FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Videosegmentierung durch Analyse audiovisueller Merkmale

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur/in

im Rahmen des Studiums

Visual Computing

eingereicht von

Christoph Fuchs, BSc
Matrikelnummer 0625267

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuer: Ao. Univ. Prof. Mag. Dr. Horst Eidenberger

Wien, 02.08.2013

(Christoph Fuchs, BSc) (Betreuer)

Erklärung

Christoph Fuchs
Tinterstrasse 36/2
A-1140 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, 02.08.2013 _____

(Christoph Fuchs, BSc)

Abstract

Due to the increasing amount of digital videos the segmentation and classification of videos is manually no longer feasible. Hence there is a need for algorithms, which are able to filter out relevant information from the video material by using suitable and significant descriptions.

This diploma thesis presents a system for classification of videos through analyses of audiovisual features. This is a complex problem on arbitrary video material because the features should be able to gather the semantic meaning of pictures and audio signals from the videos. Therefore, this thesis is limited on the scope of the video classification using scenes of one particular type of video, the Muppet Show. First, the fundamental approaches and methods for video analysis are explained in a detailed research. After a short overview over the development of the Muppet Show, a subsequently analysis of video material shows the characteristic attributes. Based on the gained knowledge significant audiovisual features and suitable classification models are presented, which are employed for the development of the prototype. Finally the quality of the classification results are evaluated using different tests. The intention is to show that visual features such as the distribution of colours as well as the segmentation of audio signals in speech, music and environmental sounds are able to capture the semantic meaning of video scenes of the Muppet Show.

Kurzfassung

Die Segmentierung und Klassifikation von Videos ist aufgrund der steigenden Anzahl von digitalen Videos manuell nicht mehr zu beherrschen. Es werden Algorithmen benötigt, die in der Lage sind aus dem Videomaterial relevante Informationen zu extrahieren, welche für aussagekräftige Beschreibungen geeignet sind.

In der vorliegenden Diplomarbeit wird ein System zur Klassifikation von Videos durch Analyse audiovisueller Merkmale vorgestellt. Ein solches Vorhaben stellt auf beliebigem Videomaterial ein komplexes Problem dar, da diese Merkmale in der Lage sein sollen, die semantische Bedeutung von Bildern und Audiosignalen aus Videos zu erfassen. Aus diesem Grund wird in dieser Arbeit der Anwendungsbereich der Videoklassifikation auf Szenen der Muppet Show beschränkt. Zunächst werden grundlegende Ansätze und Methoden zur Videoanalyse in einer umfassenden Literaturstudie erklärt. Nach einem kurzen Überblick über die Entstehung der Muppet Show, zeigt eine Analyse des Videomaterials die charakteristischen Eigenschaften auf. Basierend auf den gewonnenen Erkenntnissen werden aussagekräftige audiovisuelle Merkmale und geeignete Klassifikationsmodelle vorgestellt, die für die Entwicklung eines Prototyps herangezogen worden sind. Zuletzt wird die Qualität der Klassifikationsresultate mit Hilfe verschiedener Evaluierungstests ausgewertet. Dabei wird aufgezeigt, dass sowohl visuelle Merkmale, wie die Verteilung von Farbe, als auch die Segmentierung des Audiosignals in Musik, Sprache und Umgebungsgeräusche in der Lage sind, die semantische Bedeutung von Videoszenen aus der Muppet Show zu erfassen.

Danksagung

Ich möchte mich an dieser Stelle bei allen Personen bedanken, die mich während meines Studiums und der Erstellung dieser Diplomarbeit unterstützt haben. Besonders bedanken möchte ich mich bei meinen Eltern, ohne deren moralische und finanzielle Unterstützung meine erfolgreiche Studienzeit nicht möglich gewesen wäre.

Außerdem möchte ich Herrn Prof. Eidenberger meinen Dank aussprechen, der mich nicht nur während meiner Diplomarbeit hervorragend betreut hat, sondern mir ebenfalls den Besuch der ACM-Multimedia 2012 in Nara ermöglicht hat.

Abkürzungsverzeichnis

BoW	Bag of Words
CAC	Color Auto-Correlogram
CBVR	Content-Based Video Retrieval
CM	Color Moments
CCV	Color Coherence Vector
DCT	Diskrete Cosinus-Transformation
DFT	Diskrete Fourier-Transformation
DoG	Difference of Gaussian
GT	Ground Truth
HS	Hop Size
K-NN	K-Nearest Neighbour
LOO-CV	Leave One Out Cross Validation
MFCC	Mel-Frequency Cepstrum Coefficients
NN	Nearest Neighbour
RMS	Root Mean Square
SIFT	Scale Invariant Feature Transform
STE	Short Time Energy
SVM	Support Vector Machine
XML	Extensible Markup Language
ZCR	Zero Crossing Rate

Inhaltsverzeichnis

KAPITEL 1

Einleitung

Diese Diplomarbeit behandelt die Klassifikation von Szenen aus einzelnen Videos, basierend auf der Analyse von audiovisuellen Merkmalen. Ziel ist es aus einzelnen Szenen aussagekräftige Beschreibungen zu extrahieren, die für eine inhaltsbasierte Suche herangezogen werden können.

Eine solche Segmentierung stellt ein kaum lösbares Problem dar, vor allem wenn beliebiges Videomaterial vorliegt. In dieser Arbeit wird der Anwendungsbereich daher auf Videos der Muppet Show festgelegt. Trotz dieser Einschränkung liegt die Motivation in einer effizienten Analyse von einzelnen Sequenzen und es stellen sich dabei die Herausforderungen, die in diesem Kapitel näher erläutert werden.

1.1 Motivation

Der signifikante Zuwachs von digitalen Videos und die Möglichkeit den Zugang dieser für jedermann über Online-Plattformen oder anderen Medien zu gewähren, führen zu einer unübersehbaren Menge an Daten. Die Gründe der Zunahme liegen nicht nur in der Möglichkeit der schnellen Verbreitung, sondern auch an den fortlaufenden Entwicklungen der Datenkomprimierung, stetig sinkenden Preisen von digitalen Kameras und Speichermedien trotz steigender Speicherkapazität, sowie an der hohen Verfügbarkeit von Breitbandinternet [1]. In Abbildung 1.1 wird der Anstieg von veröffentlichten Videos auf der Online-Plattform *YouTube* innerhalb der letzten sechs Jahre visualisiert [2].

Aufgrund der rasch zunehmenden Menge an Videomaterial werden aussagekräftige Beschreibungen zur Repräsentation von Sequenzen benötigt. Eine solche Repräsentation kann auf unterschiedliche Arten realisiert werden, beispielsweise durch eine textuelle Beschreibung (*Tag*) oder einzelne Bilder. Es wird damit eine schnelle Einsicht in den Inhalt von Videos gegeben, sowie eine inhaltsbasierte Suche von einzelnen Sequenzen ermöglicht. *YouTube* gestattet das Durchsuchen des Videomaterials durch Tags, welche von den Benutzern selbst eingetragen werden können. Generell ist dieser Vorgang sowohl aufwendig als auch zeitintensiv und die Qualität der Tags ist für eine inhaltsbasierte Suche oft nicht ausreichend [3]. Anhand eines Beispiels aus

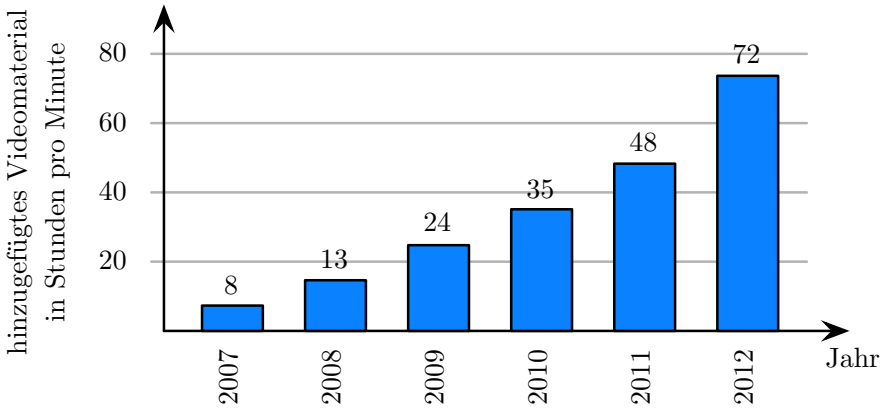


Abbildung 1.1: Hinzugefügtes Videomaterial auf *YouTube* von 2007 bis 2012 (in Anlehnung an [2]).

YouTube werden in Abbildung 1.2 manuell hinzugefügte Tags eines Videos gezeigt und deren Relevanz ausgewertet.

Aus diesem Grund werden Algorithmen benötigt, die in der Lage sind, aus dem Videomaterial relevante Informationen zu extrahieren, welche für eine aussagekräftige Beschreibung von einzelnen Sequenzen geeignet sind. Im Allgemeinen können dazu sowohl visuelle (z.B. dominante Farben, Formen von Objekten, Gesichtsdetektion etc.) als auch auditive Merkmale (z.B. *Zero Crossing Rate*, *Short Time Energy* etc.) herangezogen und analysiert werden [5, 6, 7, 8]. Je nach Anwendung und Art des Videomaterials kann sich die Zusammensetzung der tatsächlich relevanten Merkmale (*Features*) ändern, wobei diese Entscheidung Einfluss auf die Qualität des Ergebnisses und die Laufzeit der Videosegmentierung hat. Typische Einsatzgebiete von Videosegmentierung sind Video-Indexing, inhaltsbasierte Suche (*Content-Based Video Retrieval - CBVR*), Datenkomprimierung durch Entfernen von irrelevanten Informationen, sowie das Finden und Entfernen von sittenwidrigen oder moralisch bedenklichen Inhalten [9].

Abbildung 1.2: Manuell angelegte Tags eines Videos von *YouTube* und deren Relevanz [4].

