellenverzeichnis **75**

Literaturverzeichnis **79**

KAPITEL 1

Einführung

Dieses Kapitel gibt einen Überblick über die Motivation zur Auswahl des bearbeiteten Themengebietes, die Aufgabenstellung und den gewählten Lösungsansatz.

1.1 Motivation

In der heutigen Zeit mit ihrer Flut an Multimedia-Anwendungen, die aus dem Alltag nicht mehr wegzudenken sind, steigen die anfallenden Datenmengen bei gleichzeitig sinkenden Speicherpreisen, rasant an. Große, unstrukturierte Videodatensammlungen automatisiert, und damit effizient durchsuchbar zu machen, ist eine Anforderung, deren Lösung in vielen Bereichen, wie zum Beispiel für digitale Filmarchive, Online-Mediatheken, Videoüberwachung, Video-Lernplattformen, große Bedeutung hat.

Die Anwendungen, die für die Beschreibung und das Management von Videodaten bisher zur Verfügung stehen, sind noch immer sehr limitiert. Ein großes Problem stellt die semantische Lücke zwischen den aus Videos extrahierten low-level Merkmalen (z.B. Farbe, Helligkeitsverteilung) und der Anforderung der Anwenderinnen und Anwender dar, bei der Suche mit diesen Merkmalen auf dem höheren Level der tatsächlichen inhaltlichen Bedeutung zu arbeiten [1]. Der zentrale Kern der Aufgabenlösung ist deshalb, eine sinnvolle Indexierung der Daten zu erreichen, um ein strukturiertes Medium zu erhalten, das optimal nach bestimmten Kriterien durchsucht werden kann und diese semantische Lücke schließt [2]. Gängigster Ansatz dafür sind manuell generierte Inhaltsindizes, die durch Speicherung von Videoframes, versehen mit Schlagwörtern oder textuellen Beschreibungen, produziert werden. Diese sind allerdings extrem zeitaufwändig in der Erstellung, außerdem ungenau und unvollständig [2]. Große Mengen an enthaltener Information und Wissen gehen dadurch für die Suche, und somit für die Wiedergabe verloren. Aufgrund dieser Problematik wurde innerhalb des letzten Jahrzehnts intensive Forschung im Bereich der „inhaltsbasierten“ Bildersuche betrieben. „Inhaltsbasiert“ bedeutet, dass der Inhalt

von Bildern aufgrund ihrer semantischen Bedeutung und unter Verwendung von automatisiert extrahierten Eigenschaften wie Farbe, Textur, Form durchsucht wird, und nicht aufgrund von zuvor vergebenen Metadaten, wie Schlagwörter und Textbeschreibungen [3]. Der generellen Verwendung dieser Technik steht entgegen, dass sie nach wie vor einen relativ geringen Genauigkeitswert aufweist [4]. Trotz dieses Umstandes wurde in [4]. festgestellt, dass die Ergebnisse der Suchfunktion durch den Einsatz der „inhaltsbasierten" Videosuche auf jeden Fall verbessert werden können. Für die Videosuchfunktion eignet sich ein multimodaler Ansatz, da Videos neben den Bildmerkmalen auch Audiodaten und Text beinhalten, die zur Analyse des Inhaltes verwendet werden können. Multimodalität bedeutet in diesem Zusammenhang, dass zumindest zwei Kanäle genutzt werden, um eine semantische Suche auf einen bestimmten Inhalt abzubilden. Dabei muss beachtet werden, dass die Anzahl der low-level Merkmale, die aus verschiedenen Kanälen extrahiert werden können, begrenzt ist [1]. Diese Vorgehensweise eröffnet die Möglichkeit, eine wesentlich breitere Palette an Merkmalen zu verwenden. Die Kunst besteht darin, die richtige Kombination für die geplante Klassifizierung zu finden.

Ein Umstand, der auf jeden Fall für die automatisierte Indexierung von Nationalratssitzungsmitschnitten spricht, liegt in den Videoeigenschaften wie Länge und Inhalt begründet. Die Mitschnitte können bis zu 10 Stunden dauern. Dabei werden die unterschiedlichsten Themen von mehreren Abgeordneten besprochen. Die einzelnen Beiträge sind im Normalfall nicht länger als 5 bis 10 Minuten. Die wenigsten Benutzerinnen und Benutzer werden sich für das gesamte Material interessieren, sondern nach bestimmten Abgeordneten oder Vorkommnissen suchen. Eine manuelle Vergabe von Tags für diesen Zweck wäre ein zu großer Aufwand, wie im Vorfeld bereits angesprochen.

1.2 Zielsetzung und Vorgehensweise

Das Ziel dieser Diplomarbeit ist es, anhand eines Prototyps, der den Benutzerinnen und Benutzern die Funktionalität bietet, Videos von österreichischen Nationalratssitzungen nach bestimmten Szenen und Personen zu durchsuchen, darzustellen, dass der inhaltsbasierte Ansatz für die Klassifizierung in diesem Bereich zum gewünschten Ergebnis führt. Die im Zentrum stehende Klassifizierungsphase beinhaltet die Zuordnung von Videosequenzen zu zuvor definierten Kategorien, den Klassen [5]. Eine Klasse besteht aus Objekten, die dieselben Eigenschaften besitzen [6]. Im Moment gibt es von Seiten des österreichischen Parlaments [7], im Gegensatz zu anderen europäischen Parlamenten wie z.B. dem deutschen Bundestag [8], für die Bürgerinnen und Bürger kein derartiges Angebot. Nationalratssitzungen können im Normalfall für die kurze Zeitspanne von sieben Tagen in der ORF-TVthek und später nur in geringem Umfang auf Youtube nachgesehen, aber nicht durchsucht werden. Durch die Entwicklung des Prototyps sollen auch Antworten auf folgende zentrale Fragen gefunden werden:

- Welche Klassen sind für die geplante Aufgabe sinnvoll?
- Welche Merkmalskombinationen sind zur Unterscheidung der Klassen geeignet?
- Welche Klassifizierungsmethoden können verwendet werden?

Die theoretischen Grundlagen und Konzepte einer inhaltsbasierten Suchmaschine werden in dieser Arbeit ebenfalls thematisiert.

Eine der Kernfunktionen zielt darauf ab, eine von der Grundstimmung der Rede abweichende Stimmung im Sitzungssaal zu erkennen und damit für betrachtende Personen wichtige Schlüsselszenen zu klassifizieren. Dabei handelt es sich im Speziellen um Szenen die einen höheren Lärmpegel aufweisen, in denen geklatscht oder gelacht wird oder es Zwischen- bzw. Ordnungsrufe gibt. Die Grundlage für die Klassifizierung ist eine automatisierte Indexierung basierend auf einem multimodalen Ansatz, der die Analyse von Bild- und Audiomerkmalen beinhaltet. Aufgrund der im Zentrum stehenden Erkennung akustischer Besonderheiten liegt das Hauptaugenmerk auf der Auswahl der dafür nötigen Audiomerkmale. Diese müssen so gewählt werden, dass sie große Variabilität zwischen den Klassen und damit gute Unterscheidbarkeit gewährleisten. Eingesetzt werden Audiomerkmale aus dem Zeit-, Spektral- und Cepstralbereich, für deren Berechnung das Audiosignal in kleine, überlappende Zeitfenster zerlegt wird. Besondere Bedeutung haben dabei die „Mel-Frequency-Cepstral-Coefficients“ (MFCC) Merkmale, die für die Klassifikation aller Klassen relevant sind [9], [10], [11], [12], [13].

Die Bildanalyse für den geplanten Prototyp splittet sich in drei Teilbereiche auf. Als erste Aktion bildet die Lokalisierung von Gesichtern im Bild die Basis für alle weiteren Analyseschritte. Darauf folgt die Erkennung, welcher Person das Gesicht zuzuordnen ist einerseits und die Mimik Erkennung im lokalisierten Gesicht andererseits. Aus der Mimik Erkennung können Schlüsse auf die Emotion der abgebildeten Person gezogen werden. Bei der Wahl der geeigneten Bildmerkmale muss bedacht werden, dass die gesuchten Gesichter verschiedene Größen haben, nicht immer an derselben Stelle im Bild positioniert sind, dass es sich um Frontal- oder Profilbilder handeln kann und, dass es teilweise auch zu Verdeckungen kommt. Bildmerkmale werden in die drei Klassen „model-“, „shape-“ und „appearance-basiert“ eingeteilt, wobei die dritte Gruppe in der Bildklassifizierung bevorzugt verwendet wird [14]. Appearance-basierte Modelle können entweder lokale oder globale Merkmale verwenden. Aufgrund der Merkmalsanforderungen eignen sich für die implementierte Bildanalyse lokale Merkmale, zu denen etwa HOG, SIFT und SURF gehören, am besten [15], [16], [17], [18], [19].

Für die Klassifizierung werden Methoden aus dem Bereich des „Maschinellen Lernens“ verwendet. Grundsätzlich unterscheidet man auf diesem Gebiet drei Gruppen: „überwachte“, „halbüberwachte“ und „unüberwachte“ Verfahren. Die Klassifizierung gehört zu den „überwachten“ Verfahren [20]. Überwachtes Lernen bedeutet, dass die Klassenzugehörigkeit der Instanzen des Trainingssets bekannt ist. Ziel ist es, mit Hilfe der extrahierten Merkmale des Trainingssets ein Modell zu erstellen, mit dessen Hilfe es möglich ist, die Klassenzugehörigkeit von unbekannten Objekten zu berechnen. Bekannte Verfahren sind etwa k-Nearest Neigbour (kNN), Entscheidungsbäume, Bayessche Netzwerke, Support Vektor Maschinen (SVM), Gaussian Mixture Models (GMM) [21], [5], [22], [23], [14], [24].

Die Kombination von Merkmalen aus zwei Bereichen als Grundlage der Klassifizierung bildet auch die Basis der Suchfunktion, die ebenfalls aus mehreren Komponenten besteht. Zum einen bietet sie die Möglichkeit, nach Szenen mit bestimmten Abgeordneten durch

Anklicken des gewünschten Gesichts in einer Bildergalerie oder mittels Namenseingabe zu suchen, zum anderen die Möglichkeit, nach akustischen Besonderheiten zu suchen wie z.B. nach Ordnungsrufen, Applaus oder Lachen.

Im nächsten Kapitel wird auf die Vielfalt an zur Verfügung stehenden Merkmalen und Klassifikationsmethoden und die Komplexität der Auswahl einer für die angedachte Aufgabe passenden Kombination noch genauer eingegangen.

1.3 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen der inhaltsbasierten Videoverarbeitung mit den Teilbereichen Videosegmentierung, Bild-, Audioanalyse und Klassifikation genauer dargestellt. Es beinhaltet eine detaillierte Beschreibung der Merkmalsauswahl und -extraktion aus Bild- und Audiodaten und einen Überblick über mögliche Klassifikationsmodelle. Außerdem werden einige Methoden zur statistischen Evaluierung der Ergebnisse vorgestellt, gefolgt von einer Übersicht verwandter Forschungsarbeiten. Kapitel 3 widmet sich der Beschreibung der für den implementierten Prototypen verwendeten Methoden. Die Ground-Truth-Daten für Training und Evaluierung werden hier ebenfalls näher erklärt. Die graphische Benutzeroberfläche und die Suchmöglichkeiten für die Anwenderinnen und Anwender sind auch Teil dieses Kapitels. Außerdem wird kurz auf die Wahl der Entwicklungsumgebung eingegangen. Darauf folgt in Kapitel 4 die statistische Auswertung der Klassifikationsergebnisse des vorgestellten Ansatzes. Kapitel 5 fasst die aus der vorliegenden Arbeit gewonnenen Erkenntnisse nochmals zusammen und gibt einen Ausblick auf mögliche, künftige Erweiterungen.

KAPITEL 2

Aktueller Stand der Technik

In diesem Kapitel werden die theoretischen Grundlagen der inhaltsbasierten, multimodalen Videoverarbeitung anhand der anzuwendenden Teilschritte Segmentierung, Merkmalsextraktion und Klassifizierung erläutert. Weiters wird ein Überblick über die Methoden und Kennzahlen der statistischen Ergebnis-Evaluierung gegeben. Der Fokus liegt auf Verfahren, die für die gewählte Aufgabenstellung von Bedeutung sind. Im letzten Abschnitt werden verwandte Themenkomplexe mittels aktueller Forschungsberichte dargestellt.

2.1 Grundlagen einer inhaltsbasierten Videosuche

Hu et al. [5] geben einen ausführlichen Überblick über die einzelnen Teilschritte, die für eine inhaltsbasierte Videoindexierung und –wiedergabe nötig sind.

Der erste Schritt besteht aus der Zerlegung des Videos in kleinere Einheiten, um eine Reduktion der zu analysierenden Datenmenge zu erreichen. Für die Bildverarbeitung müssen dazu Shotgrenzen erkannt und Schlüsselbilder extrahiert werden. Die Audioverarbeitung hat als Grundlage die Zerlegung des Audiokanals in überlappende Frames, die von ihrer Länge in die drei Gruppen „short-time“, „mid-time“ und „long-time“ eingeteilt werden.

Die Datenanalyse im nächsten Schritt basiert auf der Extraktion von Merkmalen. Anfangs wurden hauptsächlich visuelle Merkmale verwendet. Mittlerweile sind Audiomerkmale in der Analyse von Videoinhalten ebenfalls sehr bedeutend geworden [25]. Bildmerkmale werden zum Beispiel für die Gesichtserkennung und -verifizierung oder im Bereich der Emotionserkennung verwendet, Audiomerkmale für die Audioeventerkennung oder die Sprachanalyse. Für diesen multimodalen Ansatz kommen häufig neben den Bild- und Audiomerkmalen auch Bewegungs- und Textmerkmale zum Einsatz.

Die extrahierten Daten werden unter Verwendung entsprechender Klassifikatoren für die Klassifizierung von Objekten und Ereignissen eingesetzt. Die Klassifizierung liefert

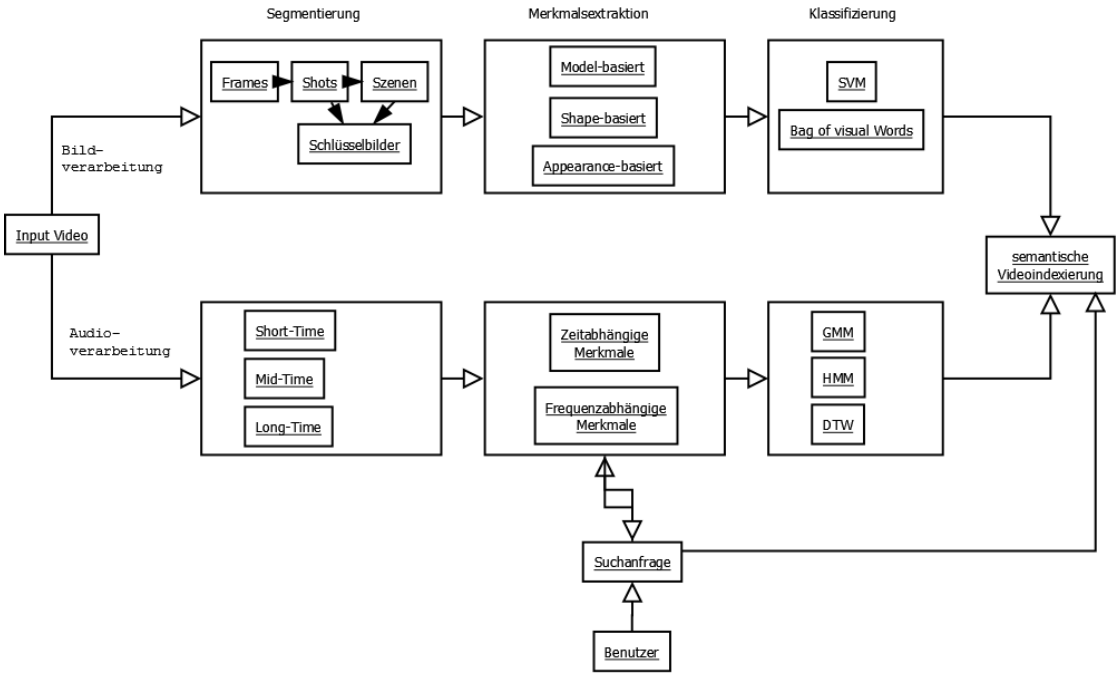

Abbildung 2.1: Schritte einer inhaltsbasierten, multimodalen Videoindexierung und Suche

als Ergebnis eine semantische Videoindexierung, die die Grundlage für die inhaltsbasierte Videosuche bildet. Abbildung 2.1 zeigt eine Übersicht der beschriebenen Schritte.

2.2 Videosegmentierung

Aufgrund der großen Datenmenge müssen Videos für die Analyse und die Merkmalsextraktion in kleinere Einheiten zerlegt werden. Datenreduktion durch Aussortierung von redundant vorliegender Information ist dabei ein wichtiger Punkt. Die kleinste Einheit eines Videos ist ein Frame = ein Einzelbild. In Europa wird mit einem TV-Standard von 25 Frames/Sekunde gearbeitet. Eine aufeinanderfolgende Sequenz von Frames einer Kameraposition bildet einen Shot. Die Frames innerhalb eines Shots weisen im Normalfall große Ähnlichkeiten auf. Deshalb ist es für die visuelle Merkmalsextraktion ausreichend, aus jedem Shot Schlüsselbilder für die weitere Verarbeitung auszuwählen und den Rest der Frames zu verwerfen. Mehrere Shots mit ähnlichem Inhalt können zu Szenen zusammengefasst werden. Diese haben meist eine semantisch höhere Aussagekraft als einzelne Shots.

2.2.1 Shot-Erkennung

Bei der Shot-Erkennung ist es nötig, die Art des Übergangs zwischen den einzelnen Shots zu berücksichtigen. Moderne Schnittprogramme stellen eine Vielzahl von möglichen

Übergängen zur Verfügung. Allerdings fallen nach [26] 99% in eine der drei folgenden Kategorien:

- **Harter Schnitt:**
  Bei dieser Art von Schnitt gibt es keine Übergangsframes zwischen zwei Shots, sondern der erste Frame des neuen Shots folgt auf den letzten Frame des vorangegangenen Shots. Es wird also zu jedem Zeitpunkt nur ein Bild mit einem Alphawert von 100% dargestellt.

- **Aus- bzw. Einblendung:**
  Hier erfolgt der Übergang zwischen dem Ende eines Shots und einem fixen Bild (Ausblendung) bzw. von einem fixen Bild zum Shot-Anfang (Einblendung) schrittweise unter Einbeziehung von mehreren Frames, indem der Alphawert graduell verringert bzw. erhöht wird.

- **Überblendung:**
  Im Fall der Überblendung werden die Frames des aktuellen Shots schrittweise ausgeblendet und gleichzeitig die Frames des neuen Shots eingeblendet. Es handelt sich also um eine Kombination aus Aus- und Einblendung. Die Frames der beteiligten Shots überlappen einander.

Die existierenden Shoterkennungsmethoden lassen sich in zwei große Gruppen unterteilen: schwellwertbasierte und auf statistischem Lernen basierende Verfahren. Bei den ersteren gibt es drei mögliche Arten von Schwellwerten: globale, adaptive und eine Kombination aus globalen und adaptiven. Adaptive Schwellwerte, die unter Verwendung eines veränderlichen Intervalls lokal festgelegt werden, sind komplizierter in der Berechnung, führen aber oftmals zu besseren Ergebnissen als globale [5]. Die auf statistischem Lernen basierenden Verfahren gehören zum Bereich der Klassifikation und können mit Hilfe von überwachten oder unüberwachten Lernmethoden realisiert werden. In Abschnitt 2.5 wird auf einige Klassifikationsmethoden noch genauer eingegangen. Verschiedenste Merkmale wie z.B. Farbhistogramme, Kantenänderungsverhältnis, Bewegungsvektoren, SIFT oder Eckpunkte werden für die Berechnungen verwendet. Lienhart vergleicht in [26] vier gängige Algorithmen für die Shoterkennung. Die Ergebnisse zeigen deutlich, dass ein großer Zusammenhang zwischen gewähltem Algorithmus und untersuchtem Videomaterial, aber auch der Art der Übergänge besteht. Für die Erkennung von harten Schnitten eignen sich demnach Methoden basierend auf Histogrammdifferenzen oder dem Kantenänderungsverhältnis (ECR). Jacobs et al. [27] präsentieren eine ECR-Methode, die auf eine Arbeit von Lienhart [28] zurückgeht. Der ECR zwischen den Frames n und n-k wird folgendermaßen berechnet, wobei σ die Anzahl der Kantenpixel des Frames und X^{in} bzw. X^{out} die Anzahl ein- bzw. ausgehender Kantenpixel repräsentiert:

$$ECR(n,k) = max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-k}^{out}}{\sigma_{n-k}}\right) \tag{2.1}$$

Der gesamte Algorithmus der propagierten Methode besteht aus folgenden Schritten:

1. Umwandlung von Frame n und Frame n-k in Grauwertbilder.
2. Kantenerkennung durch Anwendung von „Canny-Edge“, „Sobel“ oder anderen Kantenerkennungs-Algorithmen.
3. Berechnung der Anzahl der Kantenpixel σ_n und σ_{n-k} aus den Kantenbildern von Schritt 2.
4. Berechnung von imD_n, imD_{n-k} durch Dilatation und imI_n, imI_{n-k} durch Invertierung der Kantenbilder.
5. Erzeugung von Kantenbildern der ein- und ausgehenden Kantenpixel durch UND-Operation imD_n & imI_{n-k} und imD_{n-k} & imI_n.
6. Berechnung der Anzahl ein- und ausgehender Kantenpixel X_n^{in}, X_{n-k}^{out} aus den Kantenbildern von Schritt 5.
7. Berechnung des ECR unter Verwendung der Formel (2.1).

2.2.2 Extraktion von Schlüsselbildern

Nach der Zerlegung eines Videos in einzelne Shots werden aus diesen Shots Schlüsselbilder extrahiert. Das Ziel ist es, einen Algorithmus zu verwenden, der Schlüsselbilder mit geringer Fehlerrate und hoher Datenkompression generiert. Die Methoden dafür sind vielfältig und werden nach Azra und Shobha [29] in 6 Kategorien eingeteilt:

1. **Sequenzieller Vergleich von Frames**
   Hierbei werden die einem Schlüsselbild nachfolgenden Frames mit diesem verglichen. Dafür können zum Beispiel Farbhistogramme verwendet werden. Weist ein Frame große Unterschiede zum letzten Schlüsselbild auf, wird er als nächstes Schlüsselbild festgelegt. Die Vorteile dieser Methode sind ihre Einfachheit und der dadurch geringe rechnerische Aufwand. Die Nachteile sind, dass die Schlüsselbilder eher lokale als globale Shot-Eigenschaften abbilden, und dass sie unregelmäßig verteilt sind bzw. ihre Anzahl nicht vorhersehbar ist.
2. **Globaler Vergleich von Frames**
   Dieser Ansatz basiert auf der Minimierung einer Zielfunktion, die abhängig von der Anwendung vier verschiedene Formen annehmen kann:
   a) Gleichmäßige zeitliche Verteilung
      Die Schlüsselbilder innerhalb eines Shots werden so gewählt, dass sie eine gleichmäßige zeitliche Verteilung haben. Die verwendete Zielfunktion kann als Summe der Differenzen zwischen den zeitlichen Varianzen aller Shot-Segmente angenommen werden. Eine Möglichkeit, die zeitliche Varianz eines

Shot-Segments festzulegen ist, die Differenz zwischen erstem und letztem Frame im Shot zu berechnen [5].

b) Maximale Abdeckung
Bei dieser Methode wird die Anzahl der Frames, die durch ein Schlüsselbild repräsentiert werden können, maximiert. Für die Suche nach den Schlüsselbildern wird ein „Greedy-Algorithmus“ eingesetzt.

c) Minimale Übereinstimmung
Im Zentrum steht die Auswahl von Schlüsselbildern, die eine minimale Übereinstimmung aufweisen.

d) Minimaler Rekonstruktionsfehler
Hierbei werden Schlüsselbilder generiert, die den Rekonstruktionsfehler zwischen ursprünglichen und aus Schlüsselbildern rekonstruierten Frames minimieren.

Im Gegensatz zu Variante 1 werden hier globale Shot-Charakteristika abgebildet. Die Anzahl der Schlüsselbilder ist kontrollierbar, die Redundanz geringer. Diese Vorteile haben allerdings den Preis eines größeren rechnerischen Aufwands.

3. **Verwendung eines Referenzframes**
Die Schlüsselbilder werden in diesem Fall durch den Vergleich mit einem Referenzframe ausgewählt. Vorteil dieser Methode ist wiederum die Einfachheit, allerdings ist die Funktionalität stark abhängig von der Wahl des Referenzframes.

4. **Frame-Cluster**
Unter Verwendung von Clustering-Algorithmen werden Frame-Cluster gebildet. Als Schlüsselbilder definiert man jene Frames, die die geringste Entfernung zum Zentrum eines Clusters aufweisen. Die globalen Eigenschaften eines Videos können dadurch gut abgebildet werden, allerdings ist es oft schwer, semantisch aussagekräftige Cluster zu finden.

5. **Kurvendarstellungsbasiert**
Alle Frames eines Shots werden als Punkte im Merkmalsraum dargestellt. Ihre sequenzielle Verbindung bildet eine Kurve. Als Schlüsselbilder werden jene Frames gewählt, die die Kurvenform am besten repräsentieren. Der Vorteil dieser Methode ist, dass die sequenzielle Information erhalten bleibt. Hoher Rechenaufwand ist der Nachteil.

6. **Objekt- oder eventbasiert**
Jene Frames, die ein Objekt oder Event von Interesse beinhalten, also semantische Bedeutung haben, werden als Schlüsselbilder gewählt. Der Nachteil dabei ist die starke Abhängigkeit von der Qualität der Objekt- bzw. Eventerkennung.

Im nächsten Kapitel werden die Methoden zur Extraktion visueller Merkmale erläutert, die nach der Videosegmentierung zur Anwendung kommen. Außerdem wird ein Überblick

der zur Gesichtserkennung und -verifizierung und zur Emotionserkennung notwendigen Schritte gegeben.

2.3 Bildanalyse

Bevor ein Bild klassifiziert werden kann, müssen zur Aufgabenstellung passende Merkmale aus dem Bild extrahiert werden. Die gewählten Merkmale sollen unabhängig voneinander sein, d.h. keine redundante Information enthalten. In einem möglichen Vorverarbeitungsschritt werden die Bilder für die Merkmalsextraktion bearbeitet, etwa einer Größenanpassung unterzogen, normalisiert oder das Rauschen entfernt. Dabei muss allerdings darauf geachtet werden, dass nicht zu viel relevante Information verloren geht.

Neben der Unterscheidung von „model-“, „shape-“ und „appearance-basierten“ Merkmalen gibt es auch eine Unterscheidung zwischen globalen und lokalen Merkmalen. In der Bildklassifizierung werden am häufigsten „appearance-basierte“ Merkmale verwendet, die entweder global oder lokal extrahiert werden. Lisin et al. stellen als Erweiterung dieses Standards zwei Methoden vor, bei denen globale und lokale Merkmale kombiniert werden [30], allerdings mit der Einschränkung, dass diese Ansätze nur vielversprechende Ergebnisse liefern, wenn bereits im Vorfeld eine grobe Objektsegmentierung vorhanden ist.

2.3.1 Globale Bildmerkmale

Um die gesamte Bildinformation kompakt mit Hilfe eines Merkmalsvektors abzubilden, verwendet man globale Bildmerkmale. Im höher dimensionalen Merkmalsraum können diese Merkmale als Punkt dargestellt werden und eignen sich dadurch für den Einsatz mit jedem Standard-Klassifikator [30]. Die meisten Form- und Texturdeskriptoren fallen in diese Kategorie. Lisin et al. empfehlen sie für die Klassifizierung von Objekten, für die automatische Detektoren existieren wie zum Beispiel für Gesichter und Verkehrsschilder [30]. Als eines der bekanntesten Beispiele für globale Bildmerkmale gelten:

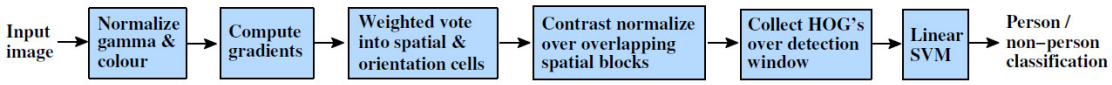

Abbildung 2.2: Schritte der Personenerkennung unter Verwendung von HOG-Merkmalen, Quelle: Dalal und Triggs [15]

Histogram of Oriented Gradients (HOG):
Diese wurden bei ihrer Einführung von Dalal und Triggs [15] für die Personenerkennung propagiert und basieren auf einer lokalen Gradientenverteilung. Für die Berechnung werden die Eingabebilder in gleichmäßig angeordnete Zellen unterteilt, für deren Pixel Orientierungshistogramme der Gradienten erstellt werden. Dalal und Triggs [15] empfehlen dafür eine Zellengröße von 8 x 8 Pixeln und ein Maximum von 9 Histogramm-Bins

bei der Verwendung von Werten (ohne Vorzeichen) von 0 bis 180 Grad für die Gradientenorientierung. Die Bin-Zuordnung im Histogram wird durch die Gradientenorientierung bestimmt; der Betrag bestimmt das Gewicht. Im nächsten Schritt werden mehrere benachbarte Zellen zu Blöcken zusammengefasst und die lokale Gradientenverteilung in Relation zu einer größeren Umgebung gesetzt. Dalal und Triggs [15] verwenden Blöcke die aus 2 x 2 Zellen bestehen und sich zu 50% überlappen. Die vorliegenden Zellen-Histogramme werden mit der Summe der Gewichte aller Gradienten im Block normalisiert, um eine größere Unabhängigkeit von Belichtungsänderungen zu erreichen. Abbildung 2.2 gibt einen zusammenfassenden Überblick der einzelnen Teilschritte. Ein Gesichtsbild und die daraus extrahierten HOG-Merkmale sind in Abbildung 2.3 gegenüber gestellt.