In weiterer Folge beschränkt sich diese Diplomarbeit auf die inhaltsbasierte Suche in einzelnen Szenen. Eine solche Segmentierung auf uneingeschränktem Material (z. B. *YouTube*) stellt aufgrund der breiten inhaltlichen Variation ein komplexes Problem dar, weshalb der Anwendungsbereich hier auf Videos der Muppet Show festgelegt wird. Diese Vereinfachung erlaubt es, eine Analyse des Videomaterials durchzuführen, um basierend auf den charakteristischen Eigenschaften (siehe Kapitel 3) geeignete audiovisuelle Merkmale zu wählen. Trotz dieser Einschränkung können bei der Segmentierung und Klassifikation Probleme auftreten, die neue Lösungsansätze oder eine Adaption von bereits bestehenden Methoden erfordern.

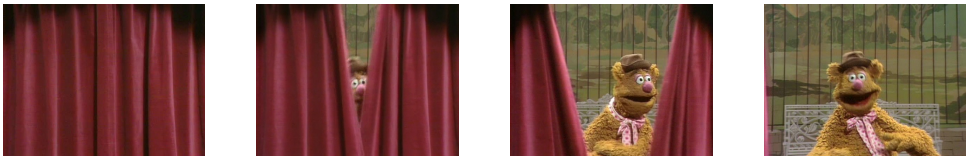
1.2 Herausforderungen

Neben den grundlegenden Problemen der Videoklassifikation, wie beispielsweise die Wahl von geeigneten Merkmalen, ist eine Klassifikation von Muppet-Szenen mit besonderen Herausforderungen verbunden, die im Folgenden näher erläutert werden:

(i) **Variationsreiche Szenengestaltung:**
Generell setzt sich die Muppet Show aus mehreren Sketchszenen zusammen. In insgesamt *120* Episoden und *9* Kinofilmen sind mehr als *800* verschiedene Muppets zu sehen, wobei die Relevanz und die Häufigkeit des Auftretens der einzelnen Charakteren sehr unterschiedlich sind [10]. Zwar besitzen in der Regel die einzelnen Figuren ein individuelles Aussehen, jedoch sind aufgrund der hohen Anzahl und der wechselnden Kostümierung einzelner Muppets zur Erkennung vor allem geeignete visuelle Merkmale gefordert. Abhängig von der Art der Szene ist nicht nur das Auftreten von spezifischen Charakteren unterschiedlich, sondern auch die Gestaltung des Bühnenbildes ist in den meisten Fällen sehr variationsreich. Diese Umstände sorgen dafür, dass eine Klassifikation von Szenen basierend auf visuellen Informationen erschwert wird.

(ii) **Übergangseffekte zwischen Szenen:**
Eine weitere charakteristische Eigenschaft der Muppet Show ist das Auftreten von verschiedenen Übergangseffekten zwischen Szenen. Bereits bestehende Algorithmen zur zeitlichen Segmentierung eines Videos sind in der Lage, einfache fortlaufende Übergänge, wie Ein-, Aus- und Überblendungen, zu detektieren. In der Muppet Show treten neben diesen auch komplexe Szenenwechsel auf. Beispiele dafür sind der aufgehende Bühnenvorhang nach Ankündigung des nächsten Sketches oder das Fokussieren einer räumlich begrenzten Szeneneinblendung durch einen Zoomeffekt (siehe Abbildung 1.3).

(iii) **Gastauftritte:**
Zusätzlich zu den einzelnen Sketchszenen enthält jede Episode der Muppet Show einen Gastauftritt. In mehreren unabhängigen Szenen unterhält der Gast entweder persönlich das Publikum oder er interagiert mit Mupptes, die ebenfalls Teil des Auftritts sind. Aufgrund dieser Verschmelzung von menschlicher Darbietung und Puppenspiel ist eine eindeutige Klassifikation der entsprechenden Szenen schwierig (siehe Abbildung 1.4). Eine weitere Herausforderung bei Gastauftritten ist, dass jeder dieser Darsteller nur innerhalb einer



(a) Szenenübergang durch öffnen des roten Vorhangs

(b) Szenenübergang durch Zoomeffekt

Abbildung 1.3: Spezielle Übergangseffekte zwischen Szenen in der Muppet Show.

Abbildung 1.4: Beispiele von Gastauftritten aus der Muppet Show.

Episode zu sehen ist. Zudem sind diese speziellen Szenen jeweils durch eine individuelle Bühnengestaltung gekennzeichnet. Diese Umstände sorgen dafür, dass nur begrenztes Material für eine Analyse zur Verfügung steht und somit die Klassifikationen solcher Szenen erschwert werden.

(iv) **Charakteristische Eigenschaften von Muppets:**
Da es sich bei Muppets um einfache Handpuppen handelt, unterliegt der Aufbau einer Szene speziellen Einschränkungen. Diese beziehen sich nicht nur auf die beinahe starre Gesichtsmimik und der eingeschränkten Bewegungsfreiheit ihrer Extremitäten, sondern auch auf die begrenzten Positionierungsmöglichkeiten innerhalb des Bühnenbildes (siehe Kapitel 3). Basierend auf diesen charakteristischen Eigenschaften können neu entwickelte oder angepasste Verfahren zur Merkmalsextraktion hilfreich für die Klassifikation von Szenen sein.

Zusammenfassend enthält die Muppet Show als Material für Videoklassifikation spezielle Eigenschaften, die nicht nur für die Merkmalsextraktion, sondern für den gesamten Ablauf der Klassifikation entscheidend sein können. Aus diesem Grund gilt es neue Lösungsansätze zu entwickeln und bereits bestehende Methode entsprechend zu adaptieren.

1.3 Aufbau der Arbeit

In Kapitel 2 wird der grundlegende Ablauf einer Videoklassifikation erklärt. Dabei werden zunächst Methoden zur zeitlichen Segmentierung und der Extraktion von aussagekräftigen Frames erläutert. Anschließend werden audiovisuelle Merkmale und entsprechende Klassifikationsmodelle vorgestellt, die für die Videoklassifikation geeignet sind. Das dritte Kapitel beschäftigt sich mit der Analyse des Videomaterials. Es wird die Entwicklungsgeschichte der Muppet Show aufgezeigt und die besonderen charakteristischen Eigenschaften beschrieben. Außerdem wird ein Überblick über jene Szenenkategorien gegeben, die für die angestrebte Videoklassifikation relevant sind. In Kapitel 4 wird ein entwickelter Prototyp zur Klassifikation von Muppet-Szenen vorgestellt, wobei die einzelnen Phasen und die Funktionsweise erklärt werden. Das fünfte Kapitel beschreibt die Qualität der Klassifikationsresultate. Es werden dabei verschiedene Evaluierungen mit mehreren Testvideos durchgeführt und analysiert. Im letzten Kapitel wird eine Schlussfolgerung, sowie ein kurzer Ausblick über weiterführende Forschungsmöglichkeiten präsentiert.

KAPITEL 2

Hintergrund

In diesem Kapitel wird der grundsätzliche Ablauf der Videoklassifikation behandelt. Dabei werden die einzelnen Schritte erklärt und auf die einzelnen Methoden näher eingegangen, die speziell für den gewählten Anwendungsbereich von Bedeutung sind.

2.1 Aufbau der Videoklassifikation

Das Ziel der automatischen Videoklassifikation ist es einzelne Szenen aufgrund ihrer Inhalte in vordefinierte Kategorien zu unterteilen. Die grundlegende Vorgangsweise basiert auf einer Abfolge einzelner Schritte (siehe Abbildung 2.1), wo hauptsächlich eine Datenreduktion vorgenommen wird, ohne dabei relevante Informationen zu verlieren.

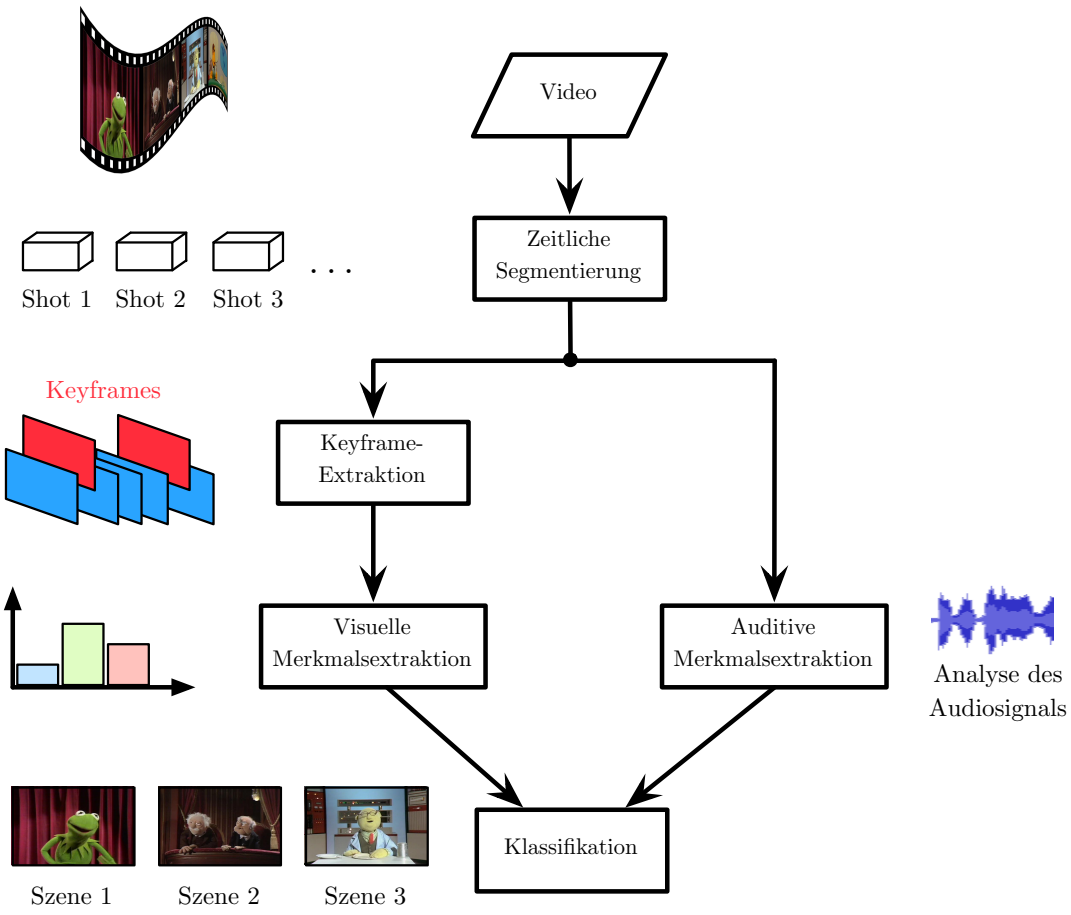


Abbildung 2.1: Grundlegende Vorgangsweise bei der Videoklassifikation.