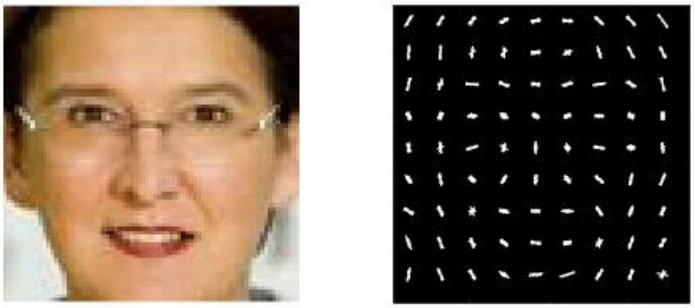

Abbildung 2.3: Gesicht: Original und extrahierte HOG-Merkmale

Dalal und Triggs [15] haben in der automatischen Fußgängererkennung mit den HOG-Merkmalen in Kombination mit SVM als Klassifikationsmethode bessere Ergebnisse erzielt als bei der Verwendung von Wavelet- oder PCA-SIFT-Merkmalen.

2.3.2 Lokale Bildmerkmale

Lokale Merkmale werden für mehrere kleine Bildregionen berechnet und haben den großen Vorteil, dass sie robuster gegenüber Verdeckungen und Bildstörungen sind. Der Vergleich von Bildern kann allerdings schwieriger sein, da die Anzahl der extrahierten Merkmalsvektoren von Bild zu Bild unterschiedlich ist. Je nach Region, für die die Merkmale berechnet werden, unterscheidet man nach Zhang et al. [31]:

- **Pixelbasierte Merkmale**
  Hier werden die Merkmale wie z. B. Grau- oder Farbwerte für jedes Pixel berechnet. Diese Gruppe ist aber, wie auch die globalen Bildmerkmale, anfällig auf Belichtungsänderungen, Rauschen oder Skalierungen.

- **Patchbasierte Merkmale**
  Diese Merkmale wie z. B. Farb-, Textur- oder Kantenwerte werden für kleine Bildregionen, die sogenannte „Points of interest“ umschließen, berechnet. Zu den

bekanntesten patchbasierten Merkmalen, die in der Objekterkennung eingesetzt werden, zählen SIFT- und die in der Berechnung schnelleren SURF-Merkmale.

Speeded-up robust Features (SURF):
Die von Bay et al. [17] entwickelten rotations- und skalierungsinvarianten Merkmale haben vom Aufbau Ähnlichkeit mit SIFT-Merkmalen, allerdings sind sie robuster und in der Berechnung schneller. Die höhere Berechnungsgeschwindigkeit basiert hauptsächlich auf der Verwendung von Integralbildern [32]. Diese haben den großen Vorteil, dass für die Intensitätsberechnung innerhalb eines rechteckigen Bildausschnittes nur 3 Additionen nötig sind. Die Berechnungszeit ist also unabhängig von der Größe des Ausschnittes, wodurch die verwendete Faltung mit Mittelwertfiltern mit konstanter Geschwindigkeit durchgeführt werden kann. Nachfolgend ein kurzer Überblick der zentralen Punkte des SURF-Algorithmus, die detaillierte Beschreibung liefert [17].

Im ersten Schritt werden sogenannte „Points of interest“ im Bild lokalisiert. Das sind in diesem Fall Bildpunkte $x = (x, y)$, an denen die Determinante der Hesse-Matrix $H(x, \sigma)$ ein lokales Maximum erreicht. $L_{xx}(x, \sigma)$ repräsentiert die 2. partielle Ableitung der Intensitätswerte nach xx, gefaltet mit einem Gaußfilter mit Standardabweichung σ. Die Glättung wird durchgeführt, um die Empfindlichkeit gegen Bildrauschen zu vermindern. L_{xy} und L_{yy} sind analog dazu nach xy und yy abgeleitet.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \tag{2.2}$$

Bay et al. [17] approximieren die Gaußfilter-Faltung mit einer Standardabweichung $\sigma = 1.2$ bei der Berechnung der Elemente der Hesse-Matrix durch Mittelwertfilter, angewendet auf die zuvor berechneten Integralbilder. Abbildung 2.4 zeigt ein Beispiel dafür.

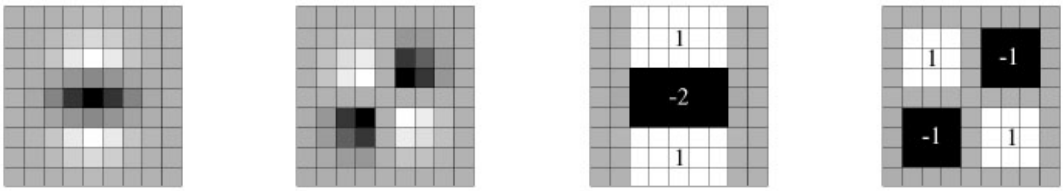

Abbildung 2.4: Die 2. partiellen Ableitungen nach y (L_{yy}) und nach xy (L_{xy}) und die entsprechenden Mittelwertfilter-Approximationen D_{yy} und D_{xy}, Quelle: Bay et al. [17]

Die approximierten Determinantenwerte werden mit einer Gewichtung $w = 0.9$ folgendermaßen berechnet:

$$det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{2.3}$$

Diese Werte und ihre lokalen Maxima („Points of interest“) werden für verschiedene Skalierungsstufen kalkuliert und gespeichert. Das geschieht durch die Anwendung des schrittweise vergrößerten Mittelwertfilters auf die erzeugten Integralbilder (Abbildung

2.5). Ausgehend von einem 9×9 Filter werden Filter der Größe 15×15, 21×21 und 27×27 für den Aufbau der ersten Oktave des Skalierungsraumes eingesetzt. Die gleichbleibenden Berechnungskosten dafür sind der große Vorteil gegenüber der z.B. bei SIFT-Merkmalen für diesen Schritt üblichen Verkleinerung des Originalbildes. Jede Oktave umfasst 4 Filter wobei der Vergrößerungsfaktor zwischen den Filtern für jede Oktave verdoppelt wird, ausgehend von 6 auf 12, 24 und 48. Filter der zweiten Oktave haben daher folgende Größe: 15×15, 27×27, 39×39 und 51×51. Skalierungsinvarianz wird erreicht, indem nur Punkte, die über alle Oktaven lokale Maxima repräsentieren, als „Points of interest“ definiert werden. Das Ziel im nächsten Schritt ist es, Rotationsinvarianz

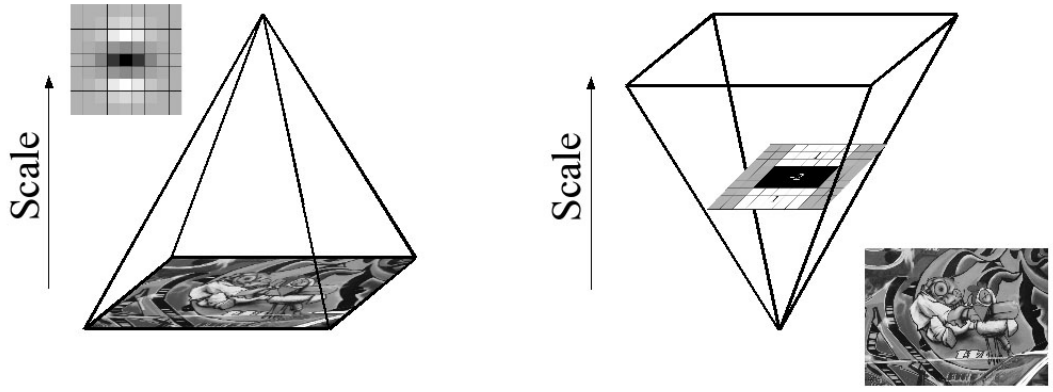

Abbildung 2.5: Aufbau des Skalierungsraumes bei SIFT- (links) und SURF-Merkmalen (rechts), Quelle: Bay et al. [17]

zu erreichen. Dafür wird zuerst die Orientierung der „Points of interest“ mit Hilfe von Haar-Wavelet-Filtern (Abbildung 2.6) innerhalb einer Kreis-Region mit Radius $6s$ (s = Skalierungsfaktor der Punktebene) berechnet. Die gewonnenen Werte werden als Punkte in der Ebene dargestellt. Danach wird ein Fenster der Größe $\frac{\pi}{3}$ in mehreren Schritten über die Kreis-Region geschoben und lokale Orientierungsvektoren durch Summierung der Punktwerte innerhalb des Fensters berechnet. Der längste dieser Vektoren gibt die endgültige Orientierung eines „Point of interest“ an (Abbildung 2.7). Die aus einem Gesichtsbild extrahierten „Points of interest“ und ihre umschließenden Merkmals-Regionen sind in Abbildung 2.8 dargestellt.

Um im letzten Schritt lokale Merkmalsvektoren aus den die „Points of interest“ umschließenden Regionen zu extrahieren, wird ein Quadrat der Größe $20s$ im jeweiligen „Point of interest“ zentriert und nach der zuvor berechneten Orientierung ausgerichtet. Dieses Quadrat wird in 16 Regionen unterteilt und für jede dieser Regionen wird an 25 Punkten mit Hilfe der Haar-Wavelet-Filter die Gradientenorientierung d_x und d_y in x- und y-Richtung berechnet. Die Summe dieser Werte und ihrer Absolutwerte pro Region bilden den finalen 64-dimensionalen SURF-Merkmalsvektor (Abbildung 2.9). Im Gegensatz dazu sind SIFT-Merkmalsvektoren 128-dimensional.

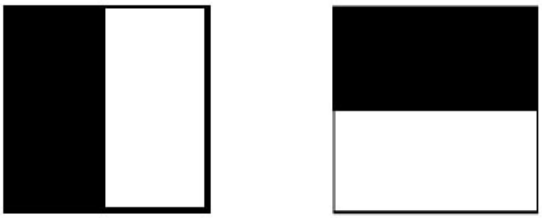

Abbildung 2.6: Haar-Wavelet-Filter für x- und y-Richtung, dunkle Regionen haben das Gewicht -1 und helle +1, Quelle: Bay et al. [17]

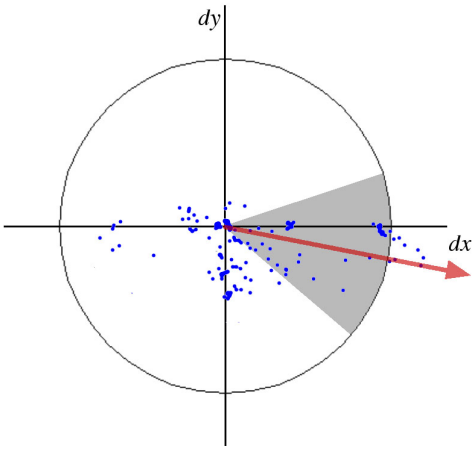

Abbildung 2.7: Berechnung der dominierenden Orientierung eines „Point of interest“ mit Hilfe eines Fensters, das in mehreren Schritten über die Kreisregion geschoben wird, Quelle: Bay et al. [17]

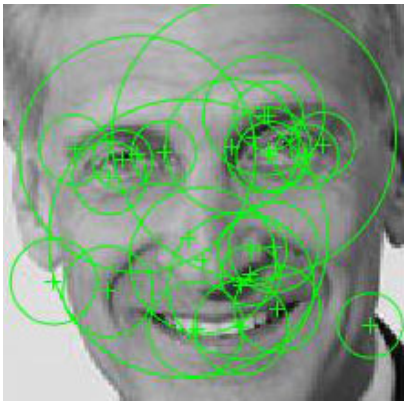

Abbildung 2.8: „Points of interest“ und ihre umschließenden Merkmals-Regionen in einem Gesichtsbild

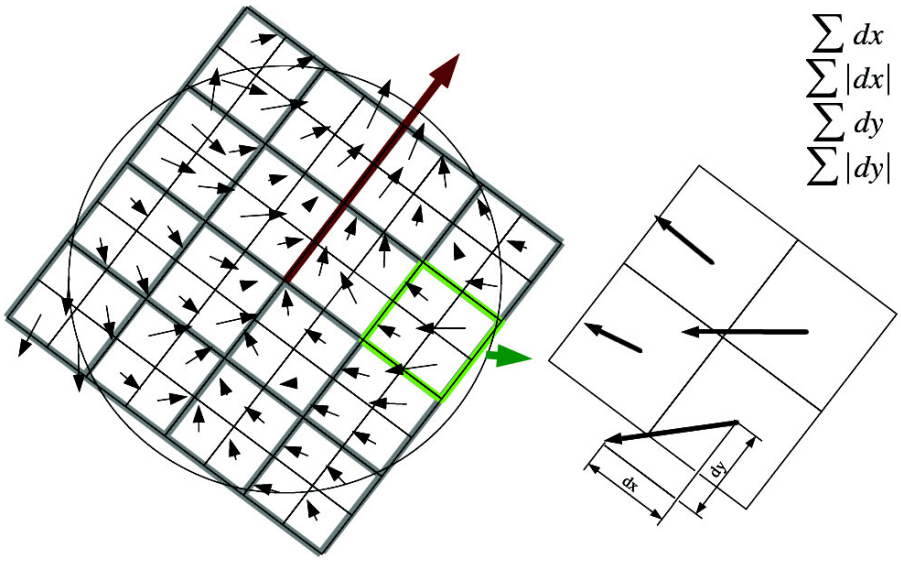

Abbildung 2.9: Berechnung des 64-dimensionalen SURF-Merkmalsvektors aus der einen „Point of interest“ umschließenden Region, Quelle: Bay et al. [17]

Du et al. [18] stellen in ihren Untersuchungen fest, dass mit SURF-Merkmalen bessere Ergebnisse in der Gesichtsverifizierung erzielt werden können als mit SIFT-Merkmalen. Diese Aussage bezieht sich hauptsächlich auf die Geschwindigkeit. Die Erkennungsrate ist bei beiden Merkmalsarten ähnlich bzw. bei der Verwendung von 128-dimensionalen SURF-Merkmalsvektoren etwas höher. Asha und Sreeraj [19] verwenden SURF-Merkmale für die inhaltsbasierte Videosuche und erzielen bei der Suche nach ähnlichen Videoclips eine durchschnittliche Genauigkeit von 75%.

2.3.3 Gesichtserkennung und -verifizierung

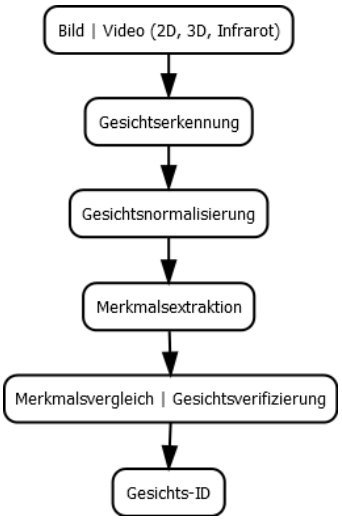

Abbildung 2.10: Schritte eines Systems zur Gesichtsverifikation, Quelle: in Anlehnung an Wójcik et al. [33]

Objekterkennung ist eine der größten Herausforderungen in der Computervision. Dieses Gebiet wird in zwei unterschiedliche Bereiche eingeteilt: die Erkennung von Objektklassen und die Erkennung von Objektinstanzen. Die Klassenerkennung beinhaltet die Feststellung des Vorhandenseins von Objekten vordefinierter Objektklassen in einem Eingabebild. Dazu gehört auch die korrekte Lokalisation und Trennung vom Hintergrund [31]. Umgelegt auf Gesichter versteht man darunter die Gesichtserkennung. Nach den Ausführungen von Zhang et al. [31] ist dies das anspruchsvollere Gebiet, auf dem trotz hoher Forschungsaktivität und leistungsfähiger Klassifikatoren noch immer keine zufriedenstellende Erkennungsgenauigkeit erzielt werden kann. Einer der bekanntesten Echtzeit-Gesichtserkennungsalgorithmen ist der „Viola-Jones-Algorithmus“ [32], dessen Basis die Verwendung von Integralbildern, AdaBoost und einer Kaskadenstruktur ist [34]. Die Vorteile der dabei verwendeten Haarlike-Merkmale sind die Berechnungsgeschwindigkeit und ihre einfache Skalierbarkeit.

Der zweite Teilbereich, die Erkennung von Objektinstanzen, lässt sich als vergleichendes Problem bezeichnen. Es geht darum, eine bestimmte Instanz einer Objektklasse, z. B. ein bestimmtes Auto, mit einer Menge an möglichen, gespeicherten Objektinstanzen (im

Beispielfall Automarken) zu vergleichen, um die Instanzzugehörigkeit festzustellen [31]. Übertragen auf Gesichter kann von Gesichtsverifizierung gesprochen werden. Darunter versteht man, dass durch den Vergleich festgestellt wird, um welche Person es sich bei der unbekannten Instanz handelt. In Abbildung 2.10 sind die Teilschritte eines Gesichtsverifizierungssystems abgebildet. Die Gesichtserkennung bildet die Basis für alle weiteren Punkte wie z.B. Gesichtsverifizierung, Schätzung der Kopfhaltung und Emotionserkennung.

Die Qualität der Erkennung und Verifizierung wird von vielen Faktoren beeinflusst wie Verdeckungen, unterschiedliche Gesichtsmerkmale (Brille, Bart, Hautfarbe), Belichtungsänderungen, Größe des Gesichts oder Orientierung zur Kamera. Wójcik et al. [33] halten fest, dass die Gesichtsverifizierung unter Idealbedingungen gute Ergebnisse liefert. Allerdings ändert sich das drastisch, sobald es Abweichungen vom idealen Gesichtsbild gibt.

Beim Ausgangsmaterial können folgende drei Gruppen unterschieden werden [35]:

- **2D-Bilder**
  Am Beginn der Forschungstätigkeit auf diesem Gebiet wurden 2D-Bilder analysiert. Sie sind noch immer die am häufigsten verwendete Datenform und bilden z.B. die Grundlage für die in Kameras eingesetzte Gesichtserkennungssoftware.

- **Videosequenzen**
  Durch die größere Datenmenge der Videosequenzen können die Bilder mit dem höchsten Informationsgehalt für die Gesichtserkennung und -verifizierung gewählt werden; der Rest wird verworfen. Durch Personen-Tracking über mehrere Frames können unterschiedliche Posen und Gesichtsausdrücke erfasst werden. Dadurch ist eine Steigerung der Erkennungsrate möglich.

- **3D- oder Infrarotdaten**
  Der Einsatz dieser Techniken auf dem Gebiet der Gesichtsverarbeitung ist noch wenig erforscht. 3D-Daten sind in der Verwendung sehr rechenintensiv, Infrarotdaten aufgrund der notwendigen Spezialkameras selten verfügbar. 3D-Daten sind robuster gegenüber Licht- und Orientierungsveränderungen. In Infrarotbildern können Blutvenen sehr gut erkannt werden; ein Muster, das für jeden Menschen einzigartig ist.

Die in beiden Bereichen eingesetzten Techniken überschneiden sich teilweise, und es gibt diverse unterschiedliche Versuche einer Gliederung. Yang et al. [36] propagieren die vier Kategorien „Knowledge-based“, „Feature invariant“, „Template Matching“ und „Appearance-based“. Grundsätzlich lässt sich nach Hatem el al. [37] und Masupha et al. [35] eine Einteilung in die drei Gruppen „merkmalsbasiert“, „bildbasiert“ und „hybrid“ treffen:

- **Merkmalsbasierte Verfahren**
  Merkmalsbasierte haben gegenüber den bildbasierten Methoden den Vorteil, dass

sie unabhängig von Rotation und Skalierung funktionieren und schneller in der Berechnung sind. Sie basieren auf der Extraktion von geometrischen Merkmalsvektoren aus relevanten Gesichtsregionen wie Augen, Mund, Kinn und Nase oder auf der Repräsentation des Gesichts durch zusammenhängende Graphen („Elastic Bunch Graph Matching“).

- **Bildbasierte (holistische) Verfahren**
  Holistische Verfahren verwenden im Gegensatz dazu Information aus dem gesamten Bild und werden von Masupha et al. [35] in statistische und auf künstlicher Intelligenz basierende Ansätze unterteilt. Zu den statistischen Methoden zählen z.B. „Principal Component Analysis“ (PCA), Eigenfaces, Fisherfaces und „Linear Discriminant Analysis (LDA)“. Die auf künstlicher Intelligenz basierenden Verfahren beinhalten z.B. Neuronale Netze, „Support Vector Maschines (SVM)“, „Hidden Markov Modelle (HMM) und „Local Binary Patterns (LBP)“. Holistische Verfahren haben einen hohen Berechnungsaufwand, da sie die gesamte Bildinformation verwenden und verlangen eine hohe Korrelation zwischen Trainings- und Testset. Sie erzielen nach Masupha et al. bessere Erkennungsraten als die merkmalsbasierten Ansätze, wenn es keine Verdeckungen, Skalierungen oder Orientierungsveränderungen gibt [35]. Einige dieser Algorithmen werden in Kapitel 2.5 noch genauer beschrieben.

- **Hybride Verfahren**
  Neben den genannten Gruppen gibt es noch hybride Verfahren, die Methoden aus den beiden Bereichen „merkmalsbasiert“ und „holistisch“ kombinieren.

2.3.4 Emotionserkennung

Nur 7% der menschlichen Kommunikation laufen über die Sprache, 38% über Parasprache und 55% über den Gesichtsausdruck [38]. Aufgrund dieser Tatsache spielt die Emotionserkennung aus Gesichtsdaten eine immer wichtigere Rolle in der automatisierten Bildverarbeitung. Mögliche Einsatzgebiete sind die Verwendung für humanoide Roboter, die automatische Erkennung von Müdigkeit bei Autolenkern oder von Stimmungsschwankungen bei psychisch Kranken zur besseren Dosierung der Medikation. Allerdings ist diese Aufgabe noch immer nicht trivial zu lösen, da eine bestimmte Emotion von verschiedenen Personen oft mit Unterschieden im Gesichtsausdruck dargestellt wird. Das gilt auch, wenn eine Person mehrmals dieselbe Emotion darstellt. Die Ergebnisse sind relativ gut bei gestellten, sinken aber dramatisch bei spontanen Gesichtsausdrücken [39]. Ansätze, die überwachte, maschinelle Lernmethoden einsetzen, sind im Moment am erfolgversprechendsten [39].

Je nach Theorie wird von unterschiedlichen Grundemotionen ausgegangen. Paul Ekmann hat aufgrund seiner Forschungsergebnisse sieben Basisemotionen definiert – Trauer, Zorn, Überraschung, Angst, Ekel, Verachtung und Freude, die unabhängig von der Sozialisierung der Personen weltweit gleich erkannt werden [40]. Gemeinsam mit Wally Friesen veröffentlichte er 1978 eine Methode zur Messung von Gesichtsmuskelbewegungen,

das Facial Action Coding System (FACS). Die Bewegung der 26 Gesichtsmuskeln wird dabei in 44 Action Units (AU) eingeteilt – einen Überblick gibt [41]. In den 80er Jahren entwickelten die beiden mit dem Emotion-FACS-System (EmFACS) eine Möglichkeit zur Codierung der sieben Basisemotionen. Dieses System basiert ebenfalls auf der Anwendung des FACS mit den erwähnten 44 AUs. Allerdings werden hier nur Teile von Videos codiert, die aufgrund der AU-Kombination Rückschlüsse auf eine Basisemotion zulassen und nicht das gesamte Video. Lucey et al. [42] verwenden das FACS für die Codierung der Bilder in der Cohn-Kanade Emotionsdatenbank, welche in diversen Emotionserkennungssystemen eingesetzt wird.

Ein Gesichtsausdruck besteht aus den drei Phasen (Abbildung 2.11) „Onset“, „Apex“ und „Offset“. Als „Onset“ wird der Beginn des Gesichtsausdruckes bezeichnet. Dieser steigert sich bis zum Höhepunkt („Apex“) und wird langsam ausgeblendet („Offset“) [43].

Abbildung 2.11: Die Phasen eines Gesichtsaudruckes, Quelle: Wu et al. [44]

Die einzelnen Schritte eines Emotionserkennungssystems können grob wie folgt zusammengefasst werden [39]:

- **Gesichtserkennung und -segmentierung**
- **Ausrichtung des Gesichts**
  Die segmentierten Gesichtsbilder werden üblicherweise auf eine Größe von 16×16 bis 96×96 Pixeln skaliert, eventuell auch rotiert und geneigt, um eine allgemein gültige Geometrie zu erhalten.
- **Merkmalsextraktion aus relevanten Gesichtsregionen (z.B. Augen- und Mundpartie)**
  Es werden entweder geometrische oder „appearance-basierte“ Merkmale verwendet. Die geometrischen Ansätze erkennen Emotionen direkt aus den Merkmalspositionen, indem sie Distanzunterschiede berechnen. „Appearance-basierte“ Merkmale basieren auf Besonderheiten der Gesichtstextur wie Falten oder Wölbungen [44] und sind robuster.
- **Klassifizierung unter Verwendung eines Trainingssets**

Pukhrambam et al. [38] nehmen als Ausgangsdaten für die Emotionserkennung im Gesicht statische Bilder, aus denen sie Merkmale der Augen- und Mundregion extrahieren, da dies die beiden wichtigsten Gesichtspartien für die Emotionserkennung sind. Für die nachfolgende Klassifizierung verwenden sie einen SVM-Klassifikator, für dessen Training Eigenfaces eingesetzt werden. Mit diesem Ansatz erzielen sie zwischen 60% und 95% Erkennungsgenauigkeit, wobei Zorn am schwierigsten und Freude am einfachsten zu erkennen ist. Gupta und Kaur [45] verwenden in ihrem Ansatz Shape-Daten von Augen, Nase, Mund und Kinn. Die Dimension der Merkmalsvektoren wird durch den Einsatz von PCA reduziert. Zur Emotionserkennung wird hier die euklidische Distanz zwischen den extrahierten Merkmalspunkten eingesetzt.

2.4 Audioanalyse

Die automatisierte Analyse von Audiodaten spielt in verschiedenen Bereichen eine Rolle. Sie wird z.B. für die Sprecheridentifikation, die Spracherkennung, die Unterscheidung von Musik und Sprache, die Erkennung von Musikgenres, die Emotionserkennung und die Erkennung von Audioereignissen eingesetzt. Breebaart et al. [11] stellen fest, dass die Unterscheidung von Musik und Sprache die einfachste dieser Aufgaben mit einer Erkennungsgenauigkeit von bis zu 95% ist. Mit steigender Zahl der zu unterscheidenden Audioklassen sinkt allerdings nach Breebaart el al. [11] die Erkennungsgenauigkeit auf etwa 80% bis 94%.

Wie auch die Bildanalyse lässt sich die Audioanalyse in die drei Abschnitte Segmentierung, Merkmalsextraktion und Klassifizierung gliedern (Abbildung 2.1). Im ersten Schritt wird der Audiokanal in überlappende bzw. nicht überlappende Frames unterschiedlicher Länge zerlegt, um das Signal stationär betrachten zu können. Dabei unterscheidet man die drei Gruppen „short-time“, „mid-time“ und „long-time“, die je nach zu lösender Aufgabe eingesetzt werden. Die Abtastrate eines digitalen Audio-CD-Signals liegt bei 44,1 kHz. Das entspricht 44.100 Abtastwerten pro Sekunde, die bei der Umwandlung des Analogsignals in ein Digitalsignal genommen werden. Breebaart et al. [11] verwenden „mid-time“-Frames mit einer Größe von 32.768 Samples und einer Überlappung von 25%. Für die „short-time“-Analyse werden diese „mid-time“-Frames in Unterframes mit einer Größe von 1.024 Samples und einer Überlappung von 50% eingeteilt. „Mid-time“-Frames haben üblicherweise eine Größe zwischen 1 und 10 Sekunden [12]. Giannakopoulos et al. [12] verwenden in ihrer Arbeit 5 Sekunden-Frames, die in 50 Millisekunden-Frames unterteilt werden. Für beide Fälle nutzen sie 50% Überlappung. „Long-time“-Frames schließen meist das gesamte Audiosignal ein.

Vor der Merkmalsextraktion werden die Signalwerte der Frames normalisiert. Bei Stereosignalen kommt üblicherweise nur ein Kanal für die Auswertung zum Einsatz. Um schon im Vorverarbeitungsschritt eine gewisse Datenreduktion zu erreichen, werden Frames, die Stille enthalten, mit Hilfe eines Thresholds für die Frameenergie aussortiert [46]. Danach werden die Signalwerte durch die Multiplikation mit einer Fensterfunktion - üblich ist ein Hamming-Fenster (Glg. 2.4, Quelle: [47]) - an den Rändern geglättet, um

störende Randeffekte zu vermeiden. N steht dabei für die Fensterbreite und k für den aktuellen Index des Eingangssignals.

$$h(k) = 0.54 - 0.46 \cos\left(\frac{2\pi k}{N-1}\right) \qquad k = 1, ..., N \tag{2.4}$$

Die verwendeten Audiomerkmale können in zeit- und frequenzabhängige Merkmale unterteilt werden. Zur ersten Gruppe zählen z.B. „Short-Time-Energy (STE)“, „Zero-Crossing-Rate (ZCR)“ und „Energy-Entropy (EE)“. Die bekanntesten frequenzabhängigen Merkmale sind die „Mel Frequency Cepstral Coefficients (MFCC)“. Giannakopoulos et al. [12] verwenden für die automatische Depressionserkennung einen 21-dimensionalen Merkmalsvektor aus zeit- und frequenzabhängigen „short-time“-Merkmalen. Auf „mid-time“-Ebene werden aus diesen Merkmalen Kenngrößen wie Standardabweichung σ, Mittelwert μ und das Verhältnis zwischen Standardabweichung und Mittelwert $\frac{\sigma}{\mu}$ berechnet.