Für gewöhnlich liegt der erste Schritt eines CBVR-Systems in der Durchführung einer zeitlichen Segmentierung (*temporal segmentation*) [11, 12, 13]. Basierend auf dem hierarchischen Aufbau von Videos, werden diese in kleine Bestandteile, sogenannte *Shots*, unterteilt, wobei sich ein Shot aus mehreren Bildern (*Frames*) aus einer Kameraperspektive zusammensetzt [14]. Anschließend gilt es, aus den einzelnen Sequenzen aussagekräftige Merkmale zu extrahieren, die in der Lage sind den Inhalt zu repräsentieren. Dabei können sowohl auditive als auch visuelle Informationen herangezogen werden.

Für die visuelle Merkmalsextraktion ist es nicht notwendig alle Frames eines Shots zu betrachten, sondern eine Analyse auf wenigen *Keyframes* ist ausreichend [15]. Videos setzen sich aus mehreren Frames pro Sekunde zusammen, wodurch in kurzer Zeit nur wenige inhaltliche Veränderungen auftreten können und somit viel Redundanz vorhanden ist. Sowohl globale Features, wie beispielsweise die Verteilung von Farbe, als auch die Auswertung von kleinen lokalen Bereichen sind in der Lage den visuellen Inhalt zu erfassen [16, 17]. Ergänzend sei noch zu erwähnen, dass abhängig vom Anwendungsbereich auch textuelle Informationen wichtige Hinweise auf den Inhalt eines Videos liefern können [18, 19]. Aufgrund der Einschränkung auf Episoden der Muppet Show und einer Analyse des Videomaterials, ist bereits im Vorhinein bekannt, dass die textuelle Merkmalsextraktion für die vorhandenen Videodaten aber nicht geeignet ist und wird somit in dieser Arbeit nicht weiter behandelt. Neben visuellen Merkmalen liefert die Analyse des Audiosignals eines Shots zusätzliche Informationen. Durch den Einsatz von auditiven Merkmalen ist eine Segmentierung von Sprache, Musik, Stille und Umgebungsgeräusche innerhalb eines Videos möglich [20, 21], die für die Beschreibung des Inhaltes von Bedeutung sein können.

Im nächsten Schritt werden die gesammelten Informationen aus den unterschiedlichen Bereichen ausgewertet und einer Klassifikation unterzogen. Mit Hilfe der extrahierten Merkmale und einer zuvor vollzogenen Trainingsphase sind geeignete Klassifikationsmodelle in der Lage, einzelne Shots in vordefinierte Kategorien zu segmentieren. Abschließend gilt es mehrere klassifizierte Shots zu einer gemeinsamen Szene zusammenzufügen. Dieser Vorgang wird als *Shot Grouping* bezeichnet [22]. Bei diesem Gruppierungsvorgang wird nicht nur die semantische Bedeutung der betroffenen Shots herangezogen, sondern auch deren zeitliches Auftreten beachtet.

Zusammenfassend setzt sich die Videoklassifikation aus mehreren Schritten zusammen. Ausgehend von einem Video wird zunächst eine zeitliche Segmentierung vorgenommen, um anschließend aus einzelnen Sequenzen audiovisuelle Merkmale zu extrahieren. Anhand dieser Informationen und geeigneter Klassifikationsmodelle erfolgt eine Klassifikation von Shots, die abschließend zu Szenen gruppiert werden. Im Folgenden werden die einzelnen Schritte der Videoklassifikation und Methoden zur Evaluierung von Klassifikationsresultaten ausführlich beschrieben.

2.2 Zeitliche Segmentierung

Der grundsätzliche Aufbau eines Videos lässt sich als eine Reihe von einzelnen Sequenzen (Shots) erklären, die anhand der semantischen Bedeutung zu einzelnen Szenen zusammengefasst werden (siehe Abbildung 2.2). Die Beschreibung eines Videos ist aufgrund des variierenden Inhaltes schwierig. Aus diesem Grund wird, basierend auf dem hierarchischen Aufbau, eine zeitliche Segmentierung vorgenommen. Ein Video wird in mehrere Shots unterteilt, wobei eine solche Sequenz aus mehrere aufeinander folgende Frames aus einer Kameraperspektive besteht [23]. Es wird angenommen, dass innerhalb dieser Shots wenige inhaltliche Veränderungen auftreten, wodurch die Beschreibung erleichtert wird [14, 24]. Abhängig von der semantischen Bedeutung bilden mehrere benachbarte Shots eine Szene. Um die benötigten Shots zu bestim-

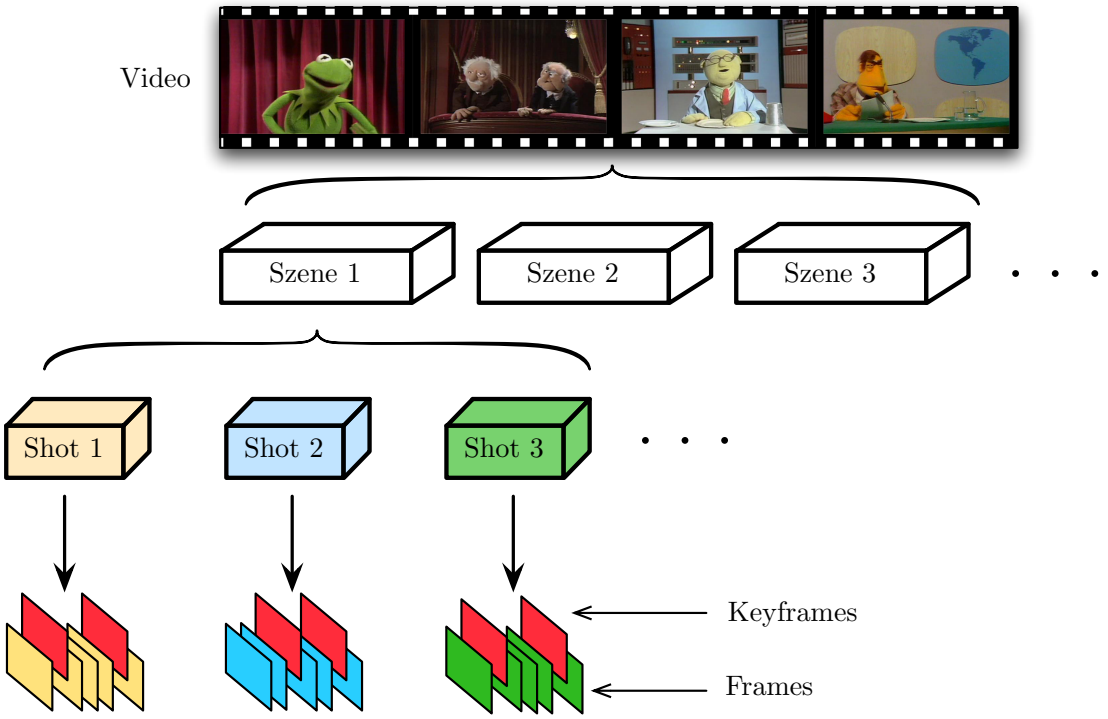


Abbildung 2.2: Hierarchischer Aufbau eines Videos.

men, gilt es die Grenzen zwischen den einzelnen Sequenzen zu detektieren, wobei verschiedene Arten von Übergängen auftreten können [25].

2.2.1 Übergangsarten zwischen Shots

Im Allgemeinen handelt es sich bei diesen Übergängen (*shot transitions*) nicht nur um visuelle Effekte, sondern es wird zusätzlicher Einfluss auf Interpretation und Verständnis von Szenen erreicht. Dabei sind folgende Übergangsarten am häufigsten in Videos vertreten:

- **Abrupter Schnitt:**
  Die meist vorkommende und einfachste Übergangsart ist ein Schnitt (*direct cut*), der in Form eines abrupten Wechsels von Shots auftritt (siehe Abbildung 2.3a). Es handelt sich hierbei um einen Übergang, wo zu jedem Zeitpunkt Frames aus einem einzigen Shot zu sehen sind und es somit zu keiner Überblendung kommt. Erst nachdem eine Sequenz vollständig beendet ist, wird der nächste Shot durch dessen ersten Frame eingeleitet [26].

- **Ein- und Ausblendung:**
  Im Gegensatz zu Schnitten verlaufen Ein- und Ausblendungen (*Fade-In / Fade-Out*) über mehrere Frames, wobei eine Überblendung zwischen einem Shot und einem Bild eingesetzt wird. Beispielsweise wird bei einer Ausblendung ein Shot allmählich durch einen

schwarzen Frame ersetzt. Üblicherweise werden solche Übergänge am Ende einer Szene eingesetzt, um die Bedeutung einer abgeschlossenen Szene zu unterstreichen (siehe Abbildung 2.3b). Einblendungen basieren auf dem gleichen Prinzip, jedoch verläuft der Übergang in entgegengesetzter Weise und besitzt einen einleitenden Effekt [27].

- **Überblendung von Shots:**
  Neben der Ein- und Ausblendung ist die Überblendung von Shots ein weiterer fortlaufender Übergang, der als *Dissolve* bezeichnet wird. Dabei wird während der Ausblendung einer Sequenz bereits die nachfolgende Sequenz eingeblendet (siehe Abbildung 2.3c) [23]. Dieser Übergang wird eingesetzt, um beispielsweise das Verstreichen von Zeit auszudrücken [27].

- **Überblendung von Shots:**
  Wipes sind ebenfalls Überblendungen von Shots. Im Gegensatz zu Fades und Dissolves werden spezielle geometrische Transformationen eingesetzt (siehe Abbildung 2.3d). Solche Übergangseffekte werden durch das Verwenden einer binären Maske erzielt, wo festgelegt wird welche Pixel von welchem Shot zu sehen sind. Typische Formen von solchen Masken sind Linien, Blöcke, ein einzelner Kreis oder komplexe Muster [28].

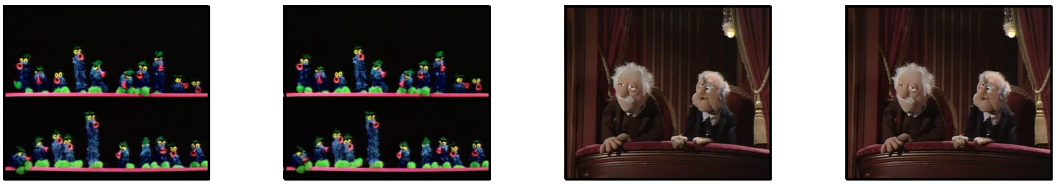
Basierend auf den Eigenschaften der einzelnen Übergangseffekte existiert bereits eine Vielzahl an Algorithmen zur Detektion von Shots, wobei entweder pixel-, regionen- oder bewegungsbasierte Informationen herangezogen werden [29, 30, 31]. Ein weit verbreiteter Ansatz zur zeitlichen Segmentierung ist *Twin Comparison*.

2.2.2 Twin Comparison

Im Allgemeinen wird bei der Twin-Comparison-Methode zwischen der Detektion von abrupten Schnitten und fortlaufenden Übergängen (Ein-, Aus- und Überblendung) unterschieden. Während Schnitte aufgrund bedeutender inhaltlicher Veränderungen zwischen zwei Frames einfach zu detektieren sind, treten bei fortlaufenden Übergängen über mehrere Frames nur geringe Änderungen auf, wodurch eine Detektion erschwert wird [14]. In Abbildung 2.4 ist das grundlegende Prinzip von Twin Comparison ersichtlich. Die Detektion der verschiedenen Übergänge erfolgt durch die Analyse von benachbarten Frames und deren Differenzen. Für diese Auswertung können sowohl einzelne Pixel, beschränkte Regionen oder globale Histogramme herangezogen werden, wobei histogrammbasierte Methoden aufgrund des guten Kompromisses zwischen Berechnungsaufwand und Genauigkeit am häufigsten verwendet werden [31].

Ausgehend von einzelnen Frames wird zunächst für jedes Bild ein Histogramm gebildet, indem die Intensitätswerte der Farbkanäle einer Quantisierung unterzogen werden. Anschließend gilt es, die Differenz von benachbarten Frames zu berechnen. In [32] sind für die Differenzbildung verschiedene Metriken getestet worden. Die sogenannte *Histogram Intersection* hat dabei am besten abgeschnitten:

$$Intersection(H_1, H_2) = \frac{\sum_i min(h_{1i}, h_{2i})}{N} \tag{2.1}$$



(a) Abrupter Schnitt (*cut*)

(b) Ausblendung (*fade-out*)

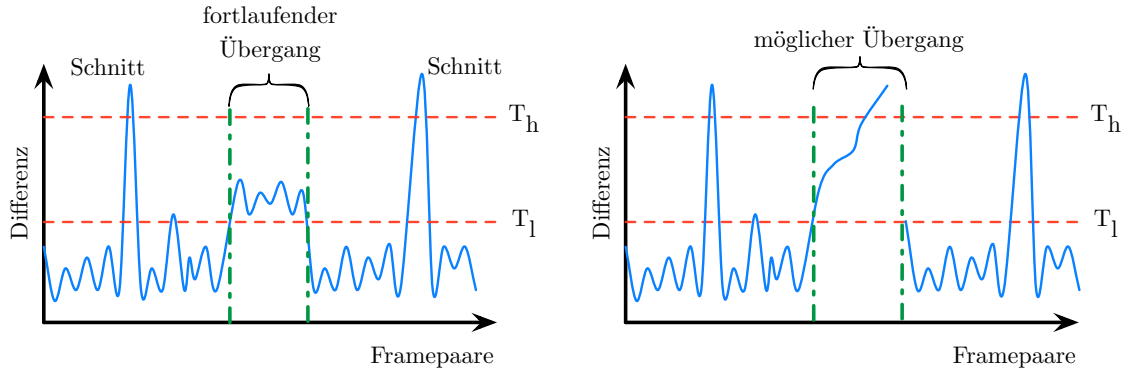
(c) Überblendung (*dissolve*)

(d) Überblendung durch komplexe Transformation (*wipe*)

Abbildung 2.3: Beispiele für verschiedene Übergangsarten zwischen Shots.

wobei h_{1i} den Farbwert i des Histogramms H_1 repräsentiert. Je größer die Histogramm-Differenz von benachbarten Frames ausfällt, desto größer ist die inhaltliche Veränderung. Basierend auf den Eigenschaften von abrupten Schnitten (siehe Abschnitt 2.2.1) und der Annahme, dass zwei verschiedene Shots unterschiedlichen Inhalt aufweisen, wird ein Schnitt detektiert, sobald die Differenz einen Schwellwert T_h übersteigt (siehe Abbildung 2.4a) [14].

Fortlaufende Übergänge können auf ähnliche Weise bestimmt werden. Aufgrund der Überlagerung von zwei benachbarten Shots fällt der inhaltliche Unterschied während des Überganges geringer aus als im Falle eines direkten Schnitts. Jedoch ist die Histogramm-Differenz von Framepaaren innerhalb eines Shots am geringsten. Deshalb wird ein zweiter Schwellwert T_l eingeführt, mit dessen Hilfe Fades, Dissolves und Wipes detektiert werden können (siehe Abbildung 2.4b). Übersteigt die Differenz von benachbarten Frames den Schwellwert T_l, so handelt es sich dabei um den möglichen Beginn eines fortlaufenden Überganges. Anschließend werden die Differenzen von nachfolgenden Framepaaren betrachtet, bis T_l unterschritten wird und das



(a) Histogramm-Differenzen von benachbarten Frames zur Detektion von Schnitten und Übergängen

(b) Detektion eines fortlaufenden Überganges

Abbildung 2.4: Prinzipielle Funktionsweise von Twin Comparison.

Ende des Übergangseffekts eintritt. Die einzelnen Differenzwerte innerhalb dieses Bereichs werden aufsummiert. Überschreitet die Summe den Schwellwert zur Schnitterkennung T_h, so wird angenommen, dass es sich dabei tatsächlich um einen fortlaufenden Übergang handelt [33].

Diese Methode zur Detektion von Shot-Grenzen wird aufgrund der beiden eingesetzten Schwellwerte als Twin Comparison bezeichnet und ist in der Lage sowohl Schnitte als auch spezielle Übergangseffekte zu detektieren. Jedoch ist die Qualität der Resultate abhängig von der Wahl von T_h und T_l [14]. In [34] werden die beiden Schwellwerte basierend auf der Verteilung der Differenzwerte von Framepaaren aus einem gesamten Video bestimmt (siehe Abbildung 2.5). Niedrige Differenzen sind am häufigsten vertreten und repräsentieren Framepaare innerhalb eines Shots, wo kaum inhaltlichen Veränderungen auftreten. Shot-Übergänge (Schnitt, Ein-, Aus- oder Überblendung) weisen höhere Differenzwerte auf und sind seltener innerhalb eines Videos vertreten. Basierend auf diesen Erkenntnissen können mit Hilfe von Mittelwert μ und

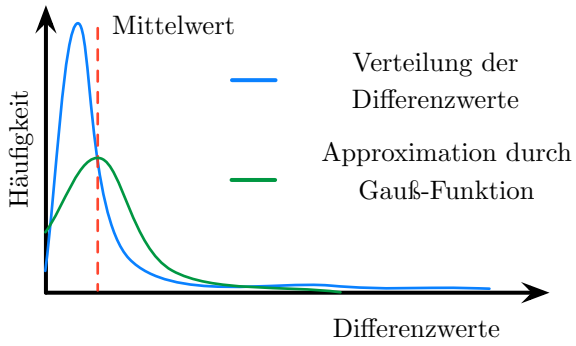


Abbildung 2.5: Verteilung von Histogramm-Differenzen zur Bestimmung der Schwellwerte (in Anlehnung an [34]).

Standardabweichung σ die beiden Schwellwerte ermittelt werden:

$$\begin{aligned} T_h &= \mu + \alpha * \sigma \\ T_l &= \mu + \beta * \sigma \qquad (\alpha > \beta) \end{aligned} \tag{2.2}$$

wobei α und β abhängig vom Inhalt des Videomaterials gewählt werden und zur Bestimmung der Höhe der Schwellwerte dienen [34].

Zusammenfassend können durch den Einsatz einer zeitlichen Segmentierung sowohl einzelne Shots aus Videos, als auch deren Übergangsgrenzen bestimmt werden. Anschließend gilt es aus den einzelnen Sequenzen audiovisuelle Merkmale zu extrahieren, wobei im Falle der visuellen Merkmalsextraktion zunächst aussagekräftige Frames, sogenannte *Keyframes*, ermittelt werden.

2.3 Keyframe-Extraktion

Einzelne Shots aus einem Video setzen sich aus einer Reihe von Frames zusammen, die häufig aus derselben Kameraperspektive aufgenommen worden sind (siehe Abschnitt 2.2) und abhängig vom Videomaterial Redundanz enthalten können. Visuelle Merkmalsextraktion auf all diesen Frames ist aufgrund der hohen Datenmenge mit einem enormen Rechenaufwand verbunden. Ziel der Keyframe-Extraktion ist es, wenige aussagekräftige Frames zu bestimmen. Dadurch wird sowohl eine Datenreduktion vorgenommen, ohne dabei wichtige visuelle Informationen zu verlieren, als auch eine strukturelle und inhaltliche Abstraktion des Videos geschaffen, die für eine inhaltsbasierte Suche verwendet werden kann [12, 35, 36]. Die einzelnen Keyframes müssen dabei gewisse Anforderungen erfüllen, die im nächsten Abschnitt näher erläutert werden.