Breebaart et al. [11] vergleichen in ihrer Arbeit vier verschiedene Arten von Merkmalen für die Unterscheidung der fünf Audioklassen „klassische Musik“, „Musik (ohne klassischer Musik)“, „Sprache“, „Applaus, Jubel“ und „Hintergrundgeräusche“. Bei den vier verwendeten Merkmalsgruppen handelt es sich um „low-level Merkmale“, MFCCs, psychoakustische Merkmale wie Rauheit, Lautstärke und Schärfe, und ein akustisches Modell das Hüllkurvenschwankungen abbildet. Für die Klassifizierung wird in allen Fällen eine quadratische Diskriminanzanalyse eingesetzt. Die Ergebnisse zeigen deutlich, dass je nach Aufgabenstellung unterschiedliche Merkmale besser geeignet sind. Für die Klassen „klassische Musik“ und „Sprache“ wurden die besten Resultate mit „low-level“-Merkmalen erzielt - Erkennungsgenauigkeit 96% und 88%. Für „Musik (ohne klassischer Musik)“ eignete sich das getestete akustische Modell am besten - Erkennungsgenauigkeit 91%, und für die Klasse „Applaus, Jubel“ konnte mit allen getesteten Merkmalskombinationen - außer den MFCCs - eine Erkennungsgenauigkeit von fast 100% erreicht werden.

Cowling und Sitte [47] stellen in ihrer Arbeit fest, dass die Techniken, die gute Ergebnisse in der Erkennung von Sprache und Musik erzielen, für die Erkennung von Umgebungsgeräuschen nicht so gut geeignet sind. Für diese Audioereignis-Erkennung (Lärm, Klatschen, Schreie usw.) werden verbreitet MFCCs verwendet. Mertens et al. stellen hingegen in [48] eine Methode vor, bei der „Linear Frequency Cepstral Coefficients“ (LFCCs) und Merkmale des Modulationsspektrogramms für die Ereigniserkennung eingesetzt werden und erzielen unter Anwendung auf das „TRECVid MED 2011“-Datenset eine Erkennungsrate von 64%.

Eine ausführliche Erklärung ausgewählter Audiomerkmale folgt in den Kapiteln 2.4.1 und 2.4.2.

Für die Klassifizierung im Audiobereich eignen sich nach Breebaart et al. [11] k-Nearest Neighbour (kNN), Gaussian Mixture Models (GMM), Neuronale Netze und Hidden Markov Models (HMM). Cowling und Sitte [47] empfehlen für die Klassifizierung von Umgebungsgeräuschen Dynamic-Time-Warping (DTW). Sie stellen außerdem fest,

dass Techniken, die auf Merkmalen von Wortteilen aufbauen (wie z. B. HMM) nicht für die Identifizierung von Umgebungsgeräuschen geeignet sind, da diesen die phonetische Struktur der Sprache fehlt. Mertens el al. [48] verwenden für die Audioereignis-Erkennung GMMs in Kombination mit einem universellen Hintergrund-GMM. Elizalde et al. [49] propagieren ein i-Vektor-System für die Klassifizierung von Audioszenen und erzielen damit eine Genauigkeit von 65,8%. Auf einige der genannten Konzepte wird in Kapitel 2.5 näher eingegangen.

2.4.1 Zeitabhängige Merkmale

Zeitabhängige Merkmale werden direkt aus den Audiosignalwerten gewonnen, weshalb auch die Komplexität der Berechnung eher niedrig ist [9]. Sie bieten eine einfache Möglichkeit, Audiosignale zu analysieren, müssen aber meist mit den komplizierter zu berechnenden frequenzabhängigen Merkmalen kombiniert werden, um gute Klassifizierungs-Ergebnisse erzielen zu können [13]. Nachfolgend werden einige für diese Arbeit relevante Merkmale genauer besprochen. Dabei repräsentiert $x_i(n), n = 1, ..., W_L$ die Signalwerte des i-ten Frames und W_L die Framelänge.

- **Short-Time-Energy (STE)**
  Die STE wird durch die Summe der quadrierten Signalwerte, normalisiert durch die Framelänge, berechnet [12] (Glg. 2.5, Quelle: [13]).

$$E(i) = \frac{1}{W_L} \sum_{n=1}^{W_L} \left| x_i(n)^2 \right| \tag{2.5}$$

- **Root-Mean-Square (RMS)**
  Der RMS approximiert die Lautheit des Audiosignals und wird aus der Quadratwurzel der STE berechnet (Glg. 2.6, Quellen: [50], [25]).

$$R(i) = \sqrt{\frac{1}{W_L} \sum_{n=1}^{W_L} \left| x_i(n)^2 \right|} \tag{2.6}$$

- **Zero-Crossing-Rate (ZCR)**
  Die ZCR gibt die Anzahl der Vorzeichenwechsel entlang des Signals, dividiert durch die Signallänge, an. Sie wird als Messwert für das Rauschen eines Signals interpretiert und zeigt normalerweise höhere Werte bei verrauschteren Signalen (ohne Sprache) [12] (Glg. 2.7, Quelle: [51]).

$$Z(i) = \frac{1}{2W_L} \sum_{n=1}^{W_L} |sgn[x_i(n)] - sgn[x_i(n-1)]| \tag{2.7}$$

  wobei $sgn()$ folgende Funktion ist (Quelle: [51]):

$$sgn\left[x_i\left(n\right)\right] = \begin{cases} 1, x_i\left(n\right) \geq 0, \\ -1, x_i\left(n\right) < 0 \end{cases} \tag{2.8}$$

- **Energy-Entropy (EE)**
  Die EE ist eine Kennzahl für abrupte Änderungen im Schallenergielevel des Audiosignals. Niedrigere Werte deuten auf abrupte Änderungen hin. Für die Berechnung werden die „short-time"-Frames in K nicht überlappende Unterframes zerlegt, für die jeweils ihre Energie (Summe der quadrierten Signalwerte), dividiert durch die Gesamtenergie des „short-time"-Frames, kalkuliert wird (Glg. 2.9, Quelle: [13]). Am Schluss wird die Entropie $H\left(i\right)$ (Glg. 2.12, Quelle: [13]) der normalisierten Energiewerte berechnet [12].

  $$e_j = \frac{E_{subFrame_j}}{E_{shortFrame_i}} \tag{2.9}$$

  wobei $E_{subFrame_j}$ folgendermaßen kalkuliert wird (Quelle: [13]):

  $$E_{subFrame_j} = \sum_{n=1}^{W_L} \left|x_i\left(n\right)^2\right| \tag{2.10}$$

  und $E_{shortFrame_i}$ (Quelle: [13]):

  $$E_{shortFrame_i} = \sum_{k=1}^{K} E_{subFrame_k} \tag{2.11}$$

  und abschließend die Entropie $H\left(i\right)$:

  $$H_i = -\sum_{j=1}^{K} e_j \cdot log_2\left(e_j\right) \tag{2.12}$$

2.4.2 Frequenzabhängige Merkmale

Um frequenzabhängige Merkmale extrahieren zu können, muss das zeitbasierte, digitale Signal in einem ersten Schritt durch die Anwendung der diskreten Fourier-Transformation (DFT) in seine Frequenzanteile zerlegt werden. Die dadurch gewonnenen DFT-Koeffizienten werden für die Berechnung der frequenzabhängigen Merkmale verwendet. Dabei ist es ausreichend, nur die erste Hälfte der Koeffizienten zu verwenden, da die zweite Hälfte hauptsächlich der Rekonstruktion des Originalsignals dient [13]. Nachfolgend gibt es eine kurze Beschreibung der für diese Arbeit wichtigen Merkmale. Wf_L steht in den Berechnungen für die Anzahl der verwendeten DFT-Koeffizienten und $X_i\left(k\right), k = 1, ..., Wf_L$ repräsentiert die Werte der DFT-Koeffizienten des i-ten Audioframes. Giannakopoulos et al. [13] empfehlen für eine bessere Kombinationsmöglichkeit mit

anderen Merkmalen die Normalisierung der Merkmalswerte durch die halbe Abtastrate $\frac{F_s}{2}$.

- **Spectral Crest**
  Der „Spectral Crest“ ist das Verhältnis des maximalen DFT-Koeffizienten zum arithmetischen Mittel der DFT-Koeffizienten [52] (Glg. 2.13, Quelle: in Anlehnung an [52]).

$$SC_i = \frac{\max X_i(k)}{\frac{\sum_{k=1}^{Wf_L} X_i(k)}{Wf_L}} \tag{2.13}$$

- **Spectral Rolloff**
  Der „Spectral Rolloff“ wird für die Unterscheidung von Sprache und „Nicht“-Sprache verwendet. Er steht für jene Frequenz, für die gilt, dass ein bestimmter Prozentsatz C der Spektralenergie darunter liegt [12]. Erfüllt sich für den m-ten DFT-Koeffizienten die Gleichung (Glg. 2.14, Quelle: in Anlehnung an [53]), dann ist er der „Spectral Rolloff“ des i-ten Frames.

$$\sum_{k=1}^{m} X_i(k) = C \sum_{k=1}^{Wf_L} X_i(k) \tag{2.14}$$

- **Spectral Centroid und Spread**
  Der „Spectral Centroid“ (Glg. 2.15 , Quelle: [54]) beschreibt den Schwerpunkt und der „Spectral Spread“ (Glg. 2.16 , Quelle: [13]) die Varianz (Bandbreite) des Spektrums [12].

$$C_i = \frac{\sum_{k=1}^{Wf_L} kX_i(k)}{\sum_{k=1}^{Wf_L} X_i(k)} \tag{2.15}$$

$$S_i = \sqrt{\frac{\sum_{k=1}^{Wf_L} (k - C_i)^2 X_i(k)}{\sum_{k=1}^{Wf_L} X_i(k)}} \tag{2.16}$$

- **Spectral Entropy**
  Die „Spectral Entropy“ wird wie die EE berechnet - allerdings im Spektralbereich [12]. Das Spektrum der „short-time“-Frames wird in N nicht überlappende Unterframes zerlegt. Um die normalisierte spektrale Energie x_i zu berechnen, wird die Energie E_i jedes Unterframes durch die gesamte spektrale Energie des „short-time“-Frames normalisiert (Glg. 2.17, Quelle: in Anlehnung an [55]). Die „Spectral Entropie“ H wird schließlich folgendermaßen kalkuliert (Glg. 2.18, Quelle: [55]):

$$x_i = \frac{E_i}{\sum_{i=1}^{N} E_i} \qquad i = 1, ..., N \tag{2.17}$$

$$H = -\sum_{i=1}^{N} x_i \cdot \log_2 (x_i) \tag{2.18}$$

Giannakopoulos et al. [13] halten fest, dass die Werte der Standardabweichung der „Spectral Entropy“ für Umgebungsgeräusche niedriger sind als für Musik und Sprache.

- **Spectral Flux**
  Der „Spectral Flux“ misst die spektrale Veränderung zweier aufeinanderfolgender Frames. Er wird aus der quadrierten Differenz der normalisierten Spektralwerte der beiden Frames berechnet [12], [53] (Glg. 2.19, Quelle: in Anlehnung an [53]). $N_i(k)$ repräsentiert dabei den k-ten normalisierten DFT-Koeffizienten des i-ten Frames (Glg. 2.20, Quelle: in Anlehnung an [13]).

$$F_i = \sum_{k=1}^{Wf_L} = (N_i(k) - N_{i-1}(k))^2 \tag{2.19}$$

$$N_i(k) = \frac{X_i(k)}{\sum_{l=1}^{Wf_L} X_i(l)} \tag{2.20}$$

- **Grundfrequenz F_0 und Harmonic Ratio**
  Für periodische Signale kann im Zeitbereich die Grundfrequenz in Hz als Kehrwert der Periodendauer T_0 angenommen werden (Glg. 2.21, Quelle: [13]). Der Term „Pitch“ (Tonhöhe), wird oft äquivalent zur Grundfrequenz verwendet, obwohl es strenggenommen nicht genau dasselbe ist [13]. Die Periodendauer kann mit Hilfe der Autokorrelation geschätzt werden. Dafür wird das Signal verschoben und für jede Verschiebung wird die Korrelation mit dem Originalsignal berechnet. Als gesuchte Periodendauer gilt jene Verschiebung m, für die die Autokorrelation (Glg. 2.22, Quelle: [13]) maximal ist. W_L steht dabei für die Anzahl der Samples im Frame.

$$F_0 = \frac{1}{T_0} \tag{2.21}$$

$$R_i(m) = \sum_{n=1}^{W_L} x_i(n)\, x_i(n-m) \tag{2.22}$$

Die Autokorrelationsfunktion wird für die weiteren Berechnungen normalisiert (Glg. 2.23, Quelle: [13]). Der „Harmonic Ratio“ berechnet sich aus dem Maximum der normalisierten Autokorrelation. T_{min} und T_{max} begrenzen dabei den Wertebereich der Periodendauer, wobei T_{min} normalerweise die Verschiebung ist bei der der erste Nulldurchgang der normalisierten Autokorrelation stattfindet (Glg. 2.24, Quelle: [13]).

$$\Gamma_i(m) = \frac{R_i(m)}{\sqrt{\sum_{n=1}^{W_L} x_i(n)^2 \sum_{n=1}^{W_L} x_i(n-m)^2}} \tag{2.23}$$

$$HR_i = \max_{T_{min} \leq m \leq T_{max}} \{\Gamma_i(m)\} \tag{2.24}$$

- **Mel Frequency Cepstral Coefficients (MFCCs)**
  MFCCs sind die Koeffizienten der diskreten Kosinustransformation (DCT) der Mel-skalierten Werte des logarithmierten Spektrums [12]. In der traditionellen Berechnung der MFCCs wird der 0. Koeffizient im Merkmalsvektor weggelassen. Zheng et al. [10] stellen hingegen fest, dass es sinnvoll ist, ihn zu verwenden. Die Performance der MFCCs hängt von der Filteranzahl, Form der Filter und ihrer Verteilung/Überlappung ab [10]. In vielen Applikationen werden nur die ersten 13 MFCCs verwendet, weil sie genug Information für die Klassifikation beinhalten [13]. Breebaart et al. [11] halten fest, dass sich der 2. MFCC sehr gut für die Unterscheidung der Klasse „Applaus, Jubel“ von anderen Klassen eignet.

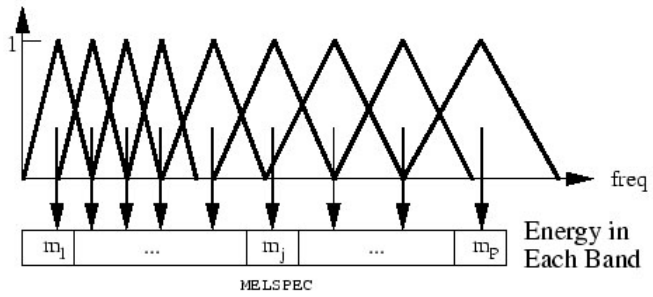

Abbildung 2.12: MEL-Filterbank, Quelle: Murali et al. [46]

Die Berechnung der MFCCs umfasst nach Murali et al [46] folgende Schritte (nach dem Vorverarbeitungsschritt und der Multiplikation mit einem Hamming-Fenster):

1. FFT
2. Logarithmierung der fouriertransformierten Werte
3. Mel-Filterung
4. DCT
5. Δ MFCCs (optional)
6. $\Delta\Delta$ MFCCs (optional)

Die unter Punkt 3 verwendete Filterbank besteht üblicherweise aus Dreiecksfiltern (Abbildung 2.12). Murali et al. [46] verwenden folgende Formel für die Transformation der Spektralwerte in Cepstralwerte (Glg. 2.25, Quelle: [46]):

$$f\left(mel\right) = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{2.25}$$

Die Δ MFCCs enthalten die Veränderung zwischen den Frames in Bezug auf die MFCCs und die ΔΔ MFCCs die Veränderung zwischen den Frames in Bezug auf die Δ MFCCs [46].

2.4.3 Spracherkennung

Die Spracherkennung ist ein Teilbereich der Audioverarbeitung. Sie spielt in der vorliegenden Arbeit eine untergeordnete Rolle. Trotzdem soll ein kurzer Überblick über ihre Konzepte und die gängigsten Open-Source-Spracherkennungstoolkits gegeben werden. Bei der Spracherkennung wird zwischen einer sprecherabhängigen und einer sprecherunabhängigen Variante unterschieden. Die sprecherabhängigen Systeme werden mit Sprachmustern von einer speziellen Person trainiert und sind dadurch in der Verwendung eingeschränkter [56]. Die statistischen Modelle zur Repräsentation von Sprache gliedern sich in Sprach- und Akustikmodelle. Im Sprachmodell wird die Grammatik und das Vokabular der Sprache abgebildet. Das Akustikmodell speichert die dazugehörige Aussprache [56]. Auf dem Markt gibt es diverse kommerzielle und freie Softwarepakete, wie z.B. AT&T Watson, Microsoft Speech Server, Google Speech API, Nuance Recognizer, Dragon NaturallySpeaking (die in Windows 7 integrierte Spracherkennung), Simon, Sphinx, Julius, Hidden Markov Model Toolkit (HTK), RWTH Aachen Automatic Speech Recognition System, SHoUT Speech Recognition Toolkit [57].

Gaida et al. [57] vergleichen in ihrer Studie die drei am weitesten verbreiteten Open-Source-Spracherkennungstoolkits nämlich:

- **Hidden Markov Model Toolkit (HTK)**
  HTK wurde ursprünglich an der Cambridge-Universität entwickelt und besteht aus mehreren in C programmierten Modulen, die Funktionen für das Training und Testen von HMMs und die Ergebnisanalyse zur Verfügung stellen [58].

- **CMU Sphinx** (inklusive Pocketsphinx) [59]
  Sphinx ist ein Toolkit der amerikanischen Carnegie Mellon-Universität und beinhaltet eine Spracherkennungsengine sowie Tools, um akustische Modelle zu trainieren [56]. Unterstützt werden z.B. die Sprachen US Englisch, UK Englisch, Französisch, Mandarin, Deutsch, Holländisch und Russisch. Diverse Sprachmodelle stehen zum Download zur Verfügung [59]. Für nicht vorhandene Sprachen gibt es die Möglichkeit, Sprachmodelle selbst zu erstellen. Das Toolkit besteht aus 4 Paketen:

  - Pocketsphinx - Spracherkennungs-Bibliothek in C
  - Sphinxtrain - Trainingstools für akustische Modelle
  - Sphinxbase - Bibliothek, die von Pocketsphinx und Sphinxtrain verwendet wird
  - Sphinx4 - Anpassbare Spracherkennungs-Bibliothek in Java

Pocketsphinx eignet sich besser, wenn auf Geschwindigkeit und Portabilität Wert gelegt wird, Sphinx4 hingegen, wenn Flexibilität und Handling im Vordergrund stehen [59].

- **Kaldi**
  Kaldi ist in C++ implementiert und hat prinzipiell Ähnlichkeiten mit HTK. Allerdings wird hier mehr Wert darauf gelegt, dem User komplette Abläufe zur Verfügung zu stellen [60]. Das Toolkit ist auf der Verwendung von Deep Neural Networks und GMMs aufgebaut [56].

Für Training, Entwicklung und Testung wurde von Gaida et al. [57] mit dem deutschen Sprachkorpus „Verbmobil“ (enthält Dialoge in Deutsch, Englisch und Japanisch) und dem englischen Sprachkorpus „Wall Street Journal 1“ gearbeitet. Damit wurden Sprach- und Akustikmodelle für HDecode, Julius, Pocketsphinx, Sphinx-4 und Kaldi trainiert. Gaida et al. [57] versuchten eine Reihung der Toolkits nach dem Verhältnis Aufwand zu Nutzen und stellten fest, dass HTK den höchsten Aufwand für das Aufsetzen, Verwenden und Optimieren verursacht, gefolgt von Sphinx und Kaldi. Kaldi ist ihrer Meinung nach auch für Nicht-Experten am einfachsten zu bedienen, weil es für Training und Dekodierung bereits fertige Konzepte anbietet, die die aktuellsten Techniken beinhalten - das aber zu Lasten der Berechnungskosten, die hier am höchsten sind. Sphinx stellt ebenfalls eine Trainingspipeline zur Verfügung. Allerdings ist sie nicht so ausgefeilt wie die von Kaldi. HTK hingegen erfordert von Seiten des Users großen Entwicklungsaufwand.

Yang et al. [61] kommen zu dem Ergebnis, dass sich das Open-Source-Toolkit Sphinx am besten für den Einsatz der automatisierten Spracherkennung in Vortragsvideos eignet, weil es dafür akustische Modelle auf Deutsch gibt, es das Sprachvokabular betreffend leichter erweiterbar ist, die Verarbeitung kontinuierlicher Sprache gut funktioniert und die Spracherkennungsrate im Mittelfeld der in [61] getesteten Produkte liegt.

2.5 Klassifizierung

Während der Klassifizierungsphase werden Videos bzw. Videoszenen, die durch die zuvor extrahierten Merkmalsvektoren repräsentiert werden, einer von mehreren bekannten Klassen zugeordnet und zwar der Klasse, deren Ähnlichkeit am größten ist [5]. Je nach Verfahren werden hierfür statistische oder geometrische Größen verwendet. Für die Verarbeitung können verschiedene Merkmalsarten kombiniert werden. Generell unterscheidet man überwachte und unüberwachte Klassifizierungsalgorithmen [21]. Zu den überwachten Methoden zählen etwa kNN, SVM und HMM. Dabei wird während der Trainingsphase mittels Merkmalsvektoren von Objekten, deren Klassenzugehörigkeit bekannt ist, ein Klassenmodell für jede Klasse generiert [21]. Diese Modelle werden danach für die Klassifizierung unbekannter Instanzen verwendet. Allerdings sind die Ergebnisse stark abhängig von einem gut gewählten Trainingsset, das sowohl positive als auch negative Beispiele enthält [5]. Die unüberwachten Verfahren beinhalten Clustering-Methoden wie „k-Means“

und haben den Vorteil, dass kein Trainingsdatenset nötig ist. Hu et al. [5] propagieren folgende Unterteilung der Videoklassifizierung:

- **Nach Editiereffekten**
  Für diese Klassifizierung kommen Strukturen zum Einsatz, die beim Editieren von Videos berücksichtigt werden und Einfluss auf die Zusammensetzung von Shots und Szenen haben. Dazu gehören z.B. Kamerabewegung, Blickwinkel und Einstellungsgröße der Kamera.

- **Nach Videogenre**
  Das Ziel dieser Klassifizierung ist die Zuordnung von unbekannten Videos in die zutreffende Genreklasse wie z.B. „Film“, „Nachrichten“, „Sport“.

- **Nach Videoereignissen**
  Ein Video beinhaltet je nach Genre unterschiedliche Ereignisse wie z.B. Schüsse, Klatschen, Hundegebell, die die Grundlage für diese Art von Videoklassifizierung bilden.

- **Nach Objekten**
  Hierbei erfolgt die Klassifizierung aufgrund von im Video vorkommenden Objekten. Die Gesichtserkennung fällt in dieses Teilgebiet.

Für die vorliegende Aufgabe sind die beiden letzten Klassifizierungsmethoden von Bedeutung.

2.5.1 Überblick Klassifikatoren

Der neueste Ansatz auf diesem Gebiet ist das „Deep Learning“, das auf der Verwendung von Neuronalen Netzen mit mehreren versteckten Schichten basiert, die die Merkmalsextraktion und die Lernphase automatisch durchführen [14]. Der Einsatz von „Deep Learning“-Algorithmen hat sich vor allem für große Datensammlungen bewährt und wird unter anderem für Apple's „Siri“ und Google's „Voice Search“ verwendet [21]. „Deep Learning“ hat allerdings den Nachteil, dass es einen hohen Berechnungsaufwand verursacht. Außerdem ist das vorhandene Trainingsset oft nicht ausreichend für die Verwendung dieser Methode [14]. Deshalb wird auf eine nähere Ausführung und den Einsatz im Rahmen dieser Arbeit verzichtet. Die nachfolgenden Abschnitte liefern hingegen eine nähere Erklärung der für die Implementierung relevanten Ansätze.

2.5.2 Bag of Visual Words

Der „Bag of Visual Words“-Algorithmus (BoVW) für die Bild-Klassifizierung hat vom Konzept her große Ähnlichkeit mit dem Natural-Language-Processing-Algorithmus „Bag of Words“, der für die Klassifizierung von schriftlichen Dokumenten verwendet wird [62]. Dabei werden Textdokumente durch Histogramme, die die Anzahl der Wortvorkommen innerhalb einer Wortfamilie abbilden, repräsentiert [63]. Ein populäres Einsatzgebiet der

„Bag of Words“-Methode ist die Unterscheidung von Spam und Nicht-Spam. Im Bereich der Bildverarbeitung basiert dieser Ansatz auf der Lokalisation von „Points of interest“ (siehe auch 2.3.2) im Bild und der Merkmalsextraktion aus der sie umgebenden Region. Lopes et al. [63] beschreiben den gesamten Prozess folgendermaßen (Abbildung 2.13):

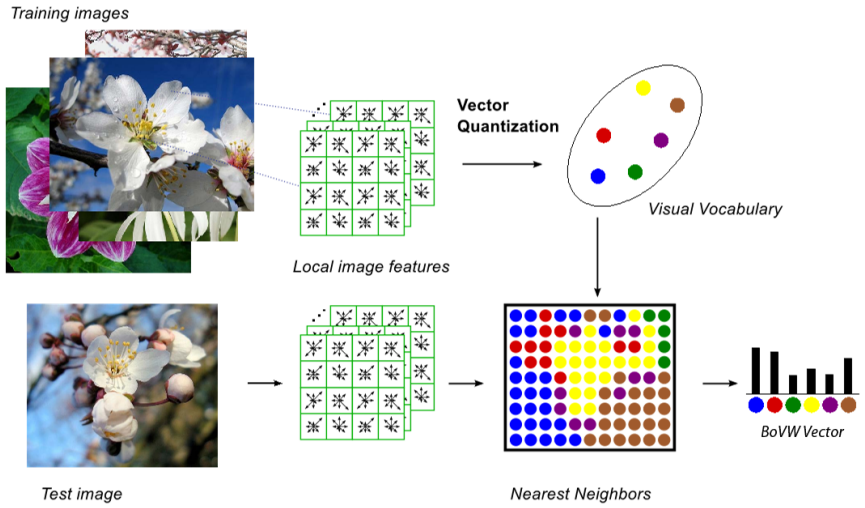

Abbildung 2.13: Schritte der BoVW-Generierung, Quelle: Hentschel und Sack [24]

- **Lokalisation der „Points of interest“**
  Diese erfolgt entweder mit einem Gitter, das über das Bild gelegt wird oder - was eher üblich ist - mit einem „Interest Point Detector“ wie SIFT oder SURF (siehe auch 2.3.2).

- **Beschreibung der Punktumgebung (Merkmalsextraktion)**
  „Interest Point Detectors“ liefern ihre eigenen Merkmalsbeschreibungen. Theoretisch wäre es aber auch möglich, andere Merkmale wie z. B. Grauwerte zu verwenden.

- **Erstellen eines Wortschatzes**
  Die Merkmale aus den Bildern des Trainingssets werden in Gruppen (Cluster) zusammengefasst. Dies geschieht normalerweise mit dem k-Means-Clusteralgorithmus unter Verwendung der euklidischen Distanz.

- **Clusterzuordnung der „Points of interest“**
  In diesem Schritt werden die zuvor extrahierten „Points of interest“ einem Wort im Wortschatz, also dem Cluster mit der größten Ähnlichkeit, zugeordnet. Jeder Cluster repräsentiert ein „visuelles Wort“ und alle „Wörter“ zusammen bilden das „Vokabular“ [62].

- **BoVW-Histogramm**
  Die Häufigkeit der vorkommenden Wörter wird im BoVW-Histogramm abgebildet.

Das so gewonnene BoVW-Histogramm ist eine Beschreibung der Bildinhalte des Trainingssets und wird unter Einsatz eines Klassifikators für das Training eines Klassen-Modells verwendet, welches danach für die Klassifizierung von unbekannten Bildern eingesetzt wird. Als Klassifikatoren haben sich in diesem Fall SVMs etabliert. Hentschel und Sack [24] bestätigen in ihrem Vergleich von möglichen Klassifikatoren für den Einsatz mit dem BoVW-Algorithmus ebenfalls, dass SVMs die besten Ergebnisse liefern. Im Speziellen empfehlen sie die Verwendung von SVMs, die als Kernelfunktion eine radiale Basisfunktion (RBF) mit χ^2-Distanz benutzen. Diese Aussage wird allerdings in der Arbeit von Yang et al. [62] dahingehend relativiert, dass sie RBF-Kernels nur für Aufgaben mit kleinem „Wortschatz“ empfehlen und ansonsten bessere Ergebnisse mit linearen SVMs erzielen.