2.3.1 Anforderungen an Keyframes

Die Qualität der extrahierten Keyframes aus einem Shot ist für visuelle Merkmalsextraktion und die anschließende Klassifikation entscheidend. Die Frames sollen in der Lage sein, sowohl Inhalt (z.B. Bühnenbild, Personen, Objekte etc.) als auch die Reihenfolge von auftretenden Ereignissen (z.B. ausgelöst durch eine Kamerabewegung) zu erfassen und zu repräsentieren [37]. Aus diesem Grund müssen bei der Extrahierung von Keyframes folgende Punkte beachtet werden:

- **Möglichst hohe Aussagekraft bei minimaler Redundanz:**
  Die wichtigste Anforderung bei der Wahl von Keyframes bezieht sich auf die Repräsentation des visuellen Inhaltes innerhalb eines gesamten Shots. Jeder einzelne Frame soll dabei sowohl möglichst aussagekräftig, als auch variationsreich sein, um eine Klassifikation der Sequenz basierend auf wenigen Bildern zu ermöglichen [38]. Zudem sollen Keyframes aus einem Shot möglichst wenig Ähnlichkeit zueinander aufweisen. Eine visuelle Merkmalsextraktion von ähnlichen Bildern liefert kaum zusätzliche entscheidende Informationen für eine erfolgreiche Klassifikation, sondern die redundanten Daten führen zu erhöhtem Rechenaufwand.

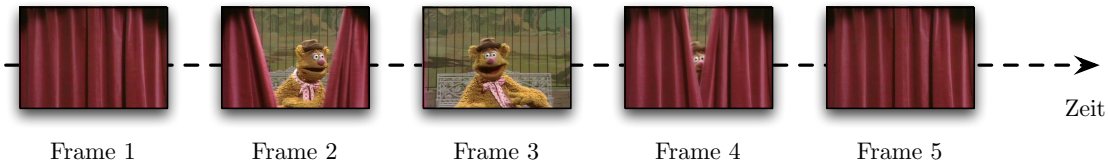
- **Anzahl der Keyframes:**
  Neben der bisherigen Anforderung, die sich vor allem auf den visuellen Inhalt bezieht, ist die Anzahl der Keyframes entscheidend, um eine geeignete Repräsentation zu erreichen. Finden innerhalb eines Shots keine signifikanten Änderungen oder Bewegungen statt, so ist ein einziger Keyframe ausreichend. Durch den Einsatz von Kamera- oder Objektbewegungen können inhaltliche Veränderungen auftreten, wie beispielsweise das Erscheinen oder Verschwinden von Objekten. Es ist erforderlich eine angepasste Anzahl von Keyframes zu extrahieren, um Situationen mit unterschiedlichen Inhalten festzuhalten [35, 38]. Jedoch sei zu beachten, dass bei einer zu großen Anzahl an Keyframes wiederum Redundanz erzeugt wird.

Zusammenfassend soll eine geeignete Menge an aussagekräftigen Keyframes in der Lage sein, den Inhalt innerhalb eines Shots zu repräsentieren und zugleich die Redundanz der visuellen Daten zu minimieren [37].

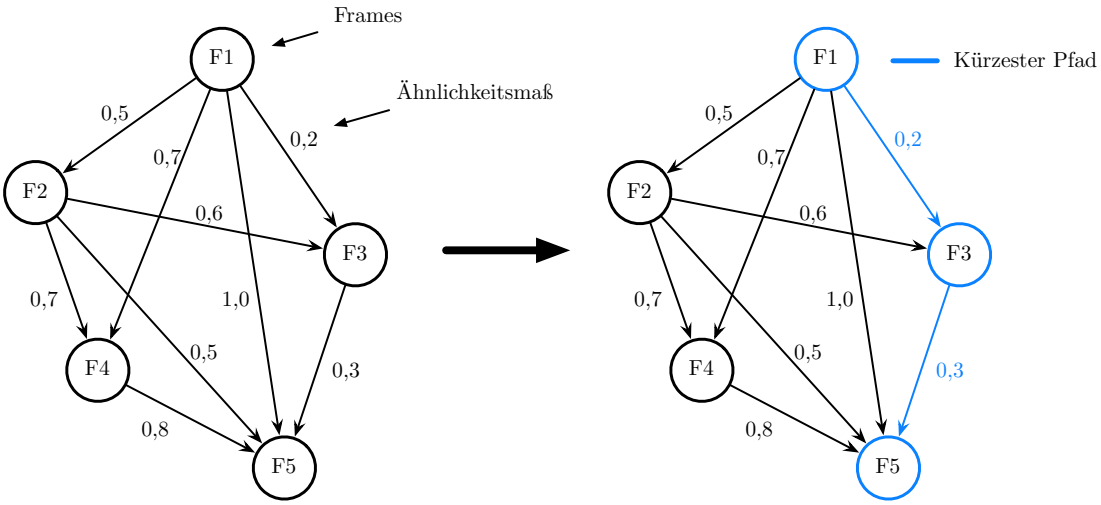
2.3.2 Methoden zur Extraktion von Keyframes

Es existiert eine Vielzahl von Algorithmen zur Extraktion von Keyframes, die sich vor allem in ihrer Komplexität und der Qualität der Resultate unterscheiden. Simple Methoden, wie beispielsweise die Verwendung des ersten oder mittleren Frames eines Shots als Keyframes [12], sind einfach zu realisieren. Jedoch besteht die Möglichkeit, dass die ermittelten Frames nur wenige visuelle Informationen enthalten und somit keine inhaltliche Repräsentation gegeben ist. Aus diesem Grund ist die Detektion von Keyframes basierend auf der Analyse von vorhandener Bewegung oder Farbverteilung von einzelnen Bildern innerhalb eines Shots sinnvoll [13, 39].

In [38] wird zunächst der mittlere Frame einer Sequenz als erster Keyframe angenommen. Anschließend werden, ähnlich wie bei der zeitlichen Segmentierung (siehe Abschnitt 2.2.2), mit Hilfe der Farbverteilung und Berechnung der Histogram Intersection die Ähnlichkeit der restlichen Bildern zum bereits festgelegtem Keyframe berechnet. Unterschreitet das Ähnlichkeitsmaß einen zuvor bestimmten Schwellwert, so handelt es sich dabei um eine auftretende inhaltliche Veränderung und ein neuer Keyframe wird an dieser Stelle hinzugefügt.
Porter u. a. beschreiben eine weitere Möglichkeit zur Bestimmung, an welchen Stellen zusätzliche Keyframes sinnvoll sind [37]. Ähnlich wie bereits zuvor beschrieben, wird zunächst für alle Framepaare innerhalb eines Shots überprüft, ob eine inhaltliche Veränderung vorliegt. Dazu wird anstelle einzelner Farbverteilungen eine Analyse von Bewegungsinformationen herangezogen. Je größer die auftretende Bewegung ausfällt, desto wahrscheinlicher ändert sich der Inhalt und ein weiterer Keyframe wird benötigt. Um diese Entscheidung zu erleichtern wird für jeden Shot ein gerichteter Graph $G = (V, E)$ konstruiert (siehe Abbildung 2.6), der sich aus den einzelnen Frames als Knoten V und deren normierten Ähnlichkeitswerten als zugehörige Gewichte der Kanten E zusammensetzt [37]. Werden zwei identische Frames betrachtet, so weisen diese eine hohe Ähnlichkeit und eine entsprechende Kantengewichtung von *1* auf. Im Gegensatz dazu bewirken zwei Frames mit unterschiedlichem Inhalt ein niedriges Kantengewicht (siehe Abbildung 2.6b). Die gesuchten Keyframes werden durch jene Knoten v_i im Graph G repräsentiert, die Teil des kürzesten Pfades $P(1, n)_{min} = \{v_1, \ldots, v_n\}$ sind, der sich vom ersten (v_1) bis zum



(a) Einzelne Frames aus einem Shot



(b) Resultierender Graph basierend auf Ähnlichkeit von Frames

Abbildung 2.6: Graphenbasierter Ansatz zur Detektion von Keyframes.

letzten Frame (v_n) eines Shots erstreckt:

$$\begin{aligned} d(v_1, v_n) &= \sum_{1 \leq i \leq n-1} w(i, i+1) \\ P(1,n)_{min} &= min\{d(v_1, v_n)\} \end{aligned} \tag{2.3}$$

Es gilt also jenen Pfad von Knoten v_i zu bestimmen, dessen Summe der zugehörigen Kantengewichte $w(i, i+1)$ minimal ist und somit möglichst unterschiedliche Keyframes enthält [37]. Basierend auf den Eigenschaften des Graphen, dass dieser nur positive Kantengewichte und keine Zyklen enthält, existieren bereits einige effiziente Verfahren zur Ermittlung des kürzesten Pfades [40], wie beispielsweise der Algorithmus von Dijkstra [41].

Die Detektion von Keyframes ist entscheidend für die visuelle Merkmalsextraktion. Grundsätzlich gilt es eine entsprechende Anzahl an Frames zu bestimmen, die in der Lage sind, die semantische Bedeutung eines Shots visuell festzuhalten ohne dabei Redundanz zu erzeugen. Anschließend werden aus diesen Frames verschiedene visuelle Merkmale berechnet, die für eine Klassifikation geeignet sind.

2.4 Visuelle Merkmalsextraktion

In diesem Abschnitt werden verschiedene Methoden zur Extraktion von visuellen Merkmalen vorgestellt, die vor allem für die Analyse unseres Videomaterials geeignet sind. Ziel ist es, basierend auf visuellen Informationen eine aussagekräftige Beschreibung von einzelnen Keyframes zu erhalten. Sowohl globale Features, wie die Analyse von Farbverteilungen, als auch die Auswertung von kleinen lokalen Bereichen ermöglichen eine solche Erfassung. Im Folgenden werden zunächst mögliche Vorverarbeitungsschritte von visuellen Signalen aufgezeigt und anschließend einzelne visuelle Features näher erklärt.

2.4.1 Analyse visueller Signale

Bevor die einzelnen Bilder für die visuelle Merkmalsextraktion herangezogen werden, kann der Einsatz von Vorverarbeitungsschritten entscheidend für die effiziente Analyse sein [42, 43]. Diese Vorgangsweise ist sowohl für Videos als auch für einzelne Bilder sinnvoll, wobei abhängig vom Inhalt folgende Punkte zu berücksichtigen sind:

- **Räumliche Quantisierung:**
  Als räumliche Quantisierung wird die Verwendung einer geringeren Auflösung des Video- oder Bildmaterials bezeichnet [43]. Durch die Verringerung der Bildgröße werden weniger Ressourcen (Speicherplatz, Rechenzeit etc.) benötig. Die damit verbundene Datenreduktion muss nicht zwingend ein Nachteil sein. Beispielsweise ermöglicht die räumliche Quantisierung eine schnellere Berechnung der Farbverteilung [44].

- **Wahl eines geeigneten Farbraums:**
  Abhängig von der Art der Anwendung und dem Ziel der Bildanalyse kann eine Transformation in einen anderen Farbraum hilfreich sein. Neben RGB existieren verschiedene Farbmodelle (HSV, YCbCr, L*a*b etc.), die aufgrund unterschiedlichen charakteristischer Eigenschaften für gewisse Anwendungsbereiche hilfreich sein können [45]. Die Wahl des Farbraumes hat keinen Einfluss auf die Funktionsweise der Methoden zur visuellen Merkmalsextraktion. Jedoch wird an dieser Stelle darauf hingewiesen, dass in den folgenden Abschnitten immer der RGB-Farbraum verwendet wird.

- **Entfernen von Bildrauschen:**
  Als Bildrauschen werden Störungen innerhalb eines Bildes bezeichnet, die nicht Teil des tatsächlichen Inhaltes sind. Davon können größere Regionen oder vereinzelte Pixel betroffen sein. Eine visuelle Auswertung des Bildinhaltes wird durch dieses Rauschen erschwert. Aus diesem Grund ist es üblich, vor der visuellen Analyse das Bild zu filtern und zu glätten, um das Rauschen möglichst zu entfernen [46]. Jedoch sei zu beachten, dass bei einem solchen Filtervorgang nicht nur Rauschunterdrückung erreicht wird, sondern auch Details verloren gehen, wie zum Beispiel die Schärfe von Kanten [45].

- **Segmentieren von einzelnen Objekten:**
  Ein weiterer Vorverarbeitungsschritt ist die Segmentierung einzelner Objekte oder Regionen aus einem Bild, die für die anschließende visuelle Merkmalsextraktion geeignet

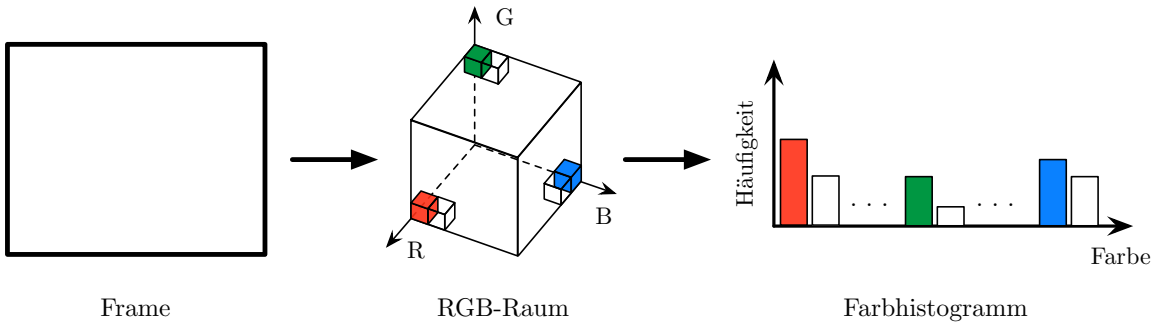


Abbildung 2.7: Die Verteilung von Farbe innerhalb eines Bildes dargestellt durch ein Farbhistogramm.

sind [43]. Jedoch können vor allem bei Videos aufgrund des dynamischen Geschehens immer wieder perspektivische Verzerrungen und Verdeckungen auftreten, wodurch sich ein Informationsverlust ergeben kann.

Nachdem die einzelnen Frames einer Vorverarbeitung unterzogen worden sind, gilt es geeignete visuelle Merkmale zu extrahieren, um eine Beschreibung des Inhaltes zu erlangen.

2.4.2 Farbhistogramme und Farbmomente

Die Wahrnehmung von Farbe erleichtert uns Menschen das Erkennen von Objekten in Bildern und das Verstehen von Videoszenen. Es ist daher naheliegend, dass in der Praxis für eine inhaltsbasierte Beschreibung die Analyse und Auswertung von Farbe essenziel ist. Ein häufig verwendetes Farbfeature ist das Farbhistogramm (*color histogram*), welches zur Repräsentation der Farbverteilung dient [11, 47]. Für die Bildung eines solchen Histogramms wird zunächst der dreidimensionale RGB-Farbraum in kleine Blöcke unterteilt, wobei jeder Block einem sogenannten *Bin* im Farbhistogramm entspricht. Anschließend wird jeder Pixel eines Frames aufgrund des Farbwertes dem entsprechenden Block im Farbraum zugeordnet und ausgewertet (siehe Abbildung 2.7). Das resultierende Farbhistogramm $H(I)$ ist ein N-dimensionaler Vektor und jeder Eintrag repräsentiert die Anzahl der Pixel eines Farbwertes [44]:

$$H(I) = \{h(I, j); i = 1, 2, \ldots, N\} \tag{2.4}$$

wobei N die Anzahl der Farben (Bins) und $h(I, j)$ die Anzahl der Pixel aus dem Frame I mit Farbe j entsprechen.

Die Vorteile des Farbhistogramms liegen in der Invarianz bezüglich Rotationen und Translationen des Ausgangsbildes [42]. Jedoch führen auftretende Beleuchtungsänderungen und die Unterteilung des Farbraumes zu Quantisierungsfehlern. Einzelne Pixel werden aufgrund minimaler Änderung des Farbwertes einem angrenzenden Bin zugeordnet, wodurch eine signifikante Änderung des Farbhistogramms entstehen kann. Somit ist eine optimale Quantisierung entscheidend für die Qualität dieses Farbfeatures [48]. Eine Alternative zu Farbhistogrammen und dem damit verbundenen Quantisierungsproblem liegt in der Verwendung von Farbmomenten *(color*

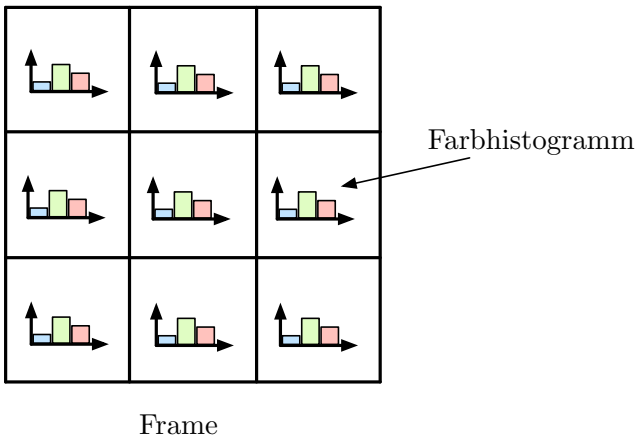


Abbildung 2.8: Aufteilung des Bildes in lokale Bereiche um Verteilung von Farbe mit räumlichen Informationen zu kombinieren.

moments – CM). Die ersten drei Momente, Mittelwert μ, Varianz σ^2 und Schiefe s, sind ebenfalls in der Lage die Verteilung von Farbwerten zu repräsentieren, wie bereits aus der Statistik bekannt ist. Für die simple Berechnung dieser Farbfeatures werden die Intensitätswerte $p_{i,j}$ der Pixel i und der einzelnen Farbkanäle j betrachtet:

$$\begin{aligned} \mu_j &= \frac{1}{n}\sum_{i=1}^{n} p_{i,j} \\ \sigma_j^2 &= \frac{1}{n}\sum_{i=1}^{n} (p_{i,j} - \mu_j)^2 \\ s_j &= \frac{1}{n}\sum_{i=1}^{n} (p_{i,j} - \mu_j)^3 \end{aligned} \tag{2.5}$$

wobei n der Anzahl der Pixel des Bildes entspricht [11, 48].

Ein weiterer Nachteil von Farbhistogrammen, aber auch von Farbmomenten, ist der Verlust von räumlichen Informationen aufgrund der globalen Betrachtung des Ausgangsbildes. Durch verschiedene Erweiterungen von Farbhistogrammen ist es möglich zusätzliche Informationen in die Verteilung von Farbe einfließen zu lassen. Beispielsweise ermöglicht eine Unterteilung des Bildes in mehrere Regionen und deren Analyse eine lokale Auswertung von Farbinformationen (siehe Abbildung 2.8) [11, 49]. Neben Farbhistogrammen und Farbmomente werden häufig weitere Features wie *Color Coherence Vectors (CCV)* und *Color Auto-Correlograms (CAC)* eingesetzt, die ebenfalls in der Lage sind, Farbe und räumliche Informationen kombiniert zu repräsentieren.

2.4.3 Color Coherence Vector

Wie bereits erläutert, beschreiben globale Farbhistogramme zwar die Verteilung von Farbe, jedoch werden räumliche Informationen nicht beachtet. Ohne diese räumlichen Informationen ist eine inhaltliche Analyse eines Bildes nicht immer eindeutig, da zwei Bilder mit unterschiedlichem Inhalt leicht ähnliche Farbverteilungen aufweisen können. In Abbildung 2.9 ist ein einfaches Beispiel zu sehen. Beide Bilder werden durch dasselbe Farbhistogramm repräsentiert. Der prozentuale Anteil von Rot ist identisch. jedoch sind im rechten Bild die entsprechenden Regionen verstreut, während im linken Bild die roten Pixel ein zusammenhängendes Objekt bilden.

Durch eine Verfeinerungen des Histogramms (*histogram refinement*) können weitere Informationen hinzugefügt werden, wie zum Beispiel Orientierung, relative Helligkeit oder die Distanz zur nächsten Kante [50]. Im Falle von CCV wird ausgehend von einem Farbhistogramm für jeden Bin überprüft, wie viele Pixel eines bestimmten Farbwerts Teil einer zusammenhängenden (kohärenten) Region sind. Eine Menge von Pixeln wird als zusammenhängende Komponente C bezeichnet, wenn zwischen zwei beliebigen Pixel p und $p‘$ $(p, p‘ \in C)$ ein Pfad $p = p_1, p_2, p_3, \ldots, p_n = p‘$ existiert, der ebenfalls vollständig Teil der Komponente C ist [51]. Ausgehend von einem Pixel p_i werden für die Bildung des Pfades alle Nachbarpixel betrachtet. Zusätzlich wird eine Region erst dann als zusammenhängend bezeichnet, wenn sie eine Mindestgröße aufweist und somit eine gewisse Anzahl an Pixeln enthält (siehe Abbildung 2.10). Der daraus resultierende Color Coherence Vector liegt in folgender Form vor:

$$ccv = \{(\alpha_1, \beta_1), \ldots, (\alpha_j, \beta_j)\} \tag{2.6}$$

wobei für jeden der j Farbwerte des Histogramms der Anteil von kohärenten Pixel α_j und nichtkohärenten Pixel β_j bestimmt wird [50]. Durch diese Erweiterung des simplen Farbhistogramms werden zusätzlich räumliche Informationen hinzugefügt, wodurch eine bessere Repräsentation der Farbverteilung erreicht wird [52]. Jedoch ist CCV als Farbfeature nicht immer ausreichend. Wie in Abbildung 2.11 zu sehen ist, können zwei Bilder mit unterschiedlichem Inhalt sowohl identische Farbhistogramme als auch die gleiche Anzahl kohärenter Pixel aufweisen. In beiden Bildern sind alle Farbwerte gleich oft vertreten und alle Pixel werden als kohärent eingestuft (siehe Abbildung 2.11). Aus diesem Grund werden weitere Erweiterungen des Farbhistogramms benötigt, wie unten anhand des Color Auto-Correlograms erklärt wird.