Lopes et al. [63] verwenden die BoVW-Methode in ihrer Arbeit für die Erkennung von Nacktszenen in Videos. Nachdem Farbinformation in diesem Fall ein wichtiger Aspekt ist, propagieren sie die Verwendung von HueSIFT-Merkmalen als Basis. Als Klassifikator dient eine lineare SVM. Untersucht wird ein Ansatz, bei dem pro Videosegment nur ein Keyframe für die Klassifizierung verwendet wird und ein weiterer, bei dem mehrere Keyframes pro Segment klassifiziert werden. Die Zuordnung erfolgt im zweiten Fall nach dem maximalen Vorkommen der Klasse innerhalb der Sequenz. Lopes et al. [63] erzielen damit innerhalb ihrer Tests eine maximale Erkennungsrate von 93,2%.

2.5.3 Support Vector Machine

Die „Support Vector Machine“ (SVM) zählt zu den überwachten Klassifizierungsverfahren und wird in vielen Bereichen wie z. B. für die Bildklassifizierung, für die Objekterkennung oder für die Kategorisierung von Texten eingesetzt [64]. Das zugrundeliegende Konzept basiert auf der Maximierung der Minimumdistanz von der die Klassen separierenden Hyperebene zu den am nächsten liegenden Punkten [20]. Diese Punkte, die sogenannten Stützvektoren (engl. Support Vectors), sind namensgebend für die Methode (Abbildung 2.14). Durch die Gewinnung eines Maximalabstandes zwischen der Hyperebene und den Stützvektoren auf beiden Seite kann der erwartete Generalisierungsfehler gesenkt werden [22]. Die Hyperebene wird durch folgende Formel definiert, wobei w für den Normalvektor der Hyperebene und b für den Bias steht (Glg. 2.26, Quelle: [64]):

$$wx + b = 0 \tag{2.26}$$

Sind die Datenpunkte zweier Klassen K_1 und K_2 linear separierbar, muss ein Paar (w, b) existieren, so dass folgende Gleichungen erfüllt sind (Glg. 2.27, 2.28, Quelle: [22]):

$$wx_i + b \geq 1 \qquad f\ddot{u}r\, alle\, x_i\, \epsilon\, K_1 \tag{2.27}$$

$$wx_i + b \leq -1 \qquad f\ddot{u}r\, alle\, x_i\, \epsilon\, K_2 \tag{2.28}$$

Eine große Stärke der SVM liegt darin, dass nicht linear separierbare Daten mittels Kernelfunktionen in einen höher dimensionalen Raum transformiert werden können, um dort eine separierende Hyperebene zu finden. Beliebte Kernelfunktionen sind Polynomfunktionen, die RBF und die Sigmoidfunktion [21]. Für den Fall, dass die Daten zu verrauscht

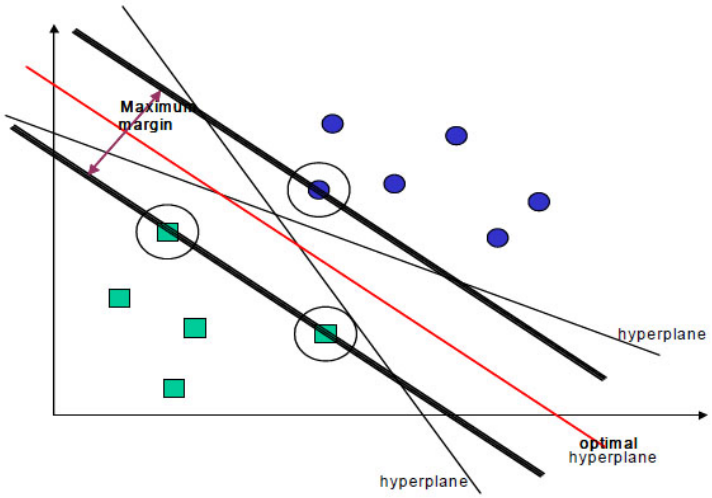

Abbildung 2.14: Binäre Klassentrennung durch eine SVM mit Hyperebene und Stützvektoren, Quelle: Kotsiantis [22]

für eine Separierung sind, werden sogenannte Fehlerterme eingesetzt, die Trainingsfehler gezielt erlauben [65]. Die Entscheidung zwischen dem Einsatz einer linearen SVM oder einer SVM mit Kernelfunktion bestimmt maßgeblich den Erfolg der Methode. Dabei muss bedacht werden, dass Kernelklassifikatoren zwar meist bessere Vorhersageergebnisse liefern, dafür aber höhere Berechnungskosten verursachen [66]. Huang und Lin [66] haben diese Fragestellung näher untersucht und eine automatisierte Methode basierend auf einer multilinearen SVM für die Entscheidungsfindung entwickelt.

Neben der Ein-Klassen-SVM, bei der die Objekte der gesuchten Klasse positiv und alle anderen negativ klassifiziert werden, gibt es auch Multi-Klassen-SVMs für Probleme, die mehr als zwei Klassen beinhalten. Für die Lösung dieser Multi-Klassen-Klassifikationsaufgaben gibt es verschiedene Möglichkeiten. Chamasemani und Singh [67] nennen z. B. SVMs, die gerichtete azyklische Graphen bzw. Binärbäume verwenden oder One-Against-One-SVMs bzw. One-Against-All-SVMs. Die Vorgehensweise soll für die beiden letzteren als wichtigste Vertreter noch kurz näher erklärt werden:

- **One-Against-All-SVM**
  Für Probleme mit M Klassen werden M binäre SVMs trainiert. Dabei werden jeweils die Objekte einer Klasse positiv und die aller anderen negativ markiert. Gesuchte Objekte werden in der Klassifizierungsphase jener Klasse zugeordnet, die von allen SVMs das maximale Ergebnis liefert [67].

- **One-Against-One-SVM**
  In diesem Fall werden für M Klassen $\frac{M \times (M-1)}{2}$ binäre Klassifikatoren trainiert, die jeweils zwei Klassen - eine positiv, eine negativ markiert - verwenden [67]. Für die

Klassifizierung eines unbekannten Objekts werden wiederum alle Klassifikatoren durchlaufen und jene Klasse gewählt, die die meisten Zuordnungen erhalten hat.

Chamasemani und Singh [67] kommen in ihren Tests bezüglich Bildklassifizierung von Schilddrüsenaufnahmen zum Schluss, dass One-Against-All-SVMs im Vergleich zu One-Against-One-SVMs unter Verwendung eines Polynom-Kernels vom Grad 3 bessere Ergebnisse liefern. Wobei bezüglich der Wahl des Kernels hinzugefügt werden muss, dass mit einem RBF-Kernel fast genauso gute Ergebnisse erzielt werden konnten - und das fast doppelt so schnell [67]. Im direkten Vergleich mit dem AdaBoost- bzw. Entscheidungsbaumklassifikator erreichen Chamasemani und Singh [67] mit der One-Against-All-SVM die höchste Erkennungsgenauigkeit von 96,9%.

2.5.4 Gaussian Mixture Model und Hidden Markov Model

Gaussian Mixture Models (GMMs) werden in der Sprach- und Sprechererkennung eingesetzt, aber auch für die Erkennung von akustischen Ereignissen benutzt. Für den letzteren Bereich weisen Pohjalainen et al. [68] darauf hin, dass die in den Studien am häufigsten vorkommenden akustischen Ereignisse Schreie und Explosionsgeräusche sind, für deren Klassifizierung mehrheitlich GMMs verwendet werden. Als Beispiel für die gängige Vorgehensweise dient die Arbeit von Vuegen et al. [23], die in ihrer Untersuchung zur Erkennung von Audioereignissen in Büroszenen MFCC-Merkmale (inklusive Δ und $\Delta\Delta$) in Kombination mit GMMs einsetzen. Ihre Vorgehensweise basiert auf dem Training eines separaten GMMs für jede Ereignisklasse. Zusätzlich wird ein GMM als universelles Hintergrundmodell trainiert. Pohjalainen et al. [68] verwenden denselben Ansatz für die Unterscheidung von Schreien, normaler Sprache und Hintergrundgeräuschen. Ihre GMMs bestehen aus 8 Komponenten mit diagonaler Kovarianz.

Ein GMM modelliert die Wahrscheinlichkeitsverteilung $p(X)$ eines Datensets $X = \{x_1, x_2, ..., x_n\}$ durch eine gewichtete Linearkombination von Gaußverteilungen [69] (Glg. 2.29, Quelle: [69]). Es wird durch die Anzahl der verwendeten Gaußverteilungen, deren Gewichtungen, Mittelwerten und Kovarianzen definiert [70].

$$p\left(X\right) = \sum_{i=1}^{M} w_i N\left(X/\mu_i, \Sigma_i\right) \tag{2.29}$$

Während des Trainings werden die Parameter dahingehend geschätzt, dass der log-likelihood maximiert wird [69]. Vuegen et al. [23] verwenden für das Training ihrer GMMs den „Expectation-Maximation"-Algorithmus (EM). Auch Dufaux et al. [70] maximieren den Likelihood durch 20 EM-Durchläufe. Pohjalainen et al. [68] hingegen nutzen nur 10 Wiederholungen. Für die Klassifizierung wird jene Klasse gewählt, deren GMM-Posterior-Wahrscheinlichkeit am höchsten ist.

Die maximale Precision (p) auf Frame-Basis, die Vuegen et al. [23] mit ihrem Setting erzielen konnten, liegt bei 73,39%. Mertens et al. [48] verwenden für die Ereigniserkennung

in Videos - basierend auf akustischen Merkmalen - ebenfalls einen GMM-Ansatz mit universellem Hintergrundmodell und erzielen damit eine Erkennungsgenauigkeit von 64%.

Neben den GMMs sind hier noch kurz die HMMs zu erwähnen, da diese beiden Ansätze oft gegenübergestellt bzw. auch gemeinsam verwendet werden. Dufaux et al. [70] vergleichen etwa diese beiden Klassifikatoren für die Klassifizierung der 6 Audioereignisse „Schreie“, „Explosionen“, „Schüsse“, „zuschlagende Türen“, „Glasbruch“ und „Andere“.

HMMs bestehen aus N Zuständen und befinden sich zu jedem Zeitpunkt t in einem dieser Zustände [71]. Für Markov-Ketten erster Ordnung hängen die Übergangswahrscheinlichkeiten a_{11} bis a_{NN} immer nur vom vorhergehenden Zustand ab [71]. Es gibt verschiedene Arten von HMMs wie z. B. Links-Rechts-HMMs, bei denen das Modell und damit die Modellzustände q_1 bis q_N linear von links nach rechts durchlaufen wird (Abbildung 2.15), oder das in der Spracherkennung häufig verwendete Bakis-HMM, bei dem zusätzlich das Überspringen des unmittelbar folgenden Zustandes möglich ist [71]. Für diese Modelle gilt, dass ein einmal verlassener Zustand nicht wieder erreicht werden kann. Die Struktur der HMMs erlaubt eine Abbildung von Sprachsignalen als Folge einzelner Laute. Für das Training der HMMs kommt als Standard der Baum-Welsh-Algorithmus zum Einsatz. Der Likelihood während der Klassifizierungsphase wird mittels Viterbi-Algorithmus berechnet (eine genaue Beschreibung dieser Algorithmen liefert Euler [71]). Dufaux et al. [70] testen neben dem GMM-Klassifikator Links-Rechts-HMMs mit drei verborgenen Zuständen und 20 Baum-Welsh-Durchläufen.

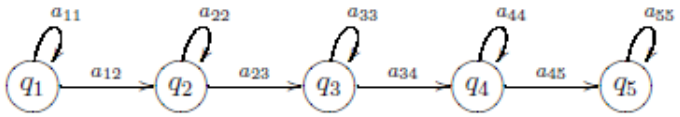

Abbildung 2.15: Links-Rechts-HMM, Quelle: Euler [71]

2.5.5 Dynamic Time Warping

Dynamic Time Warping (DTW) ist eine Methode, bei der zwei Signale durch Distanzminimierung nicht linear aufeinander abgebildet werden [72]. Im Fall der Erkennung einzelner Wörter bedeutet das, dass die Folge der extrahierten Merkmalsvektoren des gesuchten Wortes X mit der Folge der Merkmalsvektoren des Referenzwortes W verglichen und der Unterschied als „Distanz“ berechnet wird [73] (Abbildung 2.16). Für diese Kostenberechnung kann z.B. die euklidische Distanz oder die Manhattan-Distanz verwendet werden [13]. Beim DTW wird dafür mittels dynamischer Programmierung ein Kostenraster, beginnend bei Knoten (1,1), durchlaufen und für jeden Knoten die summierten Kosten, um ihn zu erreichen, berechnet [13]. Ziel ist es, einen optimalen Pfad mit möglichst geringen Kosten zu finden. Generell gilt: Je größer die Distanz, desto unterschiedlicher sind die beiden Signale. Ein Vorteil des DTW ist die Tatsache, dass der Algorithmus Signale unterschiedlicher Länge verarbeiten kann. Das ist deshalb wichtig,

weil selbst einzelne Schlagworte bei mehrmaliger Aussprache nicht immer gleich lang sind. Murali et al. [46] verwenden DTW in Verbindung mit MFCCs (39-dimensional inklusive

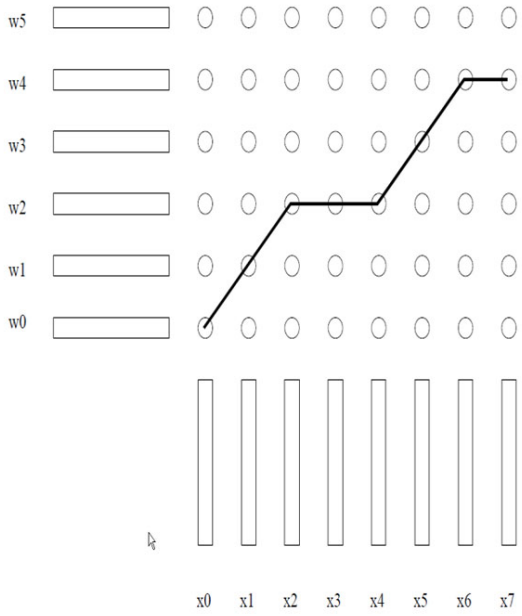

Abbildung 2.16: Mögliche Verknüpfung zweier Wörter durch ein Kostenraster des DTW, Quelle: Lama und Namburu [73]

Δ und $\Delta\Delta$) für die sprecherunabhängige Emotionserkennung aus Einzelwörtern. Mittels DTW werden die Distanzen des Eingabewortes zu den Ground-Truth-Wörtern berechnet und anschließend für die Emotionsklassifizierung mittels SVM benutzt. Murali et al. [46] erreichen damit für die vier Emotionsklassen Trauer, Freude, Zorn, Überraschung plus einer Klasse für neutrale Emotion eine Erkennungsgenauigkeit zwischen 89% und 94%.

2.6 Methoden zur statistischen Evaluierung

Um festzustellen, ob die Klassifikatoren und Merkmale passend gewählt wurden, müssen die Ergebnisse der Klassifizierung evaluiert werden. Wichtige Grundlage für die korrekte Auswertung bilden die Ground-Truth-Daten, die je nach Klassifikationsaufgabe aus korrekt gekennzeichneten Bildern, Audio- oder Videodaten bestehen [74]. Hossin und Sulaiman [75] erklären in ihrer Arbeit Metriken, die für die Messung der Leistungsfähigkeit von Binär- und Multiklassen-Klassifikatoren eingesetzt werden können. Aus der Menge der möglichen Evaluierungskennzahlen werden die für diese Arbeit relevanten nun erläutert.

2.6.1 Statistische Kennzahlen

Ein wichtiges Instrument für die Evaluierung ist die „Confusion Matrix“, aus der die nachfolgend erklärten Kennzahlen Accuracy (acc), Precision (p) und Recall (r) direkt

abgeleitet werden können [13]. Mit acc lässt sich der Klassifikator als Ganzes bewerten, p und r geben Auskunft über die Performance des Klassifikators in Bezug auf einzelne Klassen [13]. In der „Confusion Matrix“ werden die Ergebnisse der Klassifizierung dargestellt. Dabei stehen die Zeilen für die Originalklassen und die Spalten für die Resultate der Klassifizierung. In der Diagonale der „Confusion Matrix“ befindet sich die Anzahl der korrekt klassifizierten Instanzen [76] (Abbildung 2.17).

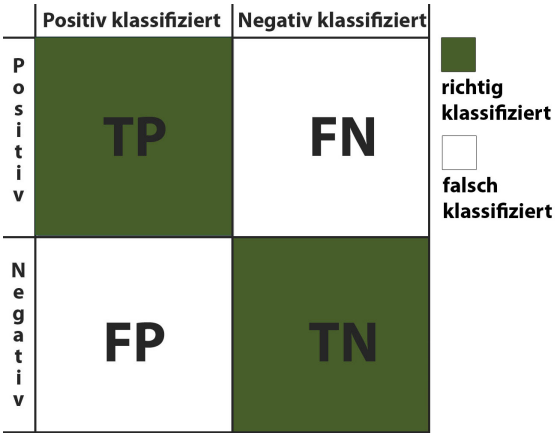

Abbildung 2.17: „Confusion Matrix“ eines binären Klassifikators, Quelle: in Anlehnung an Marom et al. [76]

- **Accuracy (acc)**
  Die Accuracy (acc) gehört zu den am häufigsten verwendeten Kennzahlen für die Evaluierung der Leistungsfähigkeit von Klassifikatoren [75]. Sie errechnet sich aus dem Verhältnis aller korrekten Vorhersagen des Klassifikators zu der gesamten Anzahl von untersuchten Instanzen [75] (Glg. 2.30, Quelle: [75]).

$$acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.30}$$

- **Precision (p)**
  Die Precision (p) misst den Anteil der korrekt als positiv klassifizierten Objekte im Verhältnis zur Gesamtzahl der positiv klassifizierten Instanzen [75] (Glg. 2.31, Quelle: [75]).

$$p = \frac{TP}{TP + FP} \tag{2.31}$$

- **Recall (r)**
  Der Recall (r) gibt das Verhältnis von korrekt positiv klassifizierten Objekten zu allen tatsächlich positiven Objekten an [77] (Glg. 2.32, Quelle: [77]).

$$r = \frac{TP}{TP + FN} \tag{2.32}$$

- **F1-Score (f1)**
  Der F1-Score (f1) repräsentiert das harmonische Mittel zwischen (p) und (r) [75] (Glg. 2.33, Quelle: [75]).

$$f1 = \frac{2 * p * r}{p + r} \tag{2.33}$$

2.6.2 Kreuzvalidierung

Ein weiteres Werkzeug für die Evaluierung der Leistungsfähigkeit eines Klassifikators ist die Kreuzvalidierung. Grundlage dafür ist die Aufteilung des verwendeten Datensatzes in ein Trainings- und ein Testset. Das Trainingsset wird für das Training des Klassifikationsmodells, und das Testset für die Evaluierung der Klassifikationsergebnisse herangezogen. Nach Giannakopoulos et al. [13] gibt es folgende Arten der Aufteilung:

- **Resubstitution**
  Bei dieser Methode wird das gesamte Datenset sowohl für das Training als auch für das Testen verwendet. Dabei besteht das Problem, dass die Performance des Klassifikators für unbekannte Daten nicht festgestellt werden kann; weshalb diese Methode nicht empfohlen wird [13].

- **Hold out**
  Hierbei wird das Datenset in zwei nicht überlappende Untergruppen - eine für das Training und eine für das Testen - zerlegt. Eine passende Aufteilung zu finden, ist dabei der schwierigste Punkt [13].

- **Repeated Hold out**
  Um die Schwachstellen der „Hold out"-Methode in den Griff zu bekommen, wird sie k-mal - immer mit einer anderen zufälligen Aufteilung der Datenmenge - wiederholt. Der Durchschnitt aller k Ergebnisse wird für die Berechnung der Performance herangezogen [13].

- **K-fold**
  Die „K-fold"-Methode basiert auf der zufälligen Aufteilung des Datensets in k nicht überlappende Untergruppen etwa gleicher Größe. Der Klassifikator wird k-mal trainiert und getestet, wobei für das Training immer $k - 1$ Untergruppen und für das Testen die verbleibende Untergruppe verwendet werden [13].

- **Repeated K-fold**
  Für diese Form der Kreuzvalidierung wird die „K-fold"-Methode mehrmals, immer mit einer anderen Unteraufteilung des Datensets, durchgeführt [13].

- **Leave-one-out**
  Bei dieser Methode handelt es sich um eine Variante der „K-fold"-Methode, bei der k gleich der Menge der Samples im Datensatz ist. Das heißt, dass für das Training in jedem Durchlauf alle Samples bis auf eines verwendet werden. Obwohl diese

Art der Kreuzvalidierung sehr zuverlässige Ergebnisse liefert, hat sie den großen Nachteil, dass sie in der Berechnung sehr aufwändig ist [13].

2.7 Verwandte Arbeiten

Zum Zeitpunkt der Arbeit ist nach Recherche der Autorin keine Untersuchung bekannt, die sich konkret mit dem Thema der inhaltsbasierten Videosuche für Parlamentssitzungen beschäftigt. Die Anwendungsgebiete solcher inhaltsbasierten Suchmaschinen sind vielfältig. So werden sie etwa für die Analyse von Nachrichten- und Vortragsvideos oder die Auswertung von Fernsehserien eingesetzt. Nachdem für den geplanten Ansatz Konzepte aus der Bild-, Audio- und Emotionsanalyse zum Einsatz kommen, ist die Studie von Arbeiten aus allen diesen Bereichen zielführend. Aufgrund des Aufbaus der Parlamentssitzungsvideos und der speziellen Art von relevanten Ereignissen in Parlamentssitzungen kann allerdings keiner der propagierten Ansätze direkt übernommen werden. Viele Untersuchungen sind auch nicht auf der Verwendung von multimodalen Daten aufgebaut [78]. Nachfolgend soll ein kleiner Überblick über interessante Arbeiten gegeben werden.

Nagaraja et al. [79] stellen eine inhaltsbasierte Videosuchmaschine vor, die Videos bestimmten Kategorien zuordnet und basierend auf einem Ähnlichkeitsabgleich dem Inputvideo ähnliche Videos anzeigt. Ihr Ansatz basiert auf der Extraktion von SIFT-, Farb- und Bewegungsmerkmalen und einer Klassifizierung mit SVM. Audiomerkmale werden nicht berücksichtigt. Mit ihrem System erzielen Nagaraja et al. [79] eine Genauigkeit von 75%.

Sivic et al. [80] untersuchen in ihrer Arbeit die automatisierte Gesichtserkennung und -verifizierung für Videos der Fernsehserie „Buffy the Vampire Slayer“. Sie verwenden dafür Frontal- und Profilgesichtsbilder und einen multimodalen Ansatz, bei dem zusätzlich Textinformationen aus Transkripten und Untertiteln für die Kennzeichnung der getrackten Gesichter benutzt werden. Für das Gesichtstracking wird der „Kanade-Lucas-Tomasi-Tracker“ eingesetzt. Die Klassifikation mit SVM gewährleistet die Unterscheidung der Gesichtstracks unterschiedlicher Personen und ist auf einer Linearkombination von Basiskernels (unterschiedliche Kernels für unterschiedliche HOG-Merkmale) aufgebaut. „Multiple Kernel Learning“ wird eingesetzt, um die beste Kombination von Merkmalen für die Unterscheidung der Charaktere festzulegen. Für die automatisierte Gesichtverifikation und die damit verbundene Kennzeichnung der Gesichter erreichen Sivic et al. [80] mit den getesteten Methoden eine Precision zwischen 62% und 90%.

Berrani et al. [81] präsentieren in ihrer Arbeit ein automatisiertes System, mit dem wiederkehrende Szenen wie z. B. Werbung oder Jingles in TV-Sendungen entdeckt werden können. Sie arbeiten mit einem Ansatz, der auf die Verarbeitung von visuellen Merkmalen konzentriert ist, testen aber auch eine Methode mit akustischen Merkmalen. Visuelle Deskriptoren werden vor der Shot-Erkennung für jeden Frame und zusätzlich nach der Shot-Erkennung für jedes Schlüsselbild auf Basis von DCT-Koeffizienten extrahiert. Für die Ähnlichkeitsmessung wird einerseits die Hamming-Distanz und andererseits die L_2-Distanz verwendet. Die von Berrani et al. [81] eingesetzte Klassifizierungsmethode ist

ein unüberwachtes Micro-Clustering-Verfahren. Für die Audiomerkmale wird das Signal mittels DFT in den Frequenzbereich transferiert und die Energie von 5 Frequenzbändern im Bereich von 300 bis 3.000 Hz berechnet. Die besten Ergebnisse mit der höchsten Clusteranzahl (7419), die im visuellen Bereich erzielt werden konnten, sind ein Recall von 98% und eine Precision von 80,6%. Zu den Tests mit Audiomerkmalen gibt es leider keine Aussage in [81].

Poria et al. [78] stellen ein Echtzeitsystem für die multimodale Analyse von Emotionen vor. Verwendet werden visuelle, akustische und textuelle Merkmale. Es gibt in dieser Arbeit allerdings nur 3 Emotionsklassen: „positive Emotion“, „negative Emotion“ und „Neutral“. Für die Extraktion von Gesichtsmerkmalen (charakteristischen Gesichtspunkten) verwenden Poria et al. die Softwarepakete „Luxand FSDK 1.7“ und „GAVAM“. Auch für die Merkmalsfindung im Audiobereich wird eine Software, nämlich „OpenEAR“, verwendet. Zum Einsatz kommen unter anderem MFCCs, „Spectral Centroid“, „Spectral Flux“ und „Beat Histogram“ (= Autokorrelation des RMS). Der Text wird mittels Konzept-Extraktion analysiert. Poria et al. [78] testen zwei Ansätze der multimodalen Datenfusion - auf der einen Seite die Fusion in einen einzigen Datenvektor vor der Klassifizierung, und auf der anderen Seite die Fusion der Klassifizierungsergebnisse der einzelnen Merkmalsgruppen. Für die Klassifizierung werden ein „Extreme Learning Machine-Klassifikator“, SVM oder ein „Naiver Bayes-Klassifikator“ eingesetzt. Mit den diversen Settings erzielen sie eine Precision zwischen 61,9% und 78,2%, wobei die besten Ergebnisse unter Verwendung der Merkmalsfusion aus allen drei Bereichen erzielt werden.

Yang et al. [61] setzen für die Indexierung und automatische Transkription von Universitätsvorlesungsvideos in Deutsch eine automatische Spracherkennung ein. Sie halten fest, dass Deutsch aufgrund des Sprachaufbaus wesentlich schwerer für die Spracherkennung zu verarbeiten ist als Englisch. Außerdem bereiten etwa Vorlesungsvideos aus technischen Bereichen mehr Schwierigkeiten als z. B. TV-Nachrichten, da in ihnen sehr viele Spezialausdrücke verwendet werden [61]. Eine der Kernaussagen ihrer Arbeit ist die Feststellung, dass die kontinuierliche Erweiterung des auf die jeweilige Domäne angepassten Sprachkorpus die Erkennungsrate wesentlich beeinflusst. Für die Umsetzung verwenden sie das Open-Source-Toolkit „Sphinx“ und erzielen unter Einsatz des „Voxforge“-Sprachkorpus, der um einen eigens trainierten Korpus von 1,6 Stunden erweitert wurde, eine Word-Error-Rate von 70,1%.

KAPITEL 3

Implementierung

In diesem Kapitel werden die einzelnen Schritte und die dabei angewandten Methoden zur Realisierung des Prototyps der inhaltsbasierten Suchmaschine für Parlamentssitzungen erläutert. Eine ausführliche Erklärung der technischen Grundlagen erfolgte bereits in Kapitel 2. Weiters werden die für das Training und die statistische Auswertung verwendeten Ground-Truth-Daten genauer beleuchtet. Details der statistischen Analyse werden in Kapitel 4 thematisiert.

3.1 Methodische Vorgehensweise

Der folgende Abschnitt widmet sich der Beschreibung aller Teilschritte, die zur Erreichung des Zieles - der Entwicklung eines funktionsfähigen Prototyps - durchgeführt werden müssen, angefangen von der Beschreibung der Anforderungen an das geplante System, dem Aufbau, bis zu den technischen Details der Segmentierung, Merkmalsextraktion und Klassifizierung.

3.1.1 Systemspezifikation

Der geplante Prototyp soll Benutzerinnen und Benutzern die Funktionalität bieten, Videomitschnitte von Parlamentssitzungen gezielt nach wichtigen Schlüsselszenen oder nach bestimmten Personen durchsuchen zu können. Als wichtige Schlüsselszenen werden Szenen definiert, die von der Grundstimmung der Rede abweichen und den Audioereignissen „Ordnungsruf“, „Klatschen“, Zwischenruf“ oder „Lachen“ zugeordnet werden können. Die Auswahl der Schlüsselszenen soll mittels Dropdown-Menü umgesetzt werden. Die Suche nach Personen soll durch Anklicken des gewünschten Gesichts in einer Bildergalerie oder mittels Namensauswahl realisiert werden. Eine kombinierte Suche nach bestimmten Abgeordneten in ausgewählten Schlüsselszenen soll ebenfalls enthalten sein. Wird hingegen nur nach einer der Audioereignisklassen gezielt gesucht, sollen im Suchergebnis

auch die diesem Audioereignis zugeordneten Personen enthalten sein. Zusätzlich soll eine Emotionserkennung aus der Gesichtsmimik implementiert werden, deren Ergebnis gemeinsam mit den Ergebnissen der Personensuche bzw. der Suche nach Audioereignissen dargestellt werden soll.

Aufgrund der großen Datenmenge, die durch Videomitschnitte von Parlamentssitzungen, die bis zu 10 Stunden dauern können, anfällt, liegt der Schwerpunkt in der Umsetzung darauf, einen inhaltsbasierten Ansatz für die Klassifizierung zu verwenden, um eine aufwändige manuelle Indexierung zu vermeiden. Der zweite zentrale Punkt ist eine multimodale Herangehensweise, bei der sowohl Audio- als auch Bildmerkmale zum Einsatz kommen. Ziel ist es festzustellen, ob mit dem gewählten inhaltsbasierten, multimodalen Ansatz die gewünschte Funktionalität erreicht werden kann. Das beinhaltet auch die Analyse, ob die definierten Ereignisklassen sinnvoll für die geplante Aufgabe sind, und ob passende Merkmalskombinationen und Klassifizierungsmethoden gefunden werden können.