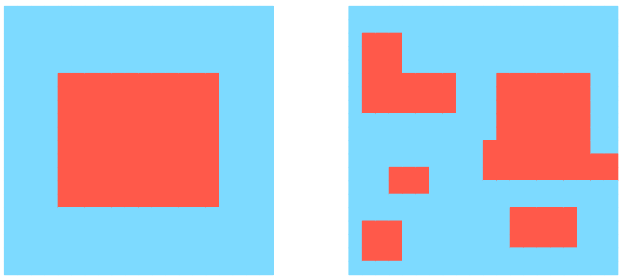


Abbildung 2.9: Zwei Bilder mit unterschiedlichem Inhalt können identische Farbverteilungen aufweisen.

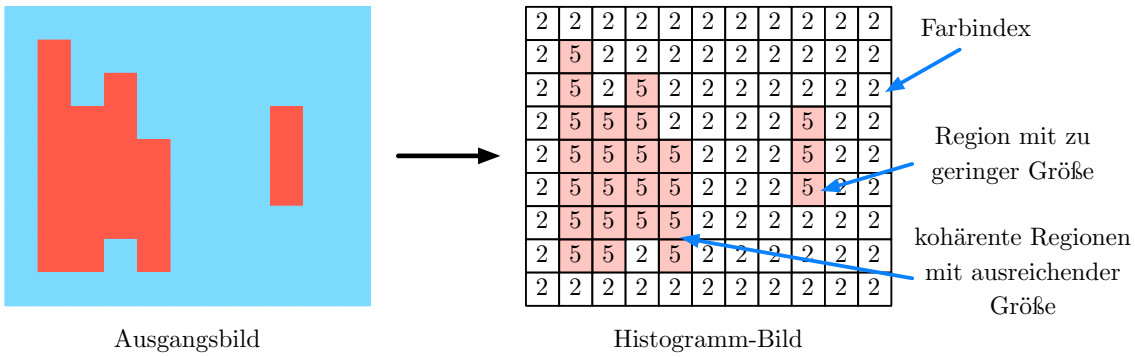


Abbildung 2.10: Bilden von zusammenhängenden Regionen basierend auf Farbwerten.

2.4.4 Color Auto-Correlogram

Ähnlich wie bei CCV werden bei der Bildung des *Color Auto-Correlograms (CAC)* die Verteilung von Farbe und räumliche Informationen miteinander kombiniert. Das CAC beschreibt die räumliche Korrelation der Farbwerte abhängig von der Distanz [53]. Ausgehend von einem Bild I wird zunächst der Farbraum quantisiert und ein Farbhistogramm mit m Bins $b_1, \ldots, b_m$ berechnet. Anschließend wird jeder Pixel eines jeden Farbwertes b_i herangezogen und dessen Umfeld analysiert. In Abbildung 2.12 wird dieser Vorgang illustriert. Startend von einem Pixel (rot) wird dessen Umfeld (grün) betrachtet, welches sich aus jenen Pixeln zusammensetzt, die eine gewisse Entfernung d aufweisen [54]. Zuletzt wir der prozentuale Anteil $\alpha_b^{(d)}$ der Übereinstimmung zwischen den Bins b_i der umliegenden Pixel und des zentralen Pixels ermittelt:

$$\alpha_b^{(d)}{}_i = \underset{p_1 \in b_i}{P} [p_2 \in b_i \mid \mid p_1 - p_2 \mid = d] \tag{2.7}$$

Für die Berechnung der Entfernung d zwischen zwei Pixeln p_1 und p_2 wird die L1-Norm herangezogen:

$$d = \mid p_1 - p_2 \mid \tag{2.8}$$

Die Ergebnisse der Analyse aller Pixel werden in einem Vektor abgespeichert, dessen Länge

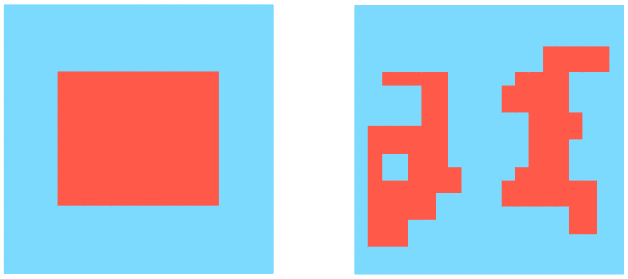


Abbildung 2.11: Zwei Bilder mit unterschiedlichem Inhalt können identische Farbhistogramme und Color Coherence Vectors aufweisen.

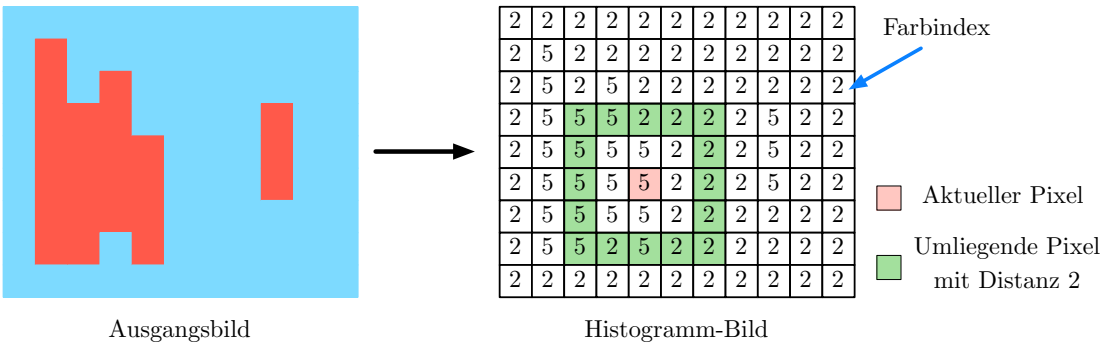


Abbildung 2.12: Analysieren der umliegenden Pixel mit der Entfernung 2 zur Bildung eines CACs.

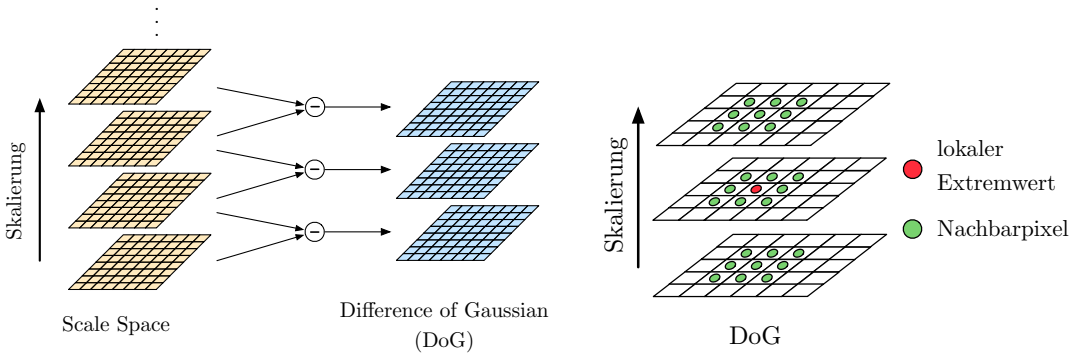
gleich der Anzahl der Bins ist. Jeder Eintrag in diesem CAC-Vektor repräsentiert die Wahrscheinlichkeit, dass ausgehend von einem Pixel, ein umliegender Pixel mit einer gewissen Entfernung erreicht werden kann und diese dem gleichem Bin angehören. Entscheidend für die Aussagekraft des CACs ist die Wahl der Distanz. So wird bei der Analyse von Bildern ein niedriger Distanzwert bevorzugt, da eine lokale Korrelation von Farbwerten signifikanter ist, als eine globale Auswertung [54].

2.4.5 SIFT-Features

Neben globalen Farbfeatures, wo der gesamte Frame ausgewertet wird, haben sich in den letzten Jahren lokale Features etabliert. Aufgrund der bereits aufgezeigten Probleme bei der Detektion von Objekten (siehe Abschnitt 2.4.1) werden Methoden benötigt, die eine translations-, rotations- und skalierungsinvariante Beschreibung erzeugen können. Im Allgemeinen werden dabei kleine informationsreiche Bereiche gesucht und mit Hilfe von Deskriptoren analysiert [55, 56]. In der Praxis werden häufig sogenannte *SIFT-Features (Scale Invariant Feature Transform)* eigensetzt, die im Vergleich zu anderen Deskriptoren eine hohe Effektivität aufweisen [57]. Die Berechnung dieser lokalen Merkmale setzt sich aus zwei grundlegenden Schritten zusammen, die anschließend näher erklärt werden [14, 58]:

1. **Detektieren von interessanten Punkten:**
   - Detektieren von Extremwerten in einem Skalierungsraum (*Scale Space*)
   - Lokalisieren von interessanten Punkten (*Keypoints*)
2. **Beschreibung der interessanten Punkten:**
   - Ermitteln der Hauptorientierung der lokalen Umgebung eines Keypoints
   - Beschreiben der Nachbarschaft von interessanten Punkten

Für die Berechnung der SIFT-Features werden aussagekräftige Punkte in einem Bild gesucht. Eine solche Detektion soll möglichst unabhängig von der Skalierung sein. Aus diesem



(a) Adjazente Schichten des Skalierungsraums werden subtrahiert zur Bildung des DoGs

(b) Lokalisieren von Keypoints durch Finden von Extremwerten über benachbarte Skalierungsschichten

Abbildung 2.13: Aufbau von Difference of Gaussian (DoG) und Detektion von Keypoints (in Anlehnung an [60]).