In den nachfolgenden Kapiteln wird ein Überblick über die technische Umsetzung gegeben.

3.1.2 Aufbau des Prototyps

Die wesentlichen Schritte der Implementierung können in folgende drei Bereiche unterteilt werden: Aufbau der Trainingsdatenbanken, Training und Klassifizierung. Trainingsdatenbanken werden für das Training der Modelle in den Bereichen Gesichtsverifizierung, Emotions- und Audioereigniserkennung benötigt. Zusätzlich wird eine Ground-Truth-Videodatenbank mit Ausschnitten von Parlamentssitzungen für das Modelltraining, aber hauptsächlich für die Evaluierung der verwendeten Methoden angelegt.

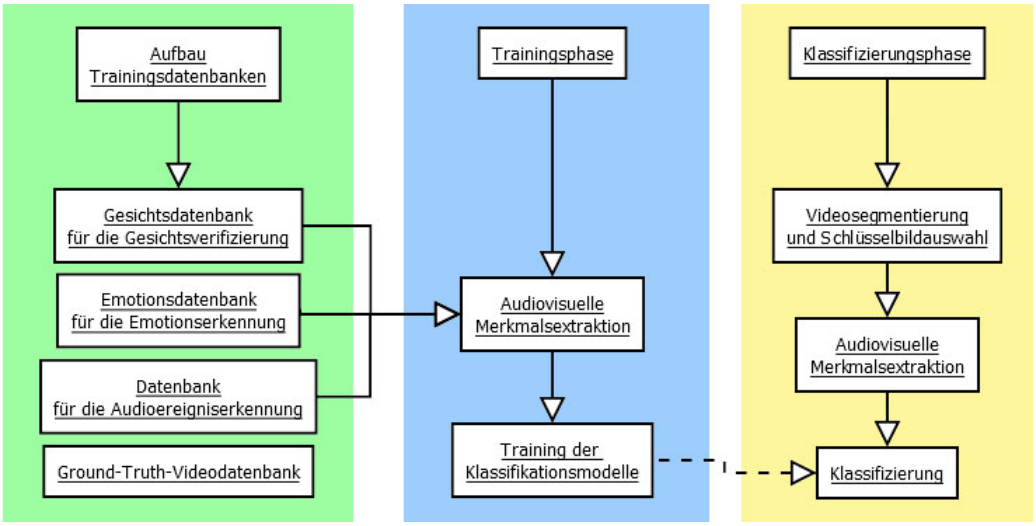

Abbildung 3.1: Implementierungsschritte für den Prototyp

Die Trainingsphase besteht aus den Teilschritten Merkmalsextraktion und Training der Klassifikatoren. Es werden sowohl akustische als auch visuelle Merkmale aus den Daten der Trainingsdatenbanken als Grundlage für das Training der jeweiligen Modelle extrahiert. Diese Modelle sollen während der Klassifizierung in der Lage sein, unbekannte Videosegmente korrekt den jeweiligen Klassen zuzuordnen.

Die Klassifizierungsphase selbst lässt sich in die Punkte zeitliche Videosegmentierung und Schlüsselbildauswahl, Merkmalsextraktion und Klassifizierung unterteilen. Zur Datenreduktion werden die Videos während der Videosegmentierung in einzelne Shots unterteilt und aus relevanten Shots werden repräsentative Schlüsselbilder für die visuelle Klassifizierung ausgewählt. Die Audiospuren der Shots werden für die Merkmalsextraktion in „short-time-“ und „mid-time-Frames“ segmentiert. Für die Klassifizierung werden aus diesen Audioframes akustische Merkmale und aus den Schlüsselbildern visuelle Merkmale mit denselben Parametereinstellungen wie beim Training extrahiert. Diese Merkmalsvektoren bilden die Grundlage für die Zuordnung zu den jeweiligen Klassen mit Hilfe der zuvor trainierten Modelle. Abbildung 3.1 gibt einen Überblick der nötigen Schritte.

3.1.3 Ground-Truth-Daten für Training und Evaluierung

In einem ersten Schritt werden Videos von Nationalratssitzungen des österreichischen Parlaments für Training und Evaluierung gespeichert. Da die Mitschnitte der Sitzungen nicht auf der Homepage des Parlaments (http://www.parlament.gv.at/) zur Verfügung gestellt werden, müssen sie direkt beim ORF oder via Youtube bezogen werden. Bei aktuellen Sitzungen besteht die Möglichkeit, den Mitschnitt bis zu einer Woche nach der Sitzung mit Hilfe eines Grabbers (z.B. MediathekView) von der Homepage des ORF (http://tvthek.orf.at/) herunterzuladen. Aufgrund der ressourcenintensiven Berechnungen – hier erweist sich Matlab bekanntermaßen als langsam – werden für die exemplarische Auswertung in dieser Arbeit keine Videos kompletter Parlamentssitzung von mehreren Stunden verwendet. Stattdessen werden drei kurze Videoausschnitte über die Videoplattform Youtube ausgewählt.

	Länge	Schnitte		Ordnungsruf		Klatschen		Zwischenruf		Lachen	
		H	Ü	E	F	E	F	E	F	E	F
Video 1	05‘17	29	0	2	2	6	27	3	7	0	0
Video 2	08‘13	44	0	11	11	12	151	5	76	0	0
Video 3	05‘40	49	9	25	25	9	64	4	64	2	12

Tabelle 3.1: Überblick Ground-Truth-Daten: Harte Schnitte (H), Überblendungen (Ü), Anzahl der Audio-Ereignisse (E), Anzahl der Frames, die den Audio-Ereignissen zugeordnet werden (F)

Die Auswahl erfolgt nach Gesichtspunkten der im Zentrum stehenden, geplanten Audio-Analyse, d.h. die Videos sollen Sequenzen der vier gesuchten Audio-Klassen „Ordnungsruf“, „Klatschen“, „Zwischenruf“ und „Lachen“ enthalten. Es zeigt sich allerdings, dass die

Klasse „Lachen“ die geringste Klassenhäufigkeit aufweist, und es daher schwierig ist passende Frames zu finden (Tabelle 3.1). Die Videos liegen als .mp4 in unterschiedlicher Auflösung (maximal 848 x 480) mit einer Framerate von 25 Bildern/Sekunde und der Videocodierung H264/AVC vor. Die Stereo-Audiospuren haben eine Abtastrate von 44,1 kHz und sind AAC codiert.

Zusätzlich wird aus den auf der Homepage des Parlaments verfügbaren Portrait-Bildern der in den Ground-Truth-Videos vorkommenden Abgeordneten eine Gesichtsdatenbank für das Training und Testen der späteren Gesichtsverifizierung angelegt. Pro Person werden 3 bis 10 Gesichtsbilder (Frontal- bzw. Profilansichten) gespeichert. Die Portrait-Bilder werden in größtmöglicher Auflösung (ca. 2125 x 1416 Pixel) verwendet. Die mit Hilfe des Viola-Jones-Algorithmus [32] daraus generierten Gesichtsbilder haben eine Größe von 75 x 75 Pixeln (Abbildung 3.2).

Abbildung 3.2: Originalbild und extrahiertes Gesichtsbild, Originalfoto: Parlamentsdirektion | Mike Ranz

Es gibt einige Datenbanken, die für das Training und Testen von Emotionserkennungssystemen eingesetzt werden können. Grundsätzlich besteht hierbei das Problem, dass viele der verfügbaren Datenbanken nur Bilder bzw. Videos mit gestellten Emotionen enthalten [82]. Wenn man diese für das Training von Emotionserkennungssystemen benutzt, die für die Erkennung spontaner Emotionen verwendet werden sollen, kann das zu einer beachtlichen Verschlechterung der Erkennungsrate führen. Bartlett et al. [82] haben dieses Phänomen in ihrer Arbeit näher untersucht, indem sie für das Training ihres Emotionserkennungssystems die Cohn-Kanade- und die Ekman-Hager-Datenbank eingesetzt haben. Diese beiden Datenbanken enthalten nur gestellte Emotionen. Angewandt auf eine Datenbank mit spontanen Emotionsbildern (RU-FACS) sank die Erkennungsrate. Bartlett et al. [82] führen das auf unkontrollierte Kopfbewegungen, Verwendung anderer Gesichtsmuskeln und Vorhandensein von Sprache bei spontanen Emotionen zurück und empfehlen in

diesem Fall, für das Training Bilder mit spontaner Emotion zu verwenden. Die in diesem Forschungsbereich sehr oft verwendete Cohn-Kanade-Datenbank hat nach Whitehill et al. [39] neben dem Fehlen von spontaner Emotion noch zwei weitere Nachteile. Erstens enden die Aufnahmen beim „Apex“ der gezeigten Emotion, d.h. mittendrin (Onset-Apex-Offset, siehe Kapitel 2.3.4), und zweitens ist bei vielen Aufnahmen der Datums- bzw. Zeitstempel über dem Kinn platziert, was das Tracken des Kinns schwieriger macht.

Nachdem es im Fall der Emotionserkennung in Parlamentsvideos um spontane Emotionen geht, fällt die Wahl bei der Trainingsdatenbank auf die MMI-Datenbank, welche sowohl gestellte als auch spontane Emotionsbilder und -videos enthält [83]. Ein weiterer Vorteil neben der freien Verfügbarkeit für Forscher – für die Nutzung ist eine Online-Registrierung nötig – ist die Tatsache, dass neben Frontalbildern auch Profilbilder enthalten sind. Außerdem wird ein Suchinterface angeboten, mit dem die vorhandenen Daten nach verschiedenen Eignungskriterien durchsucht werden können. Für die geplante Emotionserkennung werden 204 Videos ausgewählt, die mit einer Zuordnung zu einer der sechs Grundemotionen Trauer, Zorn, Überraschung, Angst, Ekel und Freude versehen sind (Tabelle 3.2). Diese Klassenzuordnungen können aus den mitgelieferten .xml-Dateien ausgelesen werden. Die Videos haben eine durchschnittliche Länge von 2 bis 3 Sekunden. Für die geplante Emotionserkennung sind die „Apex-Frames“ interessant, deshalb werden jeweils 4-6 Frames aus diesem Zeitfenster in der Mitte des Videos extrahiert. Danach kommt ebenfalls der Viola-Jones-Algorithmus [32] zum Einsatz, um frontale Gesichtsbilder für das Training des Klassifikators für die 6 Emotionsklassen zu extrahieren (Abbildung 3.3).

	1 Zorn	2 Ekel	3 Angst	4 Freude	5 Trauer	6 Überraschung
Anzahl MMI-Videos	33	30	28	38	32	43

Tabelle 3.2: Zuordnung der Videos aus der MMI-Datenbank zu 6 Grundemotionen

Abbildung 3.3: Aus den Videos der MMI-Datenbank extrahierte Gesichtsbilder der 6 Emotionsklassen Zorn, Ekel, Angst, Freude, Trauer, Überraschung

Für die Erkennung der Audioklasse „Ordnungsruf“ wird eine Trainingsdatenbank angelegt, die 34 Aufnahmen des Wortes „Ordnungsruf“, gesprochen von 5 verschiedenen Personen (2 weiblich, 3 männlich), enthält.

3.1.4 Zeitliche Videosegmentierung und Schlüsselbild-Auswahl

Prinzipiell können Videos von österreichischen Parlamentssitzungen grob in 9 Shotkategorien unterteilt werden (Abbildung 3.4):

1. Ministerin/Minister, Nahaufnahme
2. Abgeordnete/Abgeordneter am Rednerpult, Nahaufnahme
3. Abgeordnete/Abgeordneter an seinem Platz, Nahaufnahme
4. 2 Abgeordnete an ihrem Platz, Nahaufnahme
5. Vorsitzende/Vorsitzender, Nahaufnahme
6. Abgeordnete/Abgeordneter am Rednerpult plus Ministerin/Minister
7. Ministerin/Minister plus Vorsitzende/Vorsitzender
8. Plenum
9. Zuschauer

Abbildung 3.4: Neun Shotkategorien österreichischer Parlamentssitzungen (von links oben nach rechts unten)

Für die Gesichtserkennung und -verifizierung sind Nahaufnahmen - speziell die Rednerpult-Shots - am interessantesten, da die Gesichtsbilder für die Verarbeitung nicht zu klein sein dürfen. Plenum- und Zuschauershots werden aussortiert. Diverse Experimente haben gezeigt, dass Histogramm-Merkmale trotz ihrer Einfachheit zu guten Ergebnissen in der

Shoterkennung führen [26]. Da die Histogramme der Frames im vorliegenden Videomaterial keine großen Differenzen aufweisen, und es sich bei den eingesetzten Schnitten hauptsächlich um harte Schnitte handelt, wird die Videosegmentierung unter Verwendung der in Punkt 2.2.1 vorgestellten ECR-Methode realisiert. Für die Entscheidungsfunktion wird eine adaptive Threshold-Methode verwendet, die nach der von Dugad et al. in [84] vorgestellten Variante, basierend auf Mittelwert, Standardabweichung und Threshold $t = 3$, implementiert wird. Das eingesetzte Fenster hat eine Größe von 5 Frames, wobei der mittlere Frame als Shot-Grenze erkannt wird, wenn folgende Bedingungen erfüllt sind [84] [85]:

1. ECR(Mittel-Frame) = max(ECR) innerhalb des Fensters
2. ECR(Mittel-Frame) $> max(\mu_{ECRleft} + t\sqrt{\sigma_{ECRleft}}, \mu_{ECRright} + t\sqrt{\sigma_{ECRright}})$,

Die Audiosegmentierung erfolgt auf den beiden Ebenen „mid-time“ und „short-time“. Merkmale, die aus dem gesamten Audiosignal (dies entspricht dem „long-time-Segment“) berechnet werden, kommen in der vorliegenden Arbeit nicht zum Einsatz. Die Länge der „mid-time-Segmente“ beträgt 0.8 Sekunden. Auf dieser Ebene wird eine Überlappung der Segmente von 50% verwendet. Die „mid-time-Segmente“ werden in „short-time-Segmente“ mit einer Länge von 30 Millisekunden (das entspricht 1323 Samples bei einer Abtastrate von 44,1 kHz) und einer Überlappung von 50% unterteilt (Abbildung 3.5).

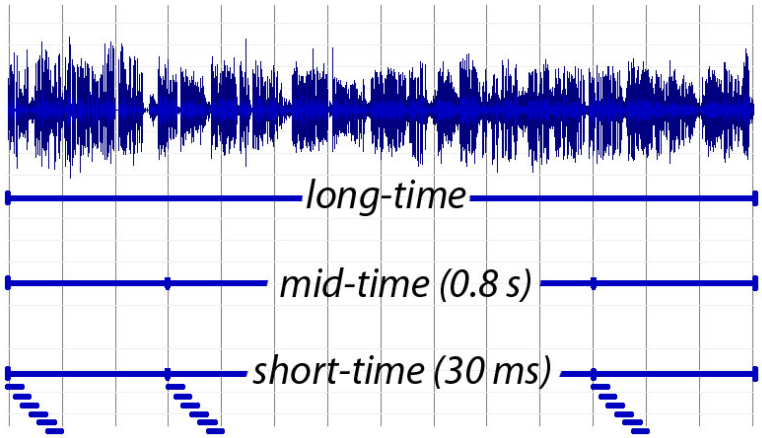

Abbildung 3.5: long-time-, mid-time- und short-time-Audiosegmentierung

Nach der Shoterkennung werden aus jeder einzelnen Sequenz Schlüsselbilder gewählt, die den jeweiligen Shot repräsentieren. Die eingesetzte Methode zur Auswahl der Schlüsselbilder ist nach der Einteilung von Azra und Shobha [29], die in Kapitel 2.2.2 näher erklärt wird, den eventbasierten Algorithmen zuzuordnen. Es werden pro Shot jene Frames gewählt, die ein relevantes Audioereignis beinhalten. Für Shots, in denen kein

relevantes Audioereignis präsent ist, werden der erste, der mittlere und der letzte Frame als Schlüsselbilder festgelegt. Dies dient dazu, bei der Gesichtserkennung, -verifizierung und Emotionserkennung mehr Daten für die Auswertung zur Verfügung zu haben. Da es innerhalb der Shots kaum visuelle Unterschiede zwischen den Frames gibt, wäre ansonsten 1 Frame pro Shot ausreichend. Auf eine Gruppierung zu Szenen kann im vorliegenden Fall verzichtet werden, da die Shots die größte zusammenhängende Einheit bilden.

3.1.5 Visuelle Merkmale, Cluster und Training

Für die Gesichtsverifizierung werden die von Dalal und Triggs [15] propagierten und in Kapitel 2.3.1 näher erklärten HOG-Merkmale eingesetzt. Diese werden für jedes der 232 aus den Portrait-Bildern der Abgeordneten mittels Viola-Jones-Algorithmus [32] generierten Gesichtsbilder extrahiert. Für die Berechnung wird eine Zellengröße von 8 x 8 Pixeln und eine Blockgröße von 2 x 2 Zellen mit 50% Überlappung gewählt, wie Dalal und Triggs es empfehlen [15]. Abweichend davon werden für die Zuordnung der Werte der Gradientenorientierung von 0 bis 180 Grad statt der propagierten 9 Histogramm-Bins 20 verwendet. Abhängig von den gewählten Parametern und der Bildgröße ergibt das pro Gesichtsbild einen 1 x 5120 großen HOG-Merkmalsvektor. Da 85% der Gesichtsbilder für das Training verwendet werden, hat die HOG-Trainingsmatrix eine Größe von 198 x 5120. Diese bildet die Grundlage für die anschließende Klassifizierung.

In einem zweiten Setting wird getestet, inwieweit sich die Verifizierungsrate verbessern lässt, wenn die HOG-Merkmale nicht aus dem gesamten Gesichtsbild berechnet werden, sondern die 10 wichtigsten „SURF-Regions of Interest“ (siehe auch Kapitel 2.3.2) als Grundlage für die Berechnung dienen. Hierbei hat die HOG-Trainingsmatrix eine Größe von 1693 x 80, weil nicht in jedem Gesichtsbild 10 „SURF-Regions of Interest“ vorhanden sind.

Da es sich im vorliegenden Fall der Gesichtsverifizierung um ein Multi-Klassen-Klassifikationsproblem handelt, kommt im nächsten Schritt für die Generierung der Klassenmodelle die One-Against-One-SVM-Methode zum Einsatz. Für die zu unterscheidenden 39 Klassen werden $\frac{39\times(39-1)}{2}$ binäre SVMs trainiert. Eine nähere Erklärung zu den SVMs, die zu den überwachten Klassifizierungsmethoden gehören, findet sich in Kapitel 2.5.3. Zusätzlich wird für die Kombination der binären Klassifikatoren das Konzept der „Error-correcting output codes (ECOC)“ verwendet. Diese Methode basiert auf der Erstellung einer Coding-Matrix, welche später die Grundlage für die Klassifizierung unbekannter Gesichter bildet. Dabei repräsentiert jede Spalte der Matrix eine der binären One-Against-One-SVMs. Die verwendete positive Trainingsklasse ist in jeder Spalte mit 1 und die negative mit -1 markiert. Alle anderen Klassen werden vom Training der jeweiligen SVM ausgeschlossen und mit 0 markiert [86] (Tabelle 3.3). Durch dieses Vorgehen wird in jeder Zeile der Matrix ein Codewort generiert, welches eine Klasse repräsentiert. Bei Verwendung von One-Against-One-SVMs ist das eine Kombination aus 0, 1 und -1. Diese Codewörter werden später für die Klassenzuordnung unbekannter Gesichter verwendet.

Die Grundlage für die Emotionserkennung bilden die aus den Videos der MMI-Datenbank extrahierten Gesichtsbilder der 4 Grundemotionen Zorn, Freude, Trauer und Überraschung. Aus diesen Bildern werden SURF-Merkmale (siehe Kapitel 2.3.2) generiert. Um die dafür notwendigen „Points of interest“ zu definieren, wird ein Gitter mit einer Schrittweite von 8 x 8 Pixeln verwendet. Die Blockweiten haben eine Größe von 32 x 32, 64 x 64, 96 x 96 und 128 x 128 Pixeln. 80 % der repräsentativsten Merkmale werden für die Gruppierung zu Merkmalsgruppen (Clustern) mit Hilfe des k-Means-Clusteralgorithmus eingesetzt. Das so generierte Vokabular für die eingesetzte BoVW-Methode besteht aus 1000 Clustern (siehe Kapitel 2.5.2). Für das anschließende Training des Klassenmodells zur Unterscheidung der Emotionsklassen werden wie bei der Gesichtsverifizierung One-Against-One-SVMs verwendet. Für die zu unterscheidenden 4 Klassen werden $\frac{4\times(4-1)}{2}$ binäre SVMs trainiert. Die Kombination der binären Klassifikatoren erfolgt auch hier mit Hilfe der „ECOC-Methode“. Die Klassen „Ekel“ und „Angst“ werden nicht integriert, da sie in den Parlamentssitzungsmitschnitten nicht vorkommen.

	SVM 1	SVM 2	SVM 3	SVM 4	SVM 5	SVM 6
Klasse 1	1	1	1	0	0	0
Klasse 2	-1	0	0	1	1	0
Klasse 3	0	-1	0	-1	0	1
Klasse 4	0	0	-1	0	-1	-1

Tabelle 3.3: Coding-Matrix für 4 Klassen unter Verwendung von 6 One-Against-One-SVMs

3.1.6 Auditive Merkmale, Cluster und Training

Für die Berechnung der Audiomerkmale wird nur ein Kanal des Stereosignals verwendet. Zuerst werden die Signalwerte auf einen Wertebereich zwischen 0 und 1 normalisiert. Die segmentierten „short-time-Frames“ werden durch den Einsatz eines Hamming-Fensters (siehe Kapitel 2.4) an den Rändern geglättet, und damit störende Randeffekte vermieden. Im Zentrum der Audioanalyse steht die Erkennung von Audioereignissen. Konkret geht es um Szenen, die einen höheren Lärmpegel als der Durchschnitt aufweisen. Diese sollen einer der 4 Audioklassen „Ordnungsruf“, „Klatschen“, „Zwischenruf“ und „Lachen“ zuordenbar sein.

Audioklasse „Ordnungsruf“

Aufbauend auf den in Kapitel 2.5.4 ausgeführten Überlegungen von Vuegen et al. [23] und Pohjalainen et al. [68] zur Erkennung von Audioereignissen wird die Erkennung der Audioklasse „Ordnungsruf“ unter Verwendung von MFCC-Merkmalen in Kombination mit einem GMM realisiert. In einem Vorverarbeitungsschritt werden die 34 Trainingsfiles mit Hilfe eines Amplitudenschwellwertes t ohne die Anfangs- und Endframes, die Stille enthalten, für die weitere Verarbeitung gespeichert. Da die Amplitudenwerte vor der Normalisierung zwischen -1 und 1 liegen, hat sich dafür der Schwellwert $t = 0.01$ bewährt. Die Trainingsfiles werden nach der Normalisierung in „short-time-Frames“ mit einer

Länge von 30 Millisekunden und einer Überlappung von 50% unterteilt. Die Berechnung der MFCC-Merkmale erfolgt aus diesen mittels Hamming-Fenster an den Rändern geglätteten „short-time-Frames“. Es wurden Versuche mit verschiedenen Merkmalsparametern durchgeführt. Tabelle 3.4 gibt einen Überblick.

	MFCC Koeff.	0. Koeff.	Δ Koeff.	ΔΔ Koeff.	Dimension gesamt	Filter
Versuch 1	13	Ja	Ja	Ja	42	26
Versuch 2	13	Nein	Ja	Ja	39	35
Versuch 3	13	Nein	Ja	Nein	26	35
Versuch 4	13	Ja	Ja	Nein	28	35

Tabelle 3.4: MFCC-Merkmale, Berechnung mit verschiedenen Parametern

Die unterschiedlichen MFCC-Merkmalsvektoren werden für das Training des GMMs unter Verwendung der Matlab-Funktion „fitgmdist“ verwendet. Um festzustellen, welche Parametereinstellungen beim Training des GMMs zur effektivsten Abbildung der Klasse „Ordnungsruf“ führen, werden Versuche mit unterschiedlicher Anzahl von Gaußverteilungen, voller bzw. diagonaler Kovarianzmatrix, unterschiedlicher Anzahl der Wiederholungen des EM-Algorithmus, unterschiedlichem Regularisierungswert und teilweiser Festlegung der Startwerte durch Einsatz des k-Means-Algorithmus durchgeführt. In Tabelle 3.5 werden die Grundwerte der aussagekräftigsten Versuche und die damit erzielten Ergebnisse dargestellt. Da der EM-Algorithmus in Versuch 25 nicht konvergierte, wurde in Versuch 26 die Zahl der Wiederholungen des EM-Algorithmus auf 10 erhöht. Konvergenz konnte mit diesen Parametern trotzdem nicht erreicht werden. Um festzustellen, welches Modell die Klasse am besten abbildet, werden die Werte des Bayesschen Informationskriteriums (BIC) herangezogen. Je negativer der BIC, desto besser passt das Modell [87]. In dieser Versuchsreihe erfüllt das GMM aus Versuch 34 diese Bedingung und wird für alle weiteren Berechnungen eingesetzt. Die Anzahl der verwendeten Gaußverteilungen entspricht bei diesem Modell in etwa der Anzahl der Phoneme in der deutschen Sprache.

Neben dem Klassifizierungsansatz mittels GMM wird auch die Verwendung von DTW für die korrekte Klassenzuweisung von „Ordnungsrufen“ untersucht. Für diese Analyse wird die DTW-Implementierung von Quan Wang [88] verwendet. Die zuvor aus den Trainingsfiles extrahierten MFCC-Merkmalsvektoren dienen auch hier als Input und werden für die Berechnung der euklidischen Distanzen zwischen allen Elementen des Trainingssets herangezogen. Die berechneten Grundwerte (Tabelle 3.6) bilden die Grundlage für das Schwellwertverfahren, welches bei der Klassifizierung der Testdateien zur Anwendung kommt.

Audioklassen „Klatschen“ und „Lachen“

Vuegen et al. [23] setzen in ihrer Arbeit den 0. MFCC-Koeffizienten für eine Schwellwertoperation ein, um Ereignisse in Audioframes zu erkennen. Da sich die Klassen „Klatschen“ und „Lachen“ aufgrund einer höheren Grundfrequenz von den restlichen Klassen (inklusive Sprache) unterscheiden, wird dieser Ansatz als grundlegendes Ent-

scheidungskriterium übernommen. Die gesuchten Frames müssen einen deutlich höheren Wert des 0. MFCC-Koeffizienten aufweisen. Als Schwellwert dient der Mittelwert des 0. MFCC-Koeffizienten des „Ordnungsruf"-Trainingssets. Liegen 75% der „shorttime-Werte" des 0. MFCC-Koeffizienten innerhalb eines „midtime-Frames" von 0,8 Sekunden über dem Schwellwert, so wird dieser „midtime-Frame" als relevantes Ereignis klassifiziert und für die Merkmalsextraktion verwendet.

	MFCC Dim.	MFCC Filter	Gaussv.	Kovarianz	Regularisierung	EM Anz.	k-Means	konv.	BIC
V1	42	26	20	voll	0,001	1	nein	ja	37155
V2	39	35	20	voll	0,001	1	nein	ja	34836
V3	39	35	8	voll	0,001	1	nein	ja	-15156
V4	39	35	8	diagonal	0,001	1	nein	ja	-10068
V5	39	35	20	diagonal	0,001	1	nein	ja	-17984
V6	39	35	34	voll	0,001	1	nein	ja	101390
V7	26	35	34	voll	0,001	1	nein	ja	85107
V8	26	35	20	voll	0,001	1	nein	ja	59669
V9	28	35	20	voll	0,001	1	nein	ja	76220
V10	28	35	34	voll	0,001	1	nein	ja	105280
V25	26	35	34	voll	0,01	1	nein	nein	-
V26	26	35	34	voll	0,01	10	nein	nein	-
V27	26	35	34	voll	0,01	10	ja	ja	100100
V32	39	35	17	diagonal	0,001	10	nein	ja	-17124
V33	39	35	17	diagonal	0,001	10	ja	ja	-17257
V34	39	35	40	diagonal	0,001	1	ja	ja	-19439

Tabelle 3.5: Versuche mit unterschiedlichen Parametersettings für das Training des GMMs der Klasse „Ordnungsruf" und BIC-Werte als Entscheidungskriterium

Im Fall der Klasse „Klatschen" wird mit einer Kombination aus zeit- und frequenzabhängigen Merkmalen gearbeitet. Zum Einsatz kommen die „Varianz der STE", die „Varianz der ZCR", die „Spectral Entropy" und der „Spectral Flux". Für die Klassifizierung von „Lachen" haben sich die frequenzabhängigen Merkmale „Standardabweichung des Spectral Rolloff" und „Standardabweichung des Spectral Crest" bewährt. Beide Werte liegen bei „Lach-Frames" deutlich unter den Werten aller anderen enthaltenen Signalklassen. Ein binärer Entscheidungsbaum kommt für diese beiden Klassen in der Klassifizierungsphase zum Einsatz (siehe Kapitel 3.1.7).

	MFCC Dim.	MFCC Filter	Min. DTW	Max. DTW	Durchschnitt DTW
Versuch 1	42	26	122,2994	894,0165	393,2307
Versuch 2	39	35	119,4869	592,5698	297,5214
Versuch 3	26	35	118,8415	591,2067	296,5734
Versuch 4	28	35	143,9499	1043,0000	459,3034

Tabelle 3.6: Euklidische DTW-Distanzen des „Ordnungsruf"-Trainings mit MFCC Merkmalsvektoren

Audioklasse „Zwischenruf“

Im Gegensatz zu den Klassen „Klatschen“ und „Lachen“ kann bei der Klasse „Zwischenruf“ nicht auf den 0. MFCC-Koeffizienten als erstes Unterscheidungskriterium zurückgegriffen werden, da dieser innerhalb der Klasse sehr unterschiedlich ist. Zwischenrufe sind außerdem oft von genereller Unruhe, von Sprache aber auch von Klatschen überlagert. Ein Klassifizierungsversuch mit Hilfe eines binären Entscheidungsbaumes unter Verwendung der Merkmale „ZCR“, „STE“, „Spectral Rolloff“, „Grundfrequenz“ und „Standardabweichung der Grundfrequenz“ führte zu keinen zufriedenstellenden Ergebnissen (maximale Precision p ca. 13,6%).

Aufgrund dieser Erkenntnisse wird für die Klasse „Zwischenruf“ ebenfalls ein GMM-Ansatz getestet. In der Trainingsphase werden zwei GMMs trainiert. Eines mit den Merkmalen der gesuchten „Zwischenruf-Frames“, das zweite mit den Merkmalen der restlichen Frames des 2. Ground-Truth-Videos. Die beiden GMMs haben als fixe Parameter eine diagonale Kovarianzmatrix, 10 Wiederholungen des EM-Algorithmus und einen Regularisierungswert von 0,001. Die verwendeten Merkmalsvektoren bestehen aus verschiedenen Kombinationen von Mittelwert, Standardabweichung, Minimum und Maximum von 12 Zeit- und Frequenzmerkmalen (STE, ZCR, RMS, EE, Spectral Centroid und Spread, Spectral Entropy, Spectral Flux, Spectral Rolloff, Grundfrequenz und Harmonic Ratio, Spectral Crest) der „midtime-Frames“. Erweitert werden diese Merkmalsvektoren teilweise durch MFCC-Merkmale, für deren Berechnung generell 35 Filter verwendet werden. Tabelle 3.7 gibt einen Überblick der getesteten GMM-Varianten. Die Werte des BIC werden wie beim GMM der Klasse „Ordnungsruf“ für die Entscheidung, welche Modelle die Daten am besten abbilden, herangezogen. In dieser Testreihe sind das die beiden GMMs aus Versuch 4, welche für alle weiteren Berechnungen verwendet werden.

	Gaussv.	Anz. Z/F Merkmale	MFCC Dim.	0. MFCC, $\Delta, \Delta\Delta$	BIC GMM1	BIC GMM2
V1	8	23			-1019	-22380
V2	8	12			-390	-12016
V3	8	44			-792	-30402
V4	8	32	39	$\Delta, \Delta\Delta$	-7091	-144750
V5	20	32	39	$\Delta, \Delta\Delta$	-1727	-141430
V6	12	44			230	-31489
V7	12	44	13		1778	-10156
V8	8	44	13		594	-7819

Tabelle 3.7: Versuche mit unterschiedlichen Parametersettings für das Training der GMMs der Klasse „Zwischenruf“ und BIC-Werte als Entscheidungskriterium

3.1.7 Suchphase

Gesichtsverifizierung und Emotionserkennung

Die ausgewählten Schlüsselbilder sind die Grundlage für die Gesichtsverifizierung und die Emotionserkennung. Zuerst wird auf den Schlüsselbildern eine Gesichtserkennung unter Einsatz des Viola-Jones-Algorithmus [32] durchgeführt, der in Matlab mit vortrainierten Klassifikatoren für Frontal- und Profilaufnahmen, Mund, Nase, Augen und Oberkörper implementiert ist. Die geforderte Mindestgesichtsgröße beträgt 75 x 75 Pixel, um für die weitere Verarbeitung genügend Information zu beinhalten. Ein Nebenaspekt dieser Einschränkung ist die automatische Verwerfung von Shots des Plenums und der Zuschauer. Die gefundenen Gesichter werden durch Einsatz eines Mund- und Nasendetektors dahingehend überprüft, ob es sich tatsächlich um ein Gesicht handelt.

Für die Gesichtsverifizierung werden nun aus den Gesichtsbildern HOG-Merkmale mit denselben Parametern wie beim Training extrahiert - einmal aus dem gesamten Bild und einmal aus den 10 wichtigsten „SURF-Regions of Interest“. Die in der Trainingsphase für die Gesichtsverifizierung trainierten One-Against-One-SVMs werden nun in der Phase der Klassifizierung für die Merkmale des gesuchten Gesichts ausgewertet. Der so generierte Vektor wird mit den Codewörtern der erstellten „ECOC-Coding-Matrix“ verglichen. Das unbekannte Gesicht wird schließlich jener Klasse zugeordnet, deren Codewort die größte Ähnlichkeit mit dem Ergebnisvektor aufweist. Meist wird die Hamming-Distanz für die Ähnlichkeitsberechnung in der Decodierungsphase eingesetzt [86]. SVMs verwenden allerdings nach Provost et al. [89] die Hinge-Verlustfunktion, welche auch im vorliegenden Fall für die Berechnung der Strafterme benutzt wurde. Abbildung 3.6 zeigt den Hinge-Verlust für eine negative Instanz. Generell gilt: Je größer der Abstand zur Entscheidungsgrenze ist, desto größer wird auch der Strafterm, der lineares Wachstum aufweist [89]. Jene Klasse, die innerhalb des Shots am häufigsten erkannt wurde, wird als Ergebnis der Gesichtsverifikation angenommen.

Für die Emotionserkennung dienen ebenfalls die extrahierten Gesichtsbilder als Grundlage. Die Klassifizierung funktioniert im Prinzip wie bei der Gesichtsverifizierung mit dem Unterschied, dass dabei SURF-Merkmale und ein BoVW-Histogramm als Grundlage für die Auswertung der für die Emotionserkennung trainierten One-Against-One-SVMs verwendet werden.

Audioerkennung

Für die Audioerkennung werden MFCC-Merkmale mit denselben Parametern extrahiert, die beim Training zum Einsatz kamen (Tabelle 3.4). Als „mid-time-Framegröße“ wurden generell 0,8 Sekunden festgelegt, da das der durchschnittlichen Länge der Trainingsfiles im „Ordnungsruf“-Trainingsset entspricht. Die „short-time-Frames“ haben wie beim Training eine Länge von 30 Millisekunden. Für beide Framegrößen beträgt die verwendete Überlappung 50 %.

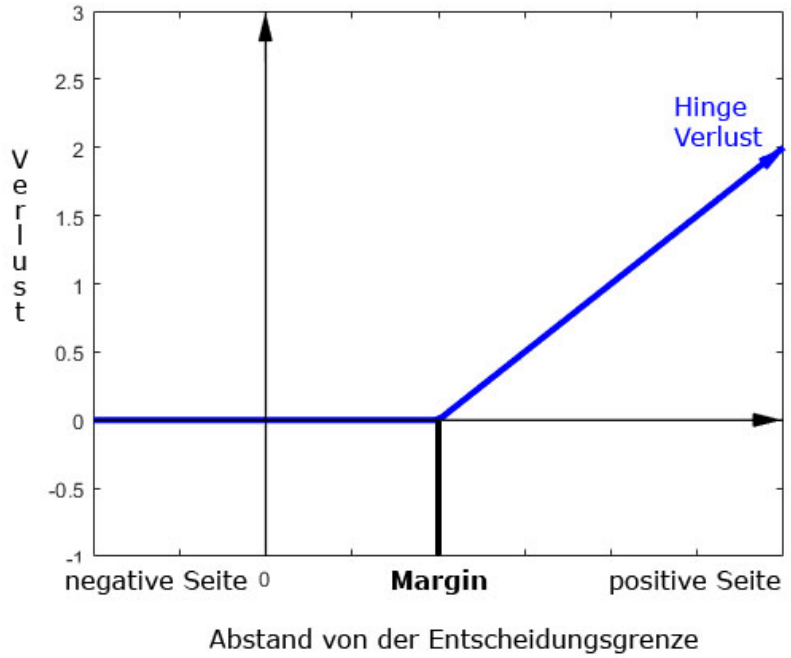

Abbildung 3.6: Darstellung der Hinge-Verlustfunktion für eine negative Instanz, Quelle: in Anlehnung an Provost et al. [89]

Audioklasse „Ordnungsruf“

Im Fall der Klassifizierung mittels DTW werden die MFCC-Merkmalsvektoren der „mid-time-Frames“ für die Berechnung der Ähnlichkeit zu jeder Instanz aus dem Trainingsset herangezogen. 2 aus den Trainingsdaten errechnete Schwellwerte bilden die Ober- und Untergrenze der Entscheidungsfunktion (Glg. 3.1).

$$\frac{\mu_{mfcctrain}}{8} \leq \mu_{mfcctest} \leq \mu_{mfcctrain} + \frac{\mu_{mfcctrain}}{4} \tag{3.1}$$

Aus den Ergebnissen der DTW-Klassifizierung, die in Kapitel 4 näher erläutert werden, lässt sich ableiten, dass der 0. MFCC-Koeffizient die Erkennungsrate negativ beeinflusst, da durch seine hohen Werte der Distanzwertebereich vergrößert wird. Außerdem zeigt sich, dass die Verwendung der $\Delta\Delta$-Koeffizienten im Merkmalsvektor keinen nennenswerten Einfluss auf das Ergebnis des DTW hat. Da sich aber durch das Training des GMM ergibt, dass $\Delta\Delta$-Koeffizienten für gute Modellanpassung nötig sind, wird für die weiteren Tests zwar auf den 0. MFCC-Koeffizienten verzichtet, ansonsten aber der 39-dimensionale MFCC-Merkmalsvektor aus Versuch 2 verwendet (Tabelle 3.4). Die in Kapitel 4 ausgewer-

teten statistischen Kennzahlen Recall r, Precision p, F1-Score f1 und Accuracy acc zeigen, dass DTW als alleinige Klassifizierungsmethode im vorliegenden Fall nur mittelmäßige Ergebnisse liefert.

Deshalb wird auch eine Kombination aus DTW und GMM für die Klassifizierung getestet. Dieser Ansatz orientiert sich an dem Verfahren von Zhang et al. [90], die für ein komplett unüberwachtes Framework zur Schlüsselworterkennung den Einsatz von Gaußschen Posteriorgrammen vorschlagen. Diese repräsentieren die A-posteriori-Wahrscheinlichkeiten eines Audioframes für die im GMM verwendeten Gauß-Komponenten.

Mit Hilfe des zuvor trainierten GMMs wird für jede „Ordnungsruf"-Trainingsdatei das entsprechende Gaußsche Posteriorgramm extrahiert und als Merkmalsmatrix für den späteren Einsatz in der DTW-Analyse gespeichert. Aus den zu untersuchenden Audiospuren werden wie beim Training des GMMs 39-dimensionale MFCC-Merkmalsvektoren extrahiert. Die MFCC-Merkmalsvektoren der „mid-time-Frames" werden unter Verwendung des trainierten GMMs zur Berechnung des Gaußschen Posteriorgramms und des negativen Log-Likelihoods für jeden „mid-time-Frame" eingesetzt. Der Mittelwert des Log-Likelihoods wird als oberer Schwellwert zur Aussortierung relevanter Frames verwendet. Danach werden die Gaußschen Posteriorgramme der Trainings- und Testdateien als Merkmale im DTW verwendet, und es wird für jeden „mid-time-Frame" ein Distanzmittelwert berechnet. N Frames mit den kleinsten Distanzwerten werden als „True positive TP" erkannt. Für die Vergleichbarkeit mit dem MFCC-DTW-Ansatz im vorangegangenen Punkt wurde N auf die Anzahl der dort als „True positive TP" erkannten Frames gesetzt. Die Evaluationsergebnisse in Kapitel 4 zeigen, dass in diesem Fall die alleinige Verwendung von Gaußschen Posteriorgrammen als Merkmale für die DTW-Klassifizierung keine Verbesserung der Ergebnisse bringt.

In einem 3. Schritt werden deshalb die Verfahren der MFCC-DTW- und der Posterior-DTW-Klassifizierung kombiniert. Die Ergebnisse der Evaluierung belegen durch eine deutliche Verbesserung der Werte, dass von den getesteten Verfahren dieses am besten zur „Ordnungsruf"-Klassifizierung geeignet ist.

Audioklassen „Klatschen" und „Lachen"

Zur Erkennung von Umgebungsgeräuschen, deren Struktur nicht wie bei Sprache aus Phonemen besteht, wird als Klassifizierungsmethode ein binärer Entscheidungsbaum verwendet. Dabei wird im Fall der Erkennung von „Klatschen" mit einer Kombination aus zeit- und frequenzabhängigen Merkmalen gearbeitet. Für die Klassifizierung von „Lachen" kommen nur frequenzabhängige Merkmale zum Einsatz.

Die Merkmale jener „mid-time-Frames“, welche mit Hilfe des 0. MFCC-Koeffizienten als relevantes Ereignis klassifiziert wurden (siehe Kapitel 3.1.6), müssen folgende Bedingungen erfüllen, um als „Klatschen“ erkannt zu werden (Glg. 3.2 bis 3.5):

$$\sigma^2_{Energy} < \mu\left(\sigma^2_{Energy}\right) \times 1.1 \tag{3.2}$$

$$\sigma^2_{ZCR} < \mu\left(\sigma^2_{ZCR}\right) \times 1.2 \tag{3.3}$$

$$flux < \mu(flux) \tag{3.4}$$

$$sEE < \mu(sEE) \times 0.9 \tag{3.5}$$

Bei der anschließenden Gruppierung der „mid-time-Frames“ zu Audiosegmenten werden mit Hilfe einer Fensterfunktion Einzelframes aussortiert, da Klatschen in jedem Fall länger als 0,8 Sek. dauert. Lücken innerhalb des Fensters werden geschlossen. Abbildung 3.7 gibt einen Überblick des Klassifizierungsablaufes.

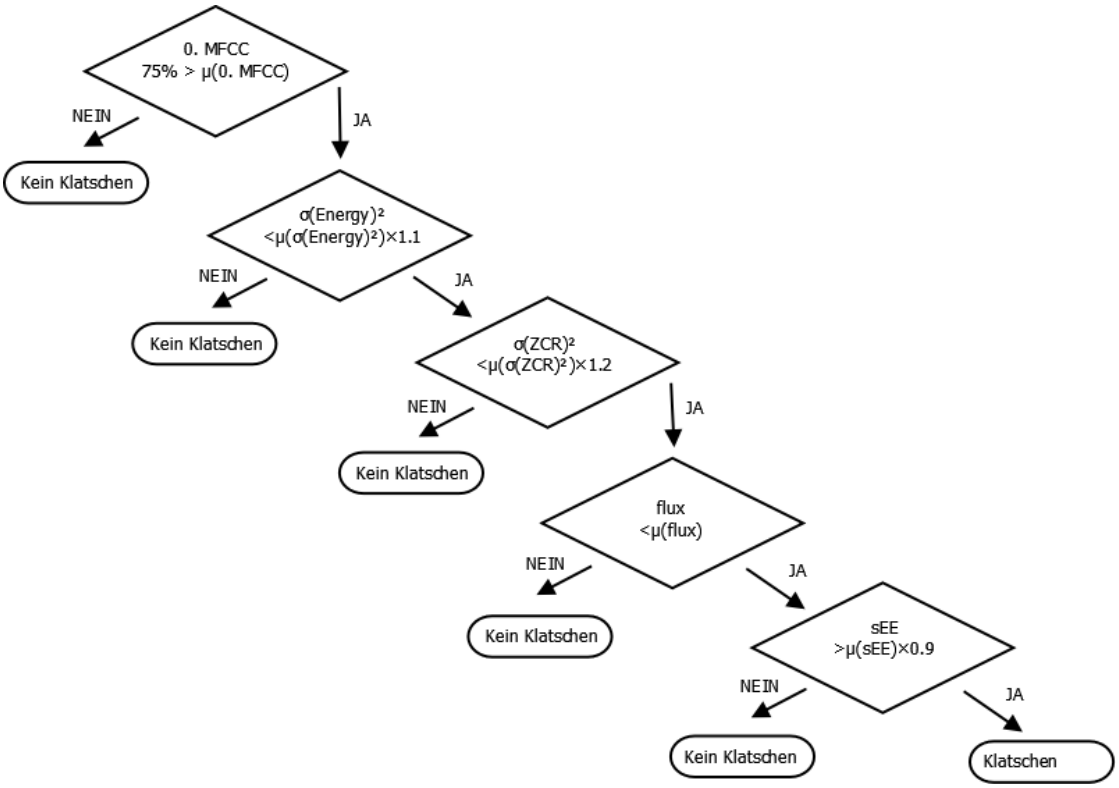

Abbildung 3.7: Binärer Entscheidungsbaum zur Klassifizierung eines „mid-time-Frames“ in die Klassen „Klatschen“ bzw. „Kein Klatschen“

Im Fall von „Lachen“ müssen die extrahierten Merkmale der „mid-time-Frames“ folgende Bedingungen erfüllen (Glg. 3.6 und 3.7):

$$\sigma_{sRO} < \mu\left(\sigma_{sRO}\right) \times 0.2 \tag{3.6}$$

$$\sigma_{sC} < \mu\left(\sigma_{sC}\right) \times 0.6 \tag{3.7}$$

Als letzten Schritt erfolgt ebenfalls die Gruppierung in Audiosegmente durch den Einsatz einer Fensterfunktion. Der binäre Entscheidungsbaum für das Erkennen von „Lachen“ wird in Abbildung 3.8 dargestellt.

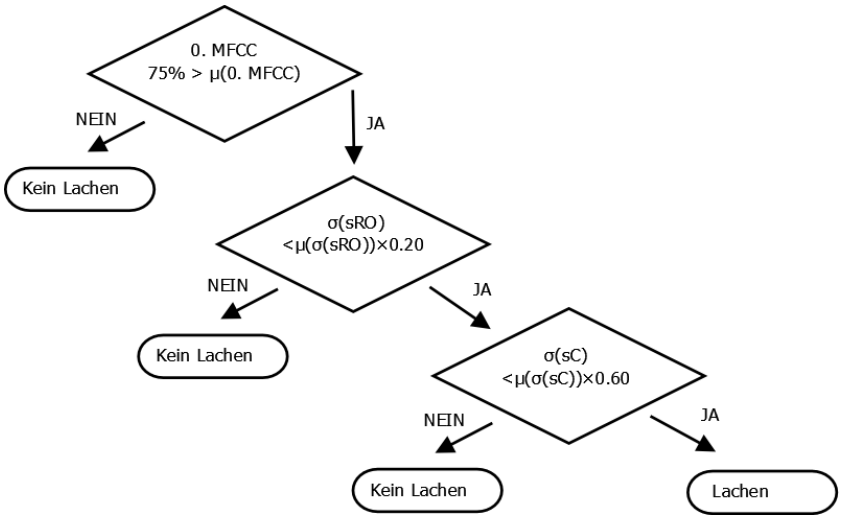

Abbildung 3.8: Binärer Entscheidungsbaum zur Klassifizierung eines „mid-time-Frames“ in die Klassen „Lachen“ bzw. „Kein Lachen“

Audioklasse „Zwischenruf“

Aus den Audiospuren werden wie beim Training der GMMs 32 zeit- und frequenzabhängige Merkmale und 39 MFCC-Merkmale extrahiert. Diese Merkmalsvektoren der „mid-time-Frames“ werden unter Verwendung der beiden trainierten GMMs zur Berechnung des negativen Log-Likelihood für jeden „mid-time-Frame“ eingesetzt. Die Zuordnung zur Klasse „Zwischenruf“ bzw. „Kein Zwischenruf“ erfolgt aufgrund des geringeren negativen Log-Likelihoods. Wie bei den Klassen „Klatschen“ und „Lachen“ wird die Gruppierung in Audiosegmente mit Hilfe einer Fensterfunktion umgesetzt.

3.2 Entwicklungsumgebung

Da aus den verwendeten Videos der Parlamentssitzungen sowohl Audio- als auch Bilddaten analysiert werden und zusätzlich ein User Interface programmiert werden soll, wird als Programmierumgebung MATLAB, Version R2016b, verwendet. MATLAB ist durch seinen Aufbau optimal für numerische Berechnungen geeignet und bietet durch seine verschiedenen Toolboxen, z.B. die Computer Vision System Toolbox, die Image Processing Toolbox, die Statistics Toolbox oder die Signal Processing Toolbox ein umfangreiches Paket an Methoden für die geplante Datenverarbeitung. Durch seine Popularität in der wissenschaftlichen Community bietet es den großen Vorteil, dass viele von wissenschaftlichen Instituten entwickelte Toolboxen zusätzlich gratis zur Verfügung stehen, welche außerdem dem neuesten Stand der Technik entsprechen [91]. Im Rahmen dieser Arbeit kommen folgende dieser zusätzlichen Funktionen und Toolboxen zum Einsatz:

- **VLFeat**
  Dabei handelt es sich um eine Open-Source-Bibliothek, die Bildverarbeitungsalgorithmen aus den Bereichen „Extraktion von lokalen Merkmalen“ und „Bildverstehen“ z.B. zu den Themen Fisher Vektoren, SIFT, k-Means und SVM zur Verfügung stellt. Zusätzlich zur MATLAB Toolbox gibt es auch eine C-Schnittstelle. Entwickelt wurde die Bibliothek von Forschern unterschiedlicher Universitäten wie Oxford, der University of California Los Angeles oder der Technischen Universität Prag [92].
- **VOICEBOX**
  Diese Sprachverarbeitungstoolbox für MATLAB wurde von Mike Brookes am Imperial College in London entwickelt. Sie beinhaltet unter anderem Algorithmen, um Audiodaten ein- bzw. auszulesen, GMMs zu generieren oder Sprachanalyse und Frequenzskalenumrechnungen durchzuführen [93].
- **RASTA/PLP/MFCC Paket von Daniel P. W. Ellis**
  Die Bibliothek von Daniel P. W. Ellis, die an der Columbia University entwickelt wurde, beinhaltet MATLAB-Routinen für die Audioverarbeitung unter anderem zur Berechnung von MFCCs [94].

Ein weiteres Plus ergibt sich aus der Möglichkeit, den MATLAB Code durch .mex-Dateien (MATLAB Executables), in C geschriebene Methoden, zu erweitern. Für die Erstellung eines User Interfaces bietet MATLAB den graphischen Layout-Editor GUIDE neben der Möglichkeit, kompliziertere Interfaces selbst zu programmieren. Zusätzlich zu den bereits erwähnten Vorteilen wird MATLAB auch deshalb ausgewählt, weil es die Möglichkeit bietet, verschiedene Ideen und Ansätze rasch testen und die erzielten Ergebnisse visualisieren zu können.

Für die Identifizierung und Markierung der relevanten Groundtruth-Audio-Ereignisse wird die Open-Source-Software „Sonic Visualiser“ verwendet. Dieses an der Queen-Mary-Universität in London entwickelte Programm bietet verschiedene Visualisierungsmöglichkeiten für Audiofiles (.wav, .ogg, .mp3) und Methoden für die Audioanalyse durch

das Vamp-Plugin [95]. Um die Groundtruth Daten für die Shot-Erkennung gewinnen zu können, wird das kostenlose Programm DaVinci Resolve 14 von Blackmagic Design eingesetzt, das Methoden für Schnitt, Farbkorrektur und Audiopostproduktion bietet und unter anderem eine automatische Shot-Erkennung inkludiert [96].

3.3 Graphische Benutzeroberfläche

Der im Zuge dieser Arbeit entwickelte Prototyp stellt auch eine graphische Benutzeroberfläche zur Verfügung. Diese wurde unter Berücksichtigung der in Kapitel 3.1.1 erläuterten Systemspezifikation entwickelt. Im Wesentlichen lässt sich die Oberfläche in folgende drei Bereiche einteilen: Training ①, Suche ② und Ergebnisse der Klassifizierung ③. Einen Überblick gibt Abbildung 3.9.

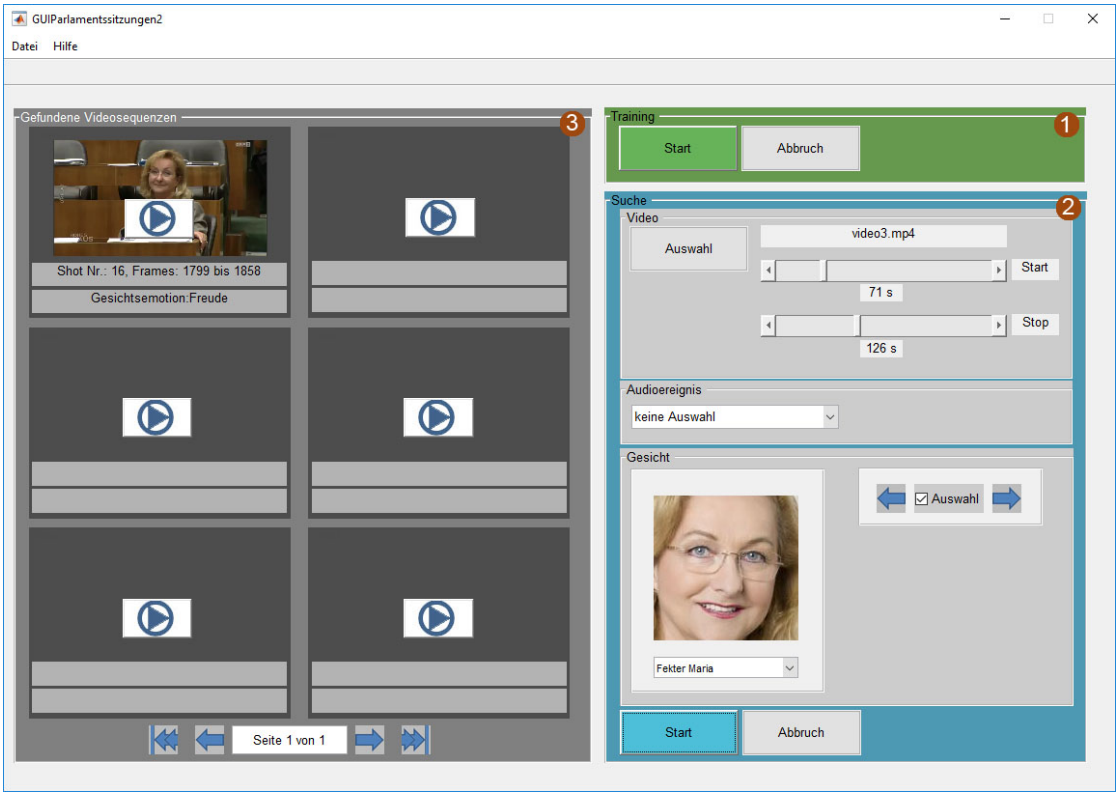

Abbildung 3.9: Graphische Benutzeroberfläche des Prototyps

Bevor nach bestimmten Audioereignissen oder Personen in Videomitschnitten von Parlamentssitzungen gesucht werden kann, müssen die Klassifikatoren für die Audioanalyse, die Gesichtsverifizierung und die Gesichtsemotionserkennung trainiert werden. Eine genaue Erklärung des Trainingsablaufes wird in Kapitel 3.1.5 und in Kapitel 3.1.6 gegeben. Der Start dieses Prozesses erfolgt durch die Anwenderinnen und Anwender in Bereich ①.

Nach erfolgreichem Abschluss der Trainingsphase können in Bereich ② die Parameter für die Suche eingegeben werden. Am Beginn steht die Auswahl des zu durchsuchenden Videos. Es besteht die Möglichkeit, die Suche auf einen bestimmten Abschnitt zu begrenzen, indem Start- und Stoppzeit mit den Schiebereglern verändert werden. Soll nach einem der vier Audioereignisse „Ordnungsruf“, „Klatschen“, „Zwischenruf“ oder „Lachen“ gesucht werden, erfolgt die Wahl der gewünschten Klasse mit Hilfe des Popupmenüs. Im nächsten Abschnitt des Suchbereiches werden die Gesichtsbilder der Abgeordneten angezeigt, nach denen gesucht werden kann. Die Auswahl kann entweder mit Hilfe der Pfeiltasten oder durch Selektion des entsprechenden Namens im Popupmenü erfolgen. Sie muss durch Setzen des Hakens in der Checkbox bestätigt werden. Es kann nach einem Audioereignis oder einer Person gesucht werden, aber auch nach einer Kombination aus beiden.

Die Ergebnisse der gewünschten Klassifizierung werden in Bereich ③ dargestellt. Repräsentativ für die positiv erkannten Shots wird jeweils der erste Frame des Shots angezeigt. Zusätzlich beinhaltet die Benutzeroberfläche die Nummer des Shots in aufsteigender Reihenfolge beginnend bei 1 und den Start- und Endframe. Außerdem wird die im Startframe erkannte Gesichtsemotion ausgegeben. Durch Drücken des Abspielknopfes kann der Shot in einem eigenen Fenster abgespielt werden (Abbildung 3.10).

Abbildung 3.10: Videoplayer zum Abspielen der Shots

KAPITEL 4

Statistische Evaluierung

In diesem Kapitel werden die für die Implementierung des Prototyps verwendeten Methoden einer statistischen Evaluierung unterzogen und eine Interpretation der damit erzielten Ergebnisse geliefert. Dies erfolgt getrennt für die einzelnen Teilbereiche Videosegmentierung, Gesichtserkennung und -verifizierung, Emotionserkennung und Erkennung von Audioereignissen. Die dazu erforderlichen statistischen Methoden wurden bereits in Kapitel 2.6 vorgestellt.