Grund wird im ersten Schritt ein Skalierungsraum $L(x, y, \sigma)$ aufgebaut, dessen einzelnen Skalierungsstufen sich durch eine wiederholte Faltung von Ausgangsbild $I(x, y)$ und einer Gauß-Funktion $G(x, y, \sigma)$ mit variierender Größe definieren lässt [58]:

$$\begin{aligned} L(x, y, \sigma) &= G(x, y, \sigma) * I(x, y) \\ G(x, y, \sigma) &= \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \end{aligned} \tag{2.9}$$

wobei es sich bei $*$ um den Faltungsoperator handelt und σ die Größe des Gaußfilters bestimmt. Innerhalb dieses Skalierungsraums gilt es aussagekräftige Punkte zu finden. Für diesen Lokalisierungsprozess können verschiedene Methoden eingesetzt werden, wie beispielsweise *Laplacian of Gaussian* oder *Harris Detector*. In [59] betrachtet Lowe benachbarte Schichten aus dem Skalierungsraum und berechnet deren Differenzen (siehe Abbildung 2.13a). Dieser Vorgang wird als *Difference of Gaussian (DoG)* bezeichnet und ist folgendermaßen definiert:

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{2.10}$$

$D(x, y, \sigma)$ entspricht der Differenz von zwei benachbarten Schichten aus dem Skalierungsraum und k repräsentiert die wiederholte Faltung von Ausgangsbild $I(x, y)$ mit der Gauß-Funktion $G(x, y, \sigma)$. Anschließend werden in den Schichten des DoGs lokale Extremwerte gesucht, die als mögliche Keypoints in Frage kommen. Dabei wird für jeden Pixel das Umfeld analysiert, welches sich aus den acht umliegenden Pixel und den jeweils neun Nachbarpixel aus der über- und untergeordneten Schicht zusammensetzt (siehe Abbildung 2.13b) [60]. Die detektierten lokalen Extremwerte repräsentieren die gesuchten Keypoints, nachdem ihre Qualität ausgewertet worden ist. Es werden jene Punkte verworfen, die entweder einen geringen Kontrast aufweisen oder sich entlang einer Kante befinden. Durch diesen Vorgang werden schließlich möglichst

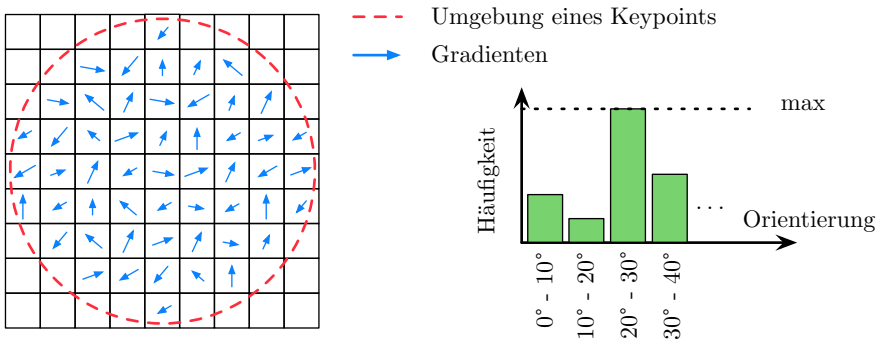


Abbildung 2.14: Berechnen der Hauptorientierung eines Keypoints basierend auf der Ausrichtung der umliegenden Gradienten.

aussagekräftige Punkte detektiert [14], deren lokale Umfelder im nächsten Schritt analysiert werden.

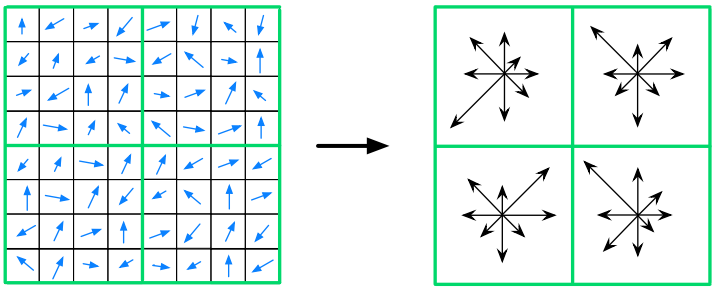
Jeder Keypoint $p = (x, y, \sigma)$ wird nicht nur durch seine Lokalisierung (x, y) sondern auch durch jene Skalierungsstufe σ definiert, wo der entsprechende Extremwert mit Hilfe von DoG detektiert worden ist. Ziel ist es, eine rotations- und skalierungsinvariante Beschreibung des lokalen Bereichs eines Keypoints zu ermitteln:

- **Skalierungsinvarianz:**
  Für eine skalierungsinvariante Analyse des Umfeldes wird nicht das Ausgangsbild betrachtet, sondern abhängig von der Skalierungsstufe σ das nächstgelegene Bild aus dem Skalierungsraum herangezogen [58].

- **Rotationsinvarianz:**
  Um eine rotationsunabhängige Analyse des Umfeldes zu ermöglichen, wird die Hauptorientierung von einzelnen Keypoints berechnet. Dazu werden die Orientierungen von Gradienten aus der umliegenden Region eines Keypoints ausgewertet und in einem Histogramm erfasst (siehe Abbildung 2.14). Die Berechnung des Gradienten für einen Pixel $p_{(i,j)}$ erfolgt durch Pixeldifferenzen [59]:

$$\begin{aligned} M_{(i,j)} &= \sqrt{(p_{(i,j)} - p_{(i+1,j)})^2 + (p_{(i,j)} - p_{(i,j+1)})^2} \\ R_{(i,j)} &= atan2(p_{(i,j)} - p_{(i+1,j)} \; ; \; p_{(i,j+1)} - p_{(i,j)}) \end{aligned} \tag{2.11}$$

  wobei $M_{(i,j)}$ der Länge und $R_{(i,j)}$ der Orientierung des Gradienten entspricht. Das Orientierungshistogramm besteht insgesamt aus *36* Quantisierungsschritten, wobei ein Abschnitt (Bin) jeweils *10* Grad entspricht. Abhängig von der Orientierung werden die einzelnen Gradienten dem entsprechenden Bin zugeordnet. Ein Bin repräsentiert dabei nicht die Anzahl der zugeordneten Gradienten, sondern es wird deren Länge aufsummiert, die zusätzlich mit einer Gauß-Funktion gewichtet werden. Je weiter ein Gradient vom Zentrum, also dem aktuellen Keypoint, entfernt ist, desto geringer fällt die Gewichtung und



Lokaler Bereich eines Keypoints

Beschreibung des lokalen Umfeldes

Abbildung 2.15: Die lokale Umgebung eines SIFT-Keypoints wird durch Orientierungshistogramme beschrieben.

somit sein Einfluss auf die Hauptorientierung aus [60]. Für die Bestimmung der Orientierung des Keypoints werden alle Bins betrachtet, die größer als *80%* des globalen Maximums sind. Aus diesem Grund ist es möglich, dass ein Keypoint mehrere Hauptorientierungen besitzen kann [14], die für die weiteren Schritte unabhängig von einander betrachtet werden.

Nachdem die Skalierung und die Hauptorientierung für einen Keypoint bestimmt worden sind, kann ein entsprechend skaliertes und rotiertes Umfeld für eine aussagekräftige Beschreibung herangezogen werden (siehe Abbildung 2.15) [59]. Ähnlich wie bei der Berechnung der Hauptorientierung von Keypoints werden die Gradienten des transformierten Umfeldes betrachtet. Nach der Unterteilung des Umfeldes in *4x4* Zellen, wird für jede Zelle ein Orientierungshistogramm berechnet. Jedoch werden nur acht Orientierungsrichtungen unterschieden. Insgesamt ergibt sich für jeden Keypoint ein *128*-dimensionaler Feature-Vektor, der sich aus *4x4* Zellen und je acht Orientierungsrichtungen zusammensetzt [14]. Abschließend werden Auswirkungen von möglichen Beleuchtungsänderungen verringert, indem der Einfluss von Gradienten mit großer Länge durch einen Schwellwert reduziert und der Feature-Vektor einer Normalisierung unterzogen wird [57].

Einfache Low-Level Features, wie die Verteilung von Farbe, als auch Features mit einer höheren semantischen Bedeutung, beispielsweise die Extraktion von lokalen Merkmalen, dienen der aussagekräftigen Beschreibung einzelner Frames aus einem Shot. Für eine erfolgreiche Segmentierung von Szenen sind visuelle Merkmale nicht immer ausreichend. Aus diesem Grund wird in dieser Arbeit zusätzlich eine Analyse des Audiosignals einer Videosequenz vorgenommen.

2.5 Auditive Merkmalsextraktion

Neben der Extraktion visueller Merkmale sind auch aussagekräftige auditive Informationen für eine effektive Videoklassifikation wichtig. Eine Analyse des Audiosignals kann sowohl für die

Detektion von Shot-Übergängen [61, 21] und die Identifikation sprechender Personen [62, 63], als auch zur Segmentierung von Musik, Sprache, Stille und Umgebungsgeräuschen innerhalb eines Videos eingesetzt werden [64, 65]. In diesem Abschnitt wird zunächst die Analyse von Audiosignalen erläutert und anschließend typische auditive Features vorgestellt, die zur Klassifikation von Musik, Sprache und Umgebungsgeräuschen häufig eingesetzt werden.

2.5.1 Analysieren von Audiosignalen

Im Allgemeinen dient eine auditive Merkmalsextraktion zur Berechnung von aussagekräftigen Features, die in der Lage sind, das Audiosignal zu repräsentieren. Generell werden dabei nicht alle Daten (*Samples*) des Signals als Ganzes betrachtet, sondern es wird zunächst eine Unterteilung in Fenster (*Windows*) vorgenommen. Wie diese Unterteilung vorgenommen wird ist abhängig von den folgenden Parametern [44]:

- **Größe der Fenster:**
  Die Größe des Fensters wird durch die Anzahl der Samples definiert, die sich innerhalb einer