4.1 Shoterkennung

Wie in Kapitel 3.1.4 ausgeführt, wird die Erkennung der Shotgrenzen in den Videos der Parlamentssitzungen mittels der ECR-Methode realisiert unter Verwendung eines adaptiven Thresholds für die Entscheidungsfunktion. Basis für diese Wahl war die Annahme, dass die oft propagierte histogrammbasierte Methode für das vorliegende Videomaterial nicht geeignet ist, da die Histogramme der einzelnen Shots zu wenige Differenzen aufweisen. Der zweite ausschlaggebende Grund für diese Vorgehensweise war die Tatsache, dass in normalen Sitzungsmitschnitten fast ausschließlich harte Schnitte verwendet werden (Tabelle 4.1). Nach den Untersuchungen von Lienhart [26] eignen sich Techniken basierend auf dem Kantenänderungsverhältnis (ECR) für die Erkennung dieser Art der Schnitte. Video 3 bildet in der Gruppe der Ground-Truth-Videos eine Ausnahme, da es sich um keinen Mitschnitt, sondern einen ORF-Beitrag der Sendung „Hohes Haus“ handelt, in dem auch Schnitte mittels Überblendungen eingesetzt wurden. Es besteht aus einer Zusammenfassung diverser Parlamentssitzungen und wurde wegen seiner Konzentration an „Ordnungsruf-Szenen“ in die Ground-Truth-Daten integriert.

Die Auswertung der statistischen Kennzahlen Recall (r), Precision (p), F1-Score (f1) und Accuracy (acc) für die Shoterkennung in den Ground-Truth-Videos (Tabelle 4.1) zeigt deutlich, dass der gewählte Ansatz für die Mitschnitte von Parlamentssitzungen zu guten Erkennungsergebnissen führt. In dieser Kategorie (Video 1 und 2) konnte eine

durchschnittliche Accuracy (acc) von 92,2% erreicht werden. Diese fällt unter Einbeziehung von Video 3, für die die ECR-Methode eine deutlich niedrigere Accuracy (acc) liefert, auf einen durchschnittlichen Wert von 87,4%. Der Hauptgrund dafür sind die Schnitte, die mittels Überblendungen realisiert worden sind und nicht korrekt erkannt werden. Nachdem diese Art der Shotübergänge in normalen Sitzungsmitschnitten nicht verwendet wird, kann die negative Auswirkung auf das Ergebnis vernachlässigt werden.

Hervorzuheben ist eine durchschnittlich erreichte Precision (p) von 98,5%. Das heißt, dass es sich bei fast 100% der mit Hilfe der eingesetzten ECR-Methode erkannten Schnitte tatsächlich um Schnitte handelt. Auch der Recall (r), der Auskunft darüber gibt, wie viele Schnitte von allen vorkommenden Schnitten richtig erkannt werden konnten, liegt für die Standardvideos (Video 1 und 2) bei zufriedenstellenden 94,3%. Auch hier wirken sich die schlechteren Analyse-Ergebnisse von Video 3 negativ aus und drücken den Wert auf 88,8%.

	Länge	Schnitte		r	p	f1	acc
		H	Ü				
Video 1	05‘17	29	0	93,1%	100,0%	96,4%	93,1%
Video 2	08‘13	44	0	95,5%	95,5%	95,5%	91,3%
Video 3	05‘40	49	9	77,6%	100,0%	87,4%	77,6%
Durchschnitt				**88,8%**	**98,5%**	**93,1%**	**87,4%**

Tabelle 4.1: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Shoterkennungsmethode, angewandt auf die Ground-Truth-Videos (Harte Schnitte (H), Überblendungen (Ü))

4.2 Gesichtserkennung und -verifizierung

Alle relevanten Shots der Ground-Truth-Videos bilden die Grundlage für die statistische Auswertung der Gesichtserkennungsergebnisse. Dabei handelt es sich um Shots der Kategorien 1 bis 5 und Shots, die keiner der 9 Grundkategorien zugeordnet werden können (Abbildung 3.4). Für diese wurde eine zusätzliche Kategorie eingeführt. Generell werden drei Schlüsselbilder pro Shot (Anfangs-, Mittel- und Endframe) für die Evaluierung verwendet. Es erfolgt eine manuelle Markierung der zu detektierenden Gesichter, welche die Ground-Truth-Basis bildet. Unter Einbeziehung dieser Basis wird die verwendete Matlab-Implementierung des Viola-Jones-Algorithmus zur Gesichtserkennung ausgewertet.

Die Ergebnisse weisen eine durchschnittliche Precision (p) von 99,5% aus (Tabelle 4.2). Dieser hohe Wert bedeutet, dass es nur zu einem sehr geringen Prozentsatz Fehlerkennungen von anderen Objekten gibt. Was sich deutlich zeigt, ist die Tatsache, dass die Auflösung des Videomaterials einen direkten Einfluss auf die Kennzahlen Recall (r), F1-Score (f1) und Accuracy (acc) hat. Die Einschränkung durch die Mindestgröße von 75 x 75 Pixeln für gesuchte Gesichter, die für die nachfolgende Gesichtsverifizierung und

Emotionserkennung gefordert wird, wirkt sich in Videos mit geringer Auflösung (insb. Video 2) negativ auf diese Kennzahlen aus. Das führt in der Gesamtbetrachtung zu geringeren Durchschnittswerten.

	Länge	Auflösung	Schnitte			Anz. G	r	p	f1	acc
			H	Ü	R					
Video 1	05‘17	848x480	29	0	18	72	80,6%	98,4%	88,7%	79,5%
Video 2	08‘13	432x240	44	0	39	165	23,7%	100,0%	38,4%	23,7%
Video 3	05‘40	636x360	49	9	37	135	50,4%	100,0%	67,1%	50,4%
Durchschnitt							**51,6%**	**99,5%**	**64,8%**	**51,2%**

Tabelle 4.2: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der implementierten Gesichtserkennung mittels Viola-Jones-Algorithmus, Harte Schnitte (H), Überblendungen (Ü), relevante Shots (R), Anzahl der relevanten Gesichter (Anz. G)

Die Untersuchungen von Lisin et al. [30] zur Objekterkennung gaben den Anstoß, den Einsatz von lokalen und globalen Merkmalen für die Gesichtsverifizierung zu testen. Lisin et al. erreichen unter Verwendung von lokalen oder globalen Merkmalen eine Accuracy (acc) von etwa 54% bei der Objektklassifizierung von Plankton in 14 Klassen. Durch die Kombination der beiden Merkmalsklassen ist eine Steigerung auf 65,5% möglich [30].

Im Fall der vorliegenden Gesichtsverifizierung sollen die 39 Personenklassen der Ground-Truth-Videos unterschieden werden. Getestet wird ein Ansatz mit globalen HOG-Merkmalen und ein zweiter, der die HOG-Merkmale lokal aus den 10 wichtigsten „SURF-Regions of Interest“ berechnet (Kapitel 2.3 und 3.1.5). Es wird mit zwei Größen (75 x 75 px und 150 x 150 px) der Gesichtsbilder trainiert und getestet, um festzustellen, ob sich die Größe auf das Ergebnis auswirkt. Die Trainings- bzw. Testdatenbank enthält einmal 232 Bilder und wird dann zum Vergleich auf die 171 besten Bilder reduziert. Getestet wird mittels Kreuzvalidierung unter Einsatz der „Hold out-Methode“ (Kapitel 2.6.2) mit einer Aufteilung in 85% Trainings- und 15% Testbilder. In den Tabellen 4.3 und 4.4 sind die Ergebnisse der unterschiedlichen Settings dargestellt. Mit den globalen HOG-Merkmalen (Tests 1 bis 3) kann eine durchschnittliche Accuracy (acc) von 52,3% für die Gesichtsverifizierung erreicht werden. Es ist ersichtlich, dass die Größe der verwendeten Gesichtsbilder nur einen unwesentlichen Einfluss auf das Ergebnis hat. Auffällig ist, dass die Verwendung der verkleinerten Trainingsdatenbank zu einem besseren Ergebnis führt (Test 3). Der lokale Ansatz (Tests 4 bis 6) liefert eine nicht zufriedenstellende Accuracy (acc) von durchschnittlich nur 3,3% und wird deshalb nicht weiter verfolgt.

Die Parameter von Test 1 und 3 (Tabelle 4.3) werden nun für die Auswertung der Gesichtsverifizierung der im vorangegangenen Schritt der Gesichtserkennung entdeckten Gesichtsbilder verwendet. Die Ergebnisse liegen mit einer durchschnittlichen Accuracy (acc) von 20,8% unter den Erwartungen. Der Eindruck, dass die Verkleinerung des Testsets generell zu einer Verbesserung der Erkennungsrate führt, lässt sich in der Auswertung der Ground-Truth-Videos nicht bestätigen (Tabellen 4.5 und 4.6). Die schlechten

Ergebnisse sind vermutlich auf das Gegenteil zurückzuführen - eine zu kleine Trainingsdatenbank für die große Anzahl von Klassen. Außerdem weisen die Gesichtsbilder einer Klasse im Training und in der Suche Unterschiede in der Bildqualität, der Frisur, der Belichtung, dem Bartwuchs und der Verwendung von Brillen auf. Das schlägt sich auch in der Accuracy (acc) der Verifizierung nieder.

	Merkmale	Größe Trainingsset	Größe Testset	Größe Gesicht	TP	FN	acc
Test 1	HOG	85%, 198 Gesichtsbilder	15%, 34 Gesichtsbilder	75 x 75 px	15	19	44,2%
Test 2	HOG	85%, 198 Gesichtsbilder	15%, 34 Gesichtsbilder	150 x 150 px	16	18	47,1%
Test 3	HOG	85%, 142 Gesichtsbilder	15%, 29 Gesichtsbilder	75 x 75 px	19	10	65,6%
Durchschnitt							**52,3%**

Tabelle 4.3: Test der Gesichtsverifizierung mit globalen HOG-Merkmalen und unterschiedlichen Parametern, True positive (TP), False negative (FN), Accuracy (acc)

	Merkmale	Größe Trainingsset	Größe Testset	Größe Gesicht	TP	FN	acc
Test 4	10 SURF-Regions HOG	85%, 198 Gesichtsbilder	15%, 34 Gesichtsbilder	75 x 75 px	1	33	3,0%
Test 5	10 SURF-Regions HOG	85%, 198 Gesichtsbilder	15%, 34 Gesichtsbilder	150 x 150 px	0	34	0,0%
Test 6	10 SURF-Regions HOG	85%, 142 Gesichtsbilder	15%, 29 Gesichtsbilder	75 x 75 px	2	27	6,9%
Durchschnitt							**3,3%**

Tabelle 4.4: Test der Gesichtsverifizierung mit lokalen SURF-HOG-Merkmalen und unterschiedlichen Parametern, True positive (TP), False negative (FN), Accuracy (acc)

Trainingsset 232 Gesichtsbilder, 75 x 75 px, HOG-Merkmale	Länge	Anz. G aus Erkennung	TP	FN	r	p	f1	acc
Video 1	05‘17	58	2	56	3,5%	100%	6,8%	3,5%
Video 2	08‘13	39	12	27	30,8%	100%	47,1%	30,8%
Video 3	05‘40	68	19	49	28,0%	100%	43,8%	28,0%
Durchschnitt					**20,8%**	**100%**	**32,6%**	**20,8%**

Tabelle 4.5: Ergebnisse der Gesichtsverifizierung mit globalen HOG-Merkmalen für die Gesichtsbilder der Ground-Truth-Videos (größere Trainingsdatenbank), True positive (TP), False negative (FN), Recall (r), Precision (p), F1 (f1), Accuracy (acc)

Trainingsset 171 Gesichtsbilder, 75 x 75 px, HOG-Merkmale	Länge	Anz. G aus Erkennung	TP	FN	r	p	f1	acc
Video 1	05‘17	58	4	54	6,9%	100%	13,0%	6,9%
Video 2	08‘13	39	9	30	23,1%	100%	37,6%	23,1%
Video 3	05‘40	68	14	54	20,6%	100%	34,2%	20,6%
Durchschnitt					**16,9%**	**100%**	**28,3%**	**16,9%**

Tabelle 4.6: Ergebnisse der Gesichtsverifizierung mit globalen HOG-Merkmalen für die Gesichtsbilder der Ground-Truth-Videos (kleinere Trainingsdatenbank), True positive (TP), False negative (FN), Recall (r), Precision (p), F1 (f1), Accuracy (acc)

4.3 Emotionserkennung

Die aus den 204 Videos der MMI-Datenbank generierten 730 Emotionsgesichtsbilder der Klassen „Zorn“, „Freude“, „Trauer“ und „Überraschung“ mit einer Größe von 75 x 75 px werden für die spätere Kreuzvalidierung mittels „Hold out-Methode“ (Kapitel 2.6.2) in ein Trainings- und ein Testset im Verhältnis 85% zu 15% eingeteilt (Tabelle 4.7).

	Aufteilung	Gesamt Bilder	Zorn	Freude	Trauer	Überraschung
Training	85%	620	147	151	142	180
Test	15%	110	26	27	25	32
Gesamt	**100%**	**730**	**173**	**178**	**167**	**212**

Tabelle 4.7: Aufteilung der MMI-Emotionsgesichtsbilder in Trainings- und Testset

	Gesichter	Zorn	Freude	Trauer	Überraschung
Video 1	58	30	19	6	3
Video 2	39	14	9	12	4
Video 3	68	41	20	4	3
Gesamt	**165**	**85**	**48**	**22**	**10**

Tabelle 4.8: Ground-Truth-Zuordnung der in der Gesichtserkennung gefundenen, relevanten Bilder in die 4 Emotionsklassen

Für die Erkennung der Gesichtsemotion werden SURF-Merkmale extrahiert und damit BoVW-Histogramme erstellt, welche für das Training der SVM-Klassifikatoren verwendet werden (Kapitel 2.3.2 und 2.5.1). Die Konfusionsmatrix der Klassifizierung unter Verwendung des Testsets (Tabelle 4.9) zeigt mit einer durchschnittlichen Accuracy (acc) von nur 53% bereits, dass die Emotionsklassifizierung unter den angenommenen Bedingungen nicht optimal funktioniert. Gute Erkennungswerte liefern die Klassen „Trauer“ mit einer Accuracy (acc) von 88% und „Überraschung“ mit einer Accuracy (acc) von

69%. Die höchsten Missklassifikationswerte liefert die Klasse „Zorn“, die zu 58,0% der Klasse „Trauer“ zugeordnet und nur mit 23%iger Accuracy (acc) erkannt wird. In einem Vergleichstest wurde das Trainingsset auch als Testset verwendet. Die Ergebnisse sind in Tabelle 4.10 zusammengefasst. Die durchschnittliche Accuracy (acc) liegt hier bei etwa 90%.

		Vorhersage			
		Zorn	Freude	Trauer	Überraschung
G T	Zorn	**23,0%**	0,0%	58,0%	19,0%
	Freude	30,0%	**33,0%**	26,0%	11,0%
	Trauer	8,0%	4,0%	**88,0%**	0,0%
	Überraschung	0,0%	0,0%	31,0%	**69,0%**

Tabelle 4.9: Konfusionsmatrix der Emotionsklassifizierung, 85% Trainingsset, 15% Testset, durchschnittliche Accuracy (acc) = 53%, Ground-Truth-Zuordnung (GT)

		Vorhersage			
		Zorn	Freude	Trauer	Überraschung
G T	Zorn	**84,0%**	1,0%	12,0%	3,0%
	Freude	4,5%	**88,0%**	4,5%	3,0%
	Trauer	3,0%	1,0%	**95,0%**	1,0%
	Überraschung	1,0%	0,5%	5,5%	**93,0%**

Tabelle 4.10: Konfusionsmatrix der Emotionsklassifizierung, Trainingsset = Testset, durchschnittliche Accuracy (acc) = 90%, Ground-Truth-Zuordnung (GT)

Für die Evaluierung der Emotionserkennung im Prototyp werden die in der Gesichtserkennung detektierten Gesichter den vier gesuchten Emotionsklassen zugeordnet und die Performance der trainierten Klassifikatoren auf diesem Set ausgewertet. Die Umsetzung der Grundeinteilung ist allerdings nicht ganz einfach und erfolgt hauptsächlich aufgrund der Mundpartie. Um eine tatsächlich richtige Ground-Truth-Klassifizierung zu erhalten, müsste ein Experte hinzugezogen werden. Tabelle 4.8 gibt einen Überblick und lässt erkennen, dass die Emotionen „Zorn“ und „Freude“ in den betrachteten Sitzungsvideos vorherrschend sind. Die Konfusionsmatrizen der Klassifizierung (Tabellen 4.11 bis 4.13) zeigen, dass die Klasse „Freude“ durch die trainierten Klassifikatoren mit einer durchschnittlichen Accuracy (acc) von 60% am besten erkannt wird. Die größten Probleme bereitet, nachdem sie in der Klassifizierung des Testsets das beste Ergebnis lieferte, die Klasse „Trauer“ mit einer durchschnittlichen Accuracy (acc) von nur 8,5%. Sie wird in hohem Maß (durchschnittlich 72%) der Klasse „Freude“ zugeordnet. Die Klasse „Überraschung“ wird in den beiden Ground-Truth-Videos 1 und 2 zu 100% falsch klassifiziert, in Video 3 dagegen zu 100% richtig. Diese großen Unterschiede lassen sich auf die Ausgeprägtheit der Gesichtsausdrücke zurückführen - für das Training dieser Klasse

wurden sehr extreme Emotionsbilder verwendet. Im Alltag zeigt sich Überraschung meist viel verhaltener in der Gesichtsmimik.

Die Klassifikationsprobleme haben verschiedene Gründe. Bereits im Trainingsset sind die Gesichtsemotionen nicht immer eindeutig erkennbar dargestellt. Die Gesichtsbilder der Ground-Truth-Videos sind teilweise Profilbilder, was die Emotionserkennung erschwert. Zusätzlich bereiten Bartwuchs, Brillen und Verdeckungen des Mundes durch Hände Schwierigkeiten in der Erkennung.

		Vorhersage			
		Zorn	Freude	Trauer	Überraschung
GT	Zorn	**20,0%**	53,0%	20,0%	7,0%
	Freude	21,0%	**68,0%**	0,0%	11,0%
	Trauer	0,0%	83,0%	**17,0%**	0,0%
	Überraschung	0,0%	100,0%	0,0%	**0,0%**

Tabelle 4.11: Konfusionsmatrix der Emotionsklassifizierung, GT-Video 1, durchschnittliche Accuracy (acc) = 26%, Ground-Truth-Zuordnung (GT)

		Vorhersage			
		Zorn	Freude	Trauer	Überraschung
GT	Zorn	**21,5%**	50,0%	21,5%	7,0%
	Freude	11,0%	**67,0%**	22,0%	0,0%
	Trauer	25,0%	58,0%	**8,5%**	8,5%
	Überraschung	25,0%	50,0%	25,0%	**0,0%**

Tabelle 4.12: Konfusionsmatrix der Emotionsklassifizierung, GT-Video 2, durchschnittliche Accuracy (acc) = 24%, Ground-Truth-Zuordnung (GT)

		Vorhersage			
		Zorn	Freude	Trauer	Überraschung
GT	Zorn	**34,0%**	44,0%	22,0%	0,0%
	Freude	15,0%	**45,0%**	25,0%	15,0%
	Trauer	0,0%	75,0%	**0,0%**	25,0%
	Überraschung	0,0%	0,0%	0,0%	**100,0%**

Tabelle 4.13: Konfusionsmatrix der Emotionsklassifizierung, GT-Video 3, durchschnittliche Accuracy (acc) = 45%, Ground-Truth-Zuordnung (GT)

4.4 Erkennung von Audioereignissen

Die Erkennung der unterschiedlichen Audioklassen wird mit Hilfe verschiedener, den Klasseneigenschaften angepasster, Methoden umgesetzt. Eine genaue Erklärung der eingesetzten Algorithmen und Merkmale wird in Kapitel 3.1.6 gegeben. Die nachfolgende

Auswertung der statistischen Kennzahlen Recall (r), Precision (p), F1-Score (f1) und Accuracy (acc) auf Basis der „mid-time-Frames“ - separat für jede Klasse - zeigt ein durchwachsenes Ergebnis. Es ist mit den gewählten Methoden möglich, eine durchschnittliche Accuracy (acc) zwischen 94,9% und 98,2% zu erreichen. Auch die durchschnittlichen Recall-Werte (r) liegen über alle Klassen hinweg zwischen 68,0% und 70,2%. Das bedeutet, dass der Anteil der korrekten Klassifikationen im Verhältnis zu allen Klassifikationen sehr hoch ist, und auch der Anteil der korrekt positiv klassifizierten Ereignisse im Verhältnis zu allen relevanten Ereignissen mit etwa 70,0% noch im Mittelfeld liegt. Für die Klasse „Klatschen“ wird zudem durchschnittlich eine Precision (p) von 75,4% und ein F1-Score (f1) von 71,6% erreicht. Diese positiven Ergebnisse werden von niedrigen Precision-Werten (p) von rund 40,0% für die Klassen „Ordnungsruf“ und „Zwischenruf“ und dadurch verursachten schlechten Ergebnissen für den F1-Score (f1) dieser Klassen geschmälert. Das ist zum Teil auf die kleinen Trainingssets für die GMMs zurückzuführen und liegt sicher auch daran, dass sich die Klassen manchmal überlagern. Zudem ist eine Generalisierung der Ergebnisse für die Klasse „Lachen“ schwierig, da es für diese Klasse nur in einem der Ground-Truth-Videos entsprechende Ereignisse gibt. Im folgenden Abschnitt werden die Evaluierungsergebnisse detailliert dargestellt.

Audioklasse „Ordnungsruf“

Die Klassifizierung der Ordnungsrufe wird in einem ersten Schritt durch einen DTW-Ansatz (Kapitel 3.1.6 und 3.1.7) unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Tabelle 3.4) realisiert. Die Auswertung der Evaluierung (Tabellen 4.14 bis 4.17) zeigt deutlich, dass der 0. MFCC-Koeffizient die Ergebnisse des DTWs negativ beeinflusst (Versuch 1 und 4).

V1 MFCC-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	50,0%	1,1%	2,2%	88,3%
Video 2	08‘13	11	11	54,5%	5,3%	9,7%	90,9%
Video 3	05‘40	25	25	72,0%	16,7%	27,1%	88,6%
Durchschnitt				**58,9%**	**7,7%**	**13,0%**	**89,3%**

Tabelle 4.14: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 1, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

Die Versuche 2 und 3 unterscheiden sich nur durch die Inkludierung bzw. Nicht-Inkludierung der $\Delta\Delta$-Koeffizienten im MFCC-Merkmalsvektor. Sie liefern sehr ähnliche Ergebnisse. Daraus kann geschlossen werden, dass diese Koeffizienten keinen Einfluss auf die DTW-Klassifizierung haben. Sie sind aber für die gute Modellanpassung im Training des GMMs nötig. Aus diesen Gründen wird für die weiteren Tests zwar auf den 0. MFCC-Koeffizienten verzichtet, ansonsten aber der 39-dimensionale MFCC-Merkmalsvektor aus Versuch 2 verwendet.

V2 MFCC-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	100,0%	6,7%	12,6%	96,5%
Video 2	08‘13	11	11	54,5%	25,0%	34,3%	98,1%
Video 3	05‘40	25	25	64,0%	32,7%	43,3%	95,1%
Durchschnitt				**72,9%**	**21,5%**	**30,1%**	**96,6%**

Tabelle 4.15: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 2, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

V3 MFCC-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	100,0%	6,9%	12,9%	96,6%
Video 2	08‘13	11	11	54,5%	25,0%	34,3%	98,1%
Video 3	05‘40	25	25	64,0%	32,7%	43,3%	95,1%
Durchschnitt				**72,9%**	**21,6%**	**30,2%**	**96,6%**

Tabelle 4.16: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 3, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

V4 MFCC-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	50,0%	0,9%	1,8%	86,0%
Video 2	08‘13	11	11	54,5%	4,7%	8,7%	89,7%
Video 3	05‘40	25	25	72,0%	14,9%	24,7%	87,1%
Durchschnitt				**58,9%**	**6,9%**	**11,8%**	**87,6%**

Tabelle 4.17: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 4, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

Da die Ergebnisse der MFCC-DTW-Klassifizierung verbesserungswürdig sind, speziell mit Blick auf eine durchschnittliche Precision (p) von 21,5% und einen durchschnittlichen F1-Score (f1) von 30,1% in Versuch 2, wird in einem weiteren Ansatz eine Klassifizierung mittels DTW unter Verwendung von Gaußschen Posteriorgrammen getestet. Aufgrund der wenig zufriedenstellenden Ergebnisse (Tabelle 4.18) werden in einem 3. Schritt die beiden Verfahren kombiniert, womit eine Steigerung der Precision (p) auf 36,5% und des F1-Scores (f1) auf 41,1% erreicht werden kann (Tabelle 4.19). Abbildung 4.1 stellt die Entwicklung des F1-Scores (f1) für die 3 Klassifizierungsmethoden dar.

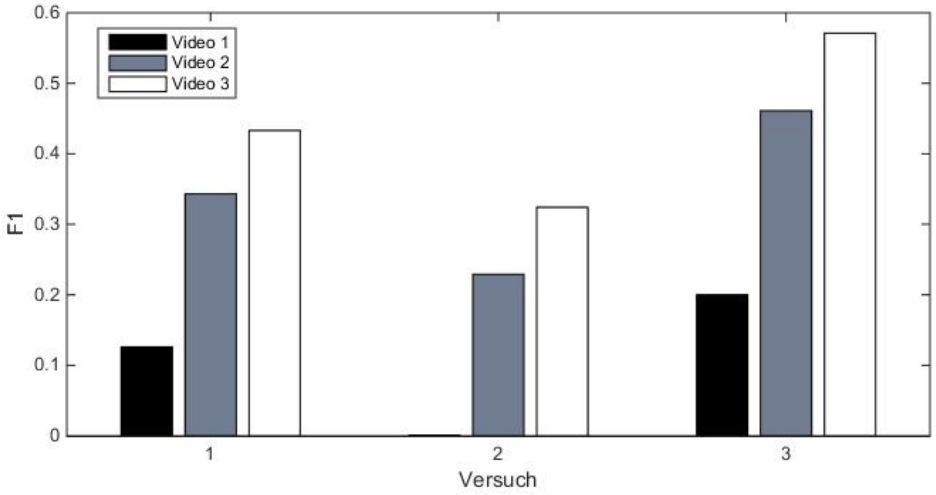

Abbildung 4.1: Vergleich F1 (f1) der 3 angewandten Klassifizierungsmethoden MFCC-DTW (Versuch 1), Posterior-DTW (Versuch 2) und Kombination der beiden Verfahren (Versuch 3) für die Klasse „Ordnungsruf“, angewandt auf die drei Ground-Truth-Videos

Posterior-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	0,0%	0,0%	0,0%	96,0%
Video 2	08‘13	11	11	36,4%	16,7%	22,9%	97,8%
Video 3	05‘40	25	25	48,0%	24,5%	32,4%	94,1%
Durchschnitt				**28,2%**	**13,8%**	**18,5%**	**96,0%**

Tabelle 4.18: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der Posterior-DTW für die Klasse „Ordnungsruf“, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

Kombi MFCC-DTW und Posterior-DTW	Länge	E	F	r	p	f1	acc
Video 1	05‘17	2	2	100,0%	11,1%	20,0%	98,0%
Video 2	08‘13	11	11	54,5%	40,0%	46,1%	98,9%
Video 3	05‘40	25	25	56,0%	58,3%	57,1%	97,5%
Durchschnitt				**70,2%**	**36,5%**	**41,1%**	**98,2%**

Tabelle 4.19: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der Kombination von MFCC-DTW und Posterior-DTW für die Klasse „Ordnungsruf“, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F)

Audioklassen „Klatschen“ und „Lachen“

Die Klassifizierung für beide Klassen erfolgt mittels eines binären Entscheidungsbaumes unter Verwendung von zeit- und frequenzabhängigen Merkmalen (Kapitel 3.1.7). Im Vergleich der Evaluationsergebnisse (Tabelle 4.20) der Klasse „Klatschen“ zeigt sich für Video 1 und 2 ein zufriedenstellendes Bild. Die Ergebnisse für Video 3 sind im Gegensatz dazu wesentlich schlechter, was vermutlich an der unterschiedlichen Audiostruktur liegt. In Video 3 wurde teilweise Musikunterlegung verwendet. Das könnte das Ergebnis negativ beeinflussen und ist in normalen Parlamentssitzungsvideos nicht üblich.

Die Klasse „Lachen“ stellt überhaupt einen Sonderfall dar, da es für diese Klasse nur in einem der Testvideos entsprechende Ereignisse gibt. Die Aussage, die getroffen werden kann, ist, dass der gewählte Ansatz in Bezug auf die Accuracy (acc), mit einem durchschnittlichen Wert von 98,6%, zufriedenstellende Ergebnisse liefert (Tabelle 4.21).

	Länge	E	F	r	p	f1	acc
Video 1	05‘17	6	27	85,2%	76,7%	80,7%	98,6%
Video 2	08‘13	12	151	73,5%	83,5%	78,2%	95,9%
Video 3	05‘40	9	64	48,4%	66,0%	55,8%	94,2%
Durchschnitt				**69,1%**	**75,4%**	**71,6%**	**96,3%**

Tabelle 4.20: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Entscheidungsbaummethode für die Klasse „Klatschen“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Klatschen“ (E), Anzahl der Frames, die der Klasse „Klatschen“ zugeordnet werden (F)

	Länge	E	F	r	p	f1	acc
Video 1	05‘17	0	0	-	-	-	99,4%
Video 2	08‘13	0	0	-	-	-	97,2%
Video 3	05‘40	2	12	42,0%	100,0%	59,2%	99,2%
Durchschnitt							**98,6%**

Tabelle 4.21: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Entscheidungsbaummethode für die Klasse „Lachen“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Lachen“ (E), Anzahl der Frames, die der Klasse „Lachen“ zugeordnet werden (F)

Audioklasse „Zwischenruf“

Für die Klasse „Zwischenruf“ ist es schwierig, Merkmale zu finden, die eine eindeutige Unterscheidung von den anderen Klassen ermöglichen. Mit dem propagierten GMM-Ansatz (Kapitel 3.1.6) sind ähnliche Ergebnisse wie in der „Ordnungsruf“-Klassifizierung mittels der Kombination von MFCC-DTW und Posterior-DTW erreichbar (Tabelle 4.22). Es wird angenommen, dass eine Verbesserung der Ergebnisse auch hier durch die Vergrößerung des Trainingssets für die GMMs erreicht werden könnte.

	Länge	E	F	r	p	f1	acc
Video 1	05‘17	3	7	57,1%	23,5%	33,3%	98,0%
Video 2	08‘13	5	76	99,9%	47,8%	64,7%	93,3%
Video 3	05‘40	4	64	46,9%	55,6%	50,9%	93,2%
Durchschnitt				**68,0%**	**42,3%**	**49,7%**	**94,9%**

Tabelle 4.22: Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten GMM-Methode für die Klasse „Zwischenruf“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Zwischenruf“ (E), Anzahl der Frames, die der Klasse „Zwischenruf“ zugeordnet werden (F)

KAPITEL 5

Zusammenfassung und Ausblick

Die Ergebnisse der Evaluierung (Kapitel 4) aller Teilschritte des Prototyps der Suchmaschine für Videos von Parlamentssitzungen zeigen deutlich, dass der inhaltsbasierte, multimodale Ansatz grundsätzlich für die automatisierte Auswertung dieser Art von Videos gut geeignet ist. Die Fokussierung auf die Erkennung von Audioereignissen zur Detektion von relevanten Szenen hat sich als richtig erwiesen, da die Audiomerkmale im vorliegenden Fall aussagekräftiger sind als die Bildmerkmale. Probleme bereitet die Tatsache, dass die Audioereignisse selten in Reinform vorliegen, sondern in den meisten Fällen als Überlagerung mehrerer Signalgruppen vorkommen, zum Beispiel in den Kombinationen Sprache/Klatschen, Zwischenrufe/Sprache oder Ordnungsrufe/Klatschen. Deshalb wäre im nächsten Schritt ein polyphoner Analyseansatz denkbar, der das gleichzeitige Auftreten von Audioklassen berücksichtigt. Zur weiteren Verbesserung der Ergebnisse wird auf jeden Fall die Vergrößerung der Trainingssets für die GMMs der Klassen „Ordnungsruf“ und „Zwischenruf“ empfohlen.

Die Auswertung der visuellen Merkmale liefert in den einzelnen Teilbereichen sehr unterschiedliche Ergebnisse. Die implementierte Shot-Erkennung mittels ECR-Methode hat sich als passend für diese Art von Videomaterial erwiesen. Nach den Erkenntnissen aus der Evaluierung der getesteten Methoden wird für alle weiteren visuellen Auswertungen eine Mindestauflösung des zugrunde liegenden Videomaterials von zumindest 848 x 480 Pixeln empfohlen. Damit könnten die Werte der Gesichtserkennung, die die Grundlage für die weitere visuelle Analyse bildet, positiv beeinflusst werden. Die in der Gesichtsverifizierung und in der Analyse der Gesichtsemotion verwendeten HOG-Merkmale sind eine gute Basis. Zur Verbesserung der Ergebnisse der Gesichtsverifizierung ist es in einem nächsten Schritt nötig, die Trainingsdatenbank zu vergrößern. Hierbei wäre es wichtig, standardisierte Frontal- und Profilbilder der Abgeordneten mit gleicher Auflösung, und aufgenommen unter gleichbleibenden Lichtbedingungen im Parlament, zu verwenden. Denkbar wäre auch, die Anzahl der Schlüsselbilder für die Analyse zu erhöhen, um die Merkmalsbasis zu vergrößern. Das beeinflusst allerdings die Geschwindigkeit der Auswertung negativ.

Zusätzlich kann über die Ausweitung der Multimodalität auf die Textanalyse von Inserts zur Stützung der Personenerkennung nachgedacht werden. Die Erkennung der Gesichtsemotion hat sich als sehr problematisch erwiesen. Die Gesichtsmimik der Abgeordneten ist in vielen Fällen für eine korrekte Auswertung nicht ausgeprägt genug. Zur Verbesserung empfiehlt sich die Überprüfung eines Ansatzes unter Verwendung eines Gesichtstrackers der „Onset“, „Apex“ und „Offset“ einer Emotion berücksichtigt.

In einem nächsten Schritt könnte der implementierte Prototyp um eine Sprachanalyse unter Verwendung eines der Toolkits, die in Kapitel 2.4.3 vorgestellt wurden, erweitert werden. Dadurch wäre die Möglichkeit gegeben, nach bestimmten Themen durch dafür markante Wörter zu suchen, oder aber nach bestimmten Personen durch die Analyse ihrer Sprache.

Abbildungsverzeichnis

2.1 Schritte einer inhaltsbasierten, multimodalen Videoindexierung und Suche . . 6
2.2 Schritte der Personenerkennung unter Verwendung von HOG-Merkmalen, Quelle: Dalal und Triggs [15] 10
2.3 Gesicht: Original und extrahierte HOG-Merkmale 11
2.4 Die 2. partiellen Ableitungen nach y (L_{yy}) und nach xy (L_{xy}) und die entsprechenden Mittelwertfilter-Approximationen D_{yy} und D_{xy}, Quelle: Bay et al. [17] 12
2.5 Aufbau des Skalierungsraumes bei SIFT- (links) und SURF-Merkmalen (rechts), Quelle: Bay et al. [17] 13
2.6 Haar-Wavelet-Filter für x- und y-Richtung, dunkle Regionen haben das Gewicht -1 und helle +1, Quelle: Bay et al. [17] 14
2.7 Berechnung der dominierenden Orientierung eines „Point of interest“ mit Hilfe eines Fensters, das in mehreren Schritten über die Kreisregion geschoben wird, Quelle: Bay et al. [17] 14
2.8 „Points of interest“ und ihre umschließenden Merkmals-Regionen in einem Gesichtsbild 14
2.9 Berechnung des 64-dimensionalen SURF-Merkmalsvektors aus der einen „Point of interest“ umschließenden Region, Quelle: Bay et al. [17] 14
2.10 Schritte eines Systems zur Gesichtsverifikation, Quelle: in Anlehnung an Wójcik et al. [33] 15
2.11 Die Phasen eines Gesichtsaudruckes, Quelle: Wu et al. [44] 18
2.12 MEL-Filterbank, Quelle: Murali et al. [46] 25
2.13 Schritte der BoVW-Generierung, Quelle: Hentschel und Sack [24] 29
2.14 Binäre Klassentrennung durch eine SVM mit Hyperebene und Stützvektoren, Quelle: Kotsiantis [22] 31
2.15 Links-Rechts-HMM, Quelle: Euler [71] 33
2.16 Mögliche Verknüpfung zweier Wörter durch ein Kostenraster des DTW, Quelle: Lama und Namburu [73] 34
2.17 „Confusion Matrix“ eines binären Klassifikators, Quelle: in Anlehnung an Marom et al. [76] 35

3.1 Implementierungsschritte für den Prototyp 40

3.2 Originalbild und extrahiertes Gesichtsbild, Originalfoto: Parlamentsdirektion | Mike Ranz 42
3.3 Aus den Videos der MMI-Datenbank extrahierte Gesichtsbilder der 6 Emotionsklassen Zorn, Ekel, Angst, Freude, Trauer, Überraschung 43
3.4 Neun Shotkategorien österreichischer Parlamentssitzungen (von links oben nach rechts unten) 44
3.5 long-time-, mid-time- und short-time-Audiosegmentierung 45
3.6 Darstellung der Hinge-Verlustfunktion für eine negative Instanz, Quelle: in Anlehnung an Provost et al. [89] 52
3.7 Binärer Entscheidungsbaum zur Klassifizierung eines „mid-time-Frames“ in die Klassen „Klatschen“ bzw. „Kein Klatschen“ 54
3.8 Binärer Entscheidungsbaum zur Klassifizierung eines „mid-time-Frames“ in die Klassen „Lachen“ bzw. „Kein Lachen“ 55
3.9 Graphische Benutzeroberfläche des Prototyps 57
3.10 Videoplayer zum Abspielen der Shots 58

4.1 Vergleich F1 (f1) der 3 angewandten Klassifizierungsmethoden MFCC-DTW (Versuch 1), Posterior-DTW (Versuch 2) und Kombination der beiden Verfahren (Versuch 3) für die Klasse „Ordnungsruf“, angewandt auf die drei Ground-Truth-Videos 68

Tabellenverzeichnis

3.1 Überblick Ground-Truth-Daten: Harte Schnitte (H), Überblendungen (Ü), Anzahl der Audio-Ereignisse (E), Anzahl der Frames, die den Audio-Ereignissen zugeordnet werden (F) 41
3.2 Zuordnung der Videos aus der MMI-Datenbank zu 6 Grundemotionen 43
3.3 Coding-Matrix für 4 Klassen unter Verwendung von 6 One-Against-One-SVMs 47
3.4 MFCC-Merkmale, Berechnung mit verschiedenen Parametern 48
3.5 Versuche mit unterschiedlichen Parametersettings für das Training des GMMs der Klasse „Ordnungsruf“ und BIC-Werte als Entscheidungskriterium 49
3.6 Euklidische DTW-Distanzen des „Ordnungsruf“-Trainings mit MFCC Merkmalsvektoren 49
3.7 Versuche mit unterschiedlichen Parametersettings für das Training der GMMs der Klasse „Zwischenruf“ und BIC-Werte als Entscheidungskriterium 50

4.1 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Shoterkennungsmethode, angewandt auf die Ground-Truth-Videos (Harte Schnitte (H), Überblendungen (Ü)) 60
4.2 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der implementierten Gesichtserkennung mittels Viola-Jones-Algorithmus, Harte Schnitte (H), Überblendungen (Ü), relevante Shots (R), Anzahl der relevanten Gesichter (Anz. G) 61
4.3 Test der Gesichtsverifizierung mit globalen HOG-Merkmalen und unterschiedlichen Parametern, True positive (TP), False negative (FN), Accuracy (acc) . 62
4.4 Test der Gesichtsverifizierung mit lokalen SURF-HOG-Merkmalen und unterschiedlichen Parametern, True positive (TP), False negative (FN), Accuracy (acc) 62
4.5 Ergebnisse der Gesichtsverifizierung mit globalen HOG-Merkmalen für die Gesichtsbilder der Ground-Truth-Videos (größere Trainingsdatenbank), True positive (TP), False negative (FN), Recall (r), Precision (p), F1 (f1), Accuracy (acc) 62
4.6 Ergebnisse der Gesichtsverifizierung mit globalen HOG-Merkmalen für die Gesichtsbilder der Ground-Truth-Videos (kleinere Trainingsdatenbank), True positive (TP), False negative (FN), Recall (r), Precision (p), F1 (f1), Accuracy (acc) 63

4.7 Aufteilung der MMI-Emotionsgesichtsbilder in Trainings- und Testset 63
4.8 Ground-Truth-Zuordnung der in der Gesichtserkennung gefundenen, relevanten Bilder in die 4 Emotionsklassen 63
4.9 Konfusionsmatrix der Emotionsklassifizierung, 85% Trainingsset, 15% Testset, durchschnittliche Accuracy (acc) = 53%, Ground-Truth-Zuordnung (GT) . . 64
4.10 Konfusionsmatrix der Emotionsklassifizierung, Trainingsset = Testset, durchschnittliche Accuracy (acc) = 90%, Ground-Truth-Zuordnung (GT) 64
4.11 Konfusionsmatrix der Emotionsklassifizierung, GT-Video 1, durchschnittliche Accuracy (acc) = 26%, Ground-Truth-Zuordnung (GT) 65
4.12 Konfusionsmatrix der Emotionsklassifizierung, GT-Video 2, durchschnittliche Accuracy (acc) = 24%, Ground-Truth-Zuordnung (GT) 65
4.13 Konfusionsmatrix der Emotionsklassifizierung, GT-Video 3, durchschnittliche Accuracy (acc) = 45%, Ground-Truth-Zuordnung (GT) 65
4.14 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 1, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) 66
4.15 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 2, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) 67
4.16 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 3, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) 67
4.17 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) des DTW für die Klasse „Ordnungsruf“ unter Verwendung von 4 unterschiedlichen MFCC-Merkmalsvektoren (Überblick 3.4), Versuch 4, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) 67
4.18 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der Posterior-DTW für die Klasse „Ordnungsruf“, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) . . 68
4.19 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der Kombination von MFCC-DTW und Posterior-DTW für die Klasse „Ordnungsruf“, Anzahl der Audioereignisse „Ordnungsruf“ (E), Anzahl der Frames, die der Klasse „Ordnungsruf“ zugeordnet werden (F) 68

4.20 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Entscheidungsbaummethode für die Klasse „Klatschen“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Klatschen“ (E), Anzahl der Frames, die der Klasse „Klatschen“ zugeordnet werden (F) 69
4.21 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten Entscheidungsbaummethode für die Klasse „Lachen“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Lachen“ (E), Anzahl der Frames, die der Klasse „Lachen“ zugeordnet werden (F) 69
4.22 Recall (r), Precision (p), F1 (f1) und Accuracy (acc) der gewählten GMM-Methode für die Klasse „Zwischenruf“, angewandt auf die Ground-Truth-Videos, Anzahl der Audioereignisse „Zwischenruf“ (E), Anzahl der Frames, die der Klasse „Zwischenruf“ zugeordnet werden (F) 70

Literaturverzeichnis

[1] C. Tamizharasan and S. Chandrakala, "A survey on multimodal content based video retrieval," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 1, pp. 69–76, 2013.

[2] N. Dimitrova, H.-J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," *IEEE Multimedia*, vol. 9, no. 3, pp. 42–55, 2002.

[3] S. Agarwal, A. K. Verma, and P. Singh, "Content based image retrieval using discrete wavelet transform and edge histogram descriptor," *International Conference on Information Systems and Computer Networks (ISCON)*, pp. 19–23, 2013.

[4] S. C. Sebastine, B. Thuraisingham, and B. Prabhakaran, "Semantic web for content based video retrieval," *IEEE International Conference on Semantic Computing (ICSC)*, pp. 103–108, 2009.

[5] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, 2011.

[6] A. Cavarero, "How to design the right object classes?" *Proceedings of IEEE Systems Man and Cybernetics Conference - SMC*, vol. 1, pp. 221–225, 1993.

[7] Österreichisches Parlament. Zuletzt geprüft am: 2017-10-23. [Online]. Available: http://www.parlament.gv.at/

[8] Deutscher Bundestag. Zuletzt geprüft am: 2017-10-23. [Online]. Available: http://www.bundestag.de/

[9] D. Mitrovic, M. Zeppelzauer, and C. Breiteneder, "Features for content-based audio retrieval," *Advances in Computers. Improving the Web*, vol. 1, no. 78, pp. 71–150, 2010.

[10] F. Zheng, G. Zhang, and Z. Song, "Comparison of different implementations of MFCC," *Journal of Computer Science and Technology*, vol. 16, no. 6, pp. 582–589, 2001.

[11] J. Breebaart and M. F. McKinney, "Features for audio classification," in *Algorithms In Ambient Intelligence*, ser. Philips Research, F. Toolenaar, W. F. J. Verhaegh, E. Aarts, and J. Korst, Eds. Dordrecht: Springer Netherlands, 2004, vol. 2, pp. 113–129.

[12] T. Giannakopoulos, C. Smailis, S. Perantonis, and C. Spyropoulos, "Realtime depression estimation using mid-term audio features," *Proceedings of the 3rd International Workshop on Artificial Intelligence and Assistive Medicine*, vol. 1213, pp. 41–45, 2014.

[13] T. Giannakopoulos and A. Pikrakis, *Introduction To Audio Analysis: A MATLAB Approach*. Kidlington Oxford UK: Academic Press is an imprint of Elsevier, 2014.

[14] D. Masri, Z. Aung, and W. L. Woon, "Image classification using appearance based features: Tech report."

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893, 2005.

[16] P. Antonopoulos, N. Nikolaidis, and I. Pitas, "Hierarchical face clustering using SIFT image features," *Proceedings of IEEE Symposium on Computational Intelligence in Image and Signal Processing*, pp. 325–329, 2007.

[17] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding (CVIU)*, vol. 3, no. 110, pp. 346–359, 2008.

[18] G. Du, F. Su, and A. Cai, "Face recognition using SURF features," *MIPPR, Pattern Recognition and Computer Vision*, no. 7496, 2009.

[19] S. Asha and M. Sreeraj, "Content based video retrieval using SURF descriptor," *Third International Conference on Advances in Computing and Communications*, pp. 212–215, 2013.

[20] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," *Proceedings of the 2016 International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1310–1315, 2016.

[21] M. Iqbal and Y. Zhu, "Supervised machine learning approaches - a survey," *ICTACT Journal on Soft Computing*, vol. 05, no. 03, pp. 946–952, 2015.

[22] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," *Informatica - International Journal of Computing and Informatics*, vol. 31, no. 3, pp. 249–268, 2007.

[23] L. Vuegen, B. Van Den Broeck, P. Karsmakers, J. F. Gemmeke, B. Vanrumste, and H. Van Hamme, "An MFCC-GMM approach for event detection and classification," *AASP Challenge on Detection and Classification of Acoustic Scenes and Events*, pp. 1–3, 2013.

[24] C. Hentschel and H. Sack, "Does one size really fit all? Evaluating classifiers in bag-of-visual-words classification," *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*, pp. 1–8, 2014.

[25] Y. Wang, Z. Liu, and J.-C. Huang, "Multimedia content analysis: Using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, 2000.

[26] R. Lienhart, "Comparison of automatic shot boundary detection algorithms," *Image and Video Processing VII*, vol. 3656, no. 29, pp. 290–301, 1999.

[27] A. Jacobs, A. Miene, G. T. Ioannidis, and O. Herzog, "Automatic shot boundary detection combining color, edge and motion features of adjacent frames," *TRECVID 2004 Workshop Notebook Papers*, pp. 197–206, 2004.

[28] R. Lienhart, "Reliable transition detection in videos: A survey and practitioner's guide," *International Journal of Image and Graphics (IJIG)*, vol. 1, no. 3, pp. 469–486, 2001.

[29] N. Azra and G. Shobha, "Key frame extraction from videos - a survey," *International Journal of Computer Science & Communication Networks*, vol. 3, no. 3, pp. 194–198, 2013.

[30] D. A. Lisin, M. A. Mattar, M. B. Blaschko, M. C. Benfield, and E. G. Learned-Miller, "Combining local and global image features for object class recognition," *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 3, pp. 47–54, 2005.

[31] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Computing Surveys*, vol. 46, no. 1 (10), pp. 1–53, 2013.

[32] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[33] W. Wójcik, K. Gromaszek, and M. Junisbekov, "Face recognition: Issues, methods and alternative applications," *Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods*, pp. 7–28, 2016.

[34] C. Zhang and Z. Zhang, "A survey of recent advances in face detection: Tech report."

[35] L. Masupha, T. Zuva, S. Ngwira, and O. Esan, "Face recognition techniques, their advantages, disadvantages and performance evaluation," *International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–5, 2015.

[36] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[37] H. Hatem, Z. Beiji, and R. Majeed, "A survey of feature base methods for human face detection," *International Journal of Control and Automation IJCA*, vol. 8, no. 5, pp. 61–78, 2015.

[38] M. Pukhrambam, A. Das, and A. Saha, "Facial components extraction and expression recognition in static images," *Proceedings of the International Conference on Green Computing and Internet of Things (ICGCIoT)*, pp. 726–731, 2015.

[39] J. Whitehill, M. Stewart Bartlett, and J. R. Movellan, "Automatic facial expression recognition," *Social Emotions in Nature and Artifact*, 2013.

[40] P. Ekman, *Gefühle lesen: Wie Sie Emotionen erkennen und richtig interpretieren*, 1st ed. München and Heidelberg: Elsevier, Spektrum, Akad. Verl., 2004.

[41] FACS (facial action coding system). Zuletzt geprüft am: 2017-10-23. [Online]. Available: https://www.cs.cmu.edu/~face/facs.htm

[42] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, and Z. Ambadar, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," *Computer Society Conference on Computer Vision and Pattern Recognition workshops (CVPR-W)*, pp. 94–101, 2010.

[43] P. Ekman, Ed., *What the face reveals: Basic and applied studies of spontaneous expression using the facial action coding system (FACS)*, ser. Series in affective science. New York, NY: Oxford Univ. Press, 1997.

[44] C.-H. Wu, J.-C. Lin, and W.-L. Wei, "Survey on audiovisual emotion recognition: Databases, features and data fusion strategies," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. 1–18, 2014.

[45] N. Gupta and N. Kaur, "Design and implementation of emotion recognition system by using Matlab," *International Journal of Engineering Research and Applications (IJERA)*, vol. 3, no. 4, pp. 2002–2006, 2013.

[46] N. Murali Krishna, P. V. Lakshmi, Y. Srinivas, and J. Sirisha Devi, "Emotion recognition using dynamic time warping technique for isolated words," *International Journal of Computer Science Issues (IJCSI)*, vol. 8, no. 5/1, pp. 306–309, 2011.

[47] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2895–2907, 2003.

[48] R. Mertens, H. Lei, L. Gottlieb, G. Friedland, and A. Divakaran, "Acoustic super models for large scale video event detection," *Proceedings of the joint ACM workshop on modeling and representing events*, pp. 19–24, 2011.

[49] B. Elizalde, H. Lei, G. Friedland, and N. Peters, "Capturing the acoustic scene characteristics for audio scene detection," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013.

[50] K. Sakhnov, E. Verteletskaya, and B. Simak, "Approach for energy-based voice detector with adaptive scaling factor," *International Journal of Computer Science (IJCS)*, vol. 36, no. 4, 2009.

[51] M. Jalil, F. A. Butt, and A. Malik, "Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals," *International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, pp. 208–212, 2013.

[52] G. Eisenberg and T. Sikora, *Identifikation und Klassifikation von Musikinstrumentenklängen in monophoner und polyphoner Musik: Zugl.: Berlin, Techn. Univ., Diss., 2008*, 1st ed. Göttingen: Cuvillier, 2008.

[53] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[54] A. I. Al-Shoshan, "Speech and music classification and separation: A review," *J. King Saud Univ.*, vol. 19 Eng. Sci. (1), pp. 95–133, 2006.

[55] H. Misra, S. Ikbal, H. Bourlard, and H. Hermansky, "Spectral entropy based feature for robust ASR," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. I–193–6, 2004.

[56] S. D. Ganesh and P. K. Sahu, "A study on automatic speech recognition toolkits," *International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, pp. 365–368, 2015.

[57] C. Gaida, P. Lange, R. Petrick, P. Proba, A. Malatawy, and D. Suendermann-Oeft, "Comparing open-source speech recognition toolkits: Tech report."

[58] HTK Toolkit. Zuletzt geprüft am: 2017-10-23. [Online]. Available: http://htk.eng.cam.ac.uk/

[59] CMU Sphinx Spracherkennung. Zuletzt geprüft am: 2017-10-23. [Online]. Available: https://cmusphinx.github.io/

[60] Kaldi. Zuletzt geprüft am: 2017-10-23. [Online]. Available: http://kaldi-asr.org/doc/

[61] H. Yang, C. Oehlke, and C. Meinel, "German speech recognition: A solution for the analysis and processing of lecture recordings," *10th IEEE/ACIS International Conference on Computer and Information Science*, pp. 201–206, 2011.

[62] J. Yang, Y.-G. Jiang, A. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," *Proceedings of the international Workshop on Multimedia Information Retrieval*, pp. 197–206, 2007.

[63] A. P. B. Lopes, S. E. F. de Avila, A. N. A. Peixoto, R. S. Oliveira, M. d. M. Coelho, and A. d. A. Araújo, "Nude detection in video using bag-of-visual-features," *Proceedings of the 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pp. 224–231, 2009.

[64] D. K. Srivastava and L. Bhambhu, "Data classification using support vector machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1–7, 2010.

[65] H. Bhavsar and M. H. Panchal, "A review on support vector machine for data classification," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, pp. 185–189, 2012.

[66] H.-Y. Huang and C.-J. Lin, "Linear and kernel classification: When to use which? tech report."

[67] F. F. Chamasemani and Y. P. Singh, "Multi-class support vector machine (SVM) classifiers - an application in hypothyroid detection and classification," *Proceedings of Sixth International Conference on Bio-Inspired Computing*, pp. 351–356, 2011.

[68] J. Pohjalainen, T. Raitio, and P. Alku, "Detection of shouted speech in the presence of ambient noise," *Journal of the Acoustical Society of America*, vol. 133, no. 4, pp. 1–4, 2013.

[69] P. R. Reddy, K. Rout, and K. S. R. Murty, "Query word retrieval from continuous speech using GMM posteriorgrams," *Proceedings of the International Conference on Signal Processing and Communications (SPCOM)*, pp. 1–6, 2014.

[70] A. Dufaux, L. Besacier, M. Ansorge, and F. Pellandini, "Automatic sound detection and recognition for noisy environment," *10th European Signal Processing Conference*, pp. 1–4, 2000.

[71] S. Euler, *Grundkurs Spracherkennung*, ser. Computational Intelligence. Wiesbaden: Friedr. Vieweg & Sohn Verlag | GWV Fachverlage GmbH, 2006.

[72] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," pp. 506–510.

[73] P. Lama and M. Namburu, "Speech recognition with dynamic time warping using MATLAB: Cs 525, spring 2010 – project report."

[74] S. Krig, *Computer Vision Metrics: Survey, Taxonomy, and Analysis: Kapitel 7: Ground Truth Data, Content, Metrics, and Analysis.* Apress, 2014.

[75] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.

[76] N. D. Marom, L. Rokach, and A. Shmilovici, "Using the confusion matrix for improving ensemble classifiers," *26th Convention of Electrical and Electronics Engineers in Israel (IEEEI)*, pp. 555–559, 2010.

[77] M. Sundaram and A. Mani, "Face recognition: Demystification of multifarious aspect in evaluation metrics," *Face Recognition - Semisupervised Classification, Subspace Projection and Evaluation Methods*, pp. 75–92, 2016.

[78] S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content," *Neurocomputing*, vol. 174, pp. 50–59, 2016.

[79] G. S. Nagaraja, M. S. Rajashekara, and T. S. Deepak, "Content based video retrieval using support vector machine classification," *Proceedings of the International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pp. 821–827, 2015.

[80] J. Sivic, M. Everingham, and A. Zisserman, "Who are you? – learning person specific classifiers from video," *Conference on Computer Vision & Pattern Recognition (CVPR)*, pp. 1145–1152, 2009.

[81] S.-A. Berrani, G. Manson, and P. Lechat, "A non-supervised approach for repeated sequence detection in TV broadcast streams," *Signal Processing: Image Communication*, vol. 23, no. 7, pp. 525–537, 2008.

[82] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan, "Automatic recognition of facial actions in spontaneous expressions," *Journal of Multimedia*, vol. 1, no. 6, pp. 22–35, 2006.

[83] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, "Web-based database for facial expression analysis," *International Conference on Multimedia and Expo (ICME)*, 2005.

[84] R. Dugad, K. Ratakonda, and N. Ahuja, "Robust video shot change detection," *IEEE Second Workshop on Multimedia Signal Processing*, pp. 376–381, 1998.

[85] Y. Yusoff, W. Christmas, and J. Kittler, "Video shot cut detection using adaptive thresholding," *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 362–371, 2000.

[86] M. A. Bagheri, G. A. Montazer, and S. Escalera, "Error correcting output codes for multiclass classification: Application to two image vision problems," *16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)*, pp. 508–513, 2012.

[87] A. E. Raftery, "Bayesian model selection in social research," *Sociological Methodology*, vol. 25, pp. 111–163, 1995.

[88] Dynamic Time Warping (DTW), (c) 2014, Quan Wang. Zuletzt geprüft am: 2018-01-03. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/43156-dynamic-time-warping--dtw-

[89] F. Provost and T. Fawcett, *Data Science für Unternehmen: Data Mining und datenanalytisches Denken praktisch anwenden.* mitp-Verlag, 2017.

[90] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition & Understanding, 2009.* Piscataway, NJ: IEEE, 2009, pp. 398–403.

[91] H. Eidenberger, *Fundamental media understanding*, 2nd ed. Norderstedt: Books on Demand, 2011.

[92] VLFeat. Zuletzt geprüft am: 2017-11-13. [Online]. Available: http://www.vlfeat.org/index.html

[93] VOICEBOX. Zuletzt geprüft am: 2017-11-13. [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html#random

[94] PLP, RASTA and MFCC. Zuletzt geprüft am: 2017-11-13. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/

[95] C. Cannam, C. Landone, and M. Sandler, "Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files," *Proceedings of the ACM International Conference on Multimedia*, pp. 1467–1468, 2010.

[96] Blackmagic Design: DaVinci Resolve 14. Zuletzt geprüft am: 2017-11-13. [Online]. Available: https://www.blackmagicdesign.com/at/products/davinciresolve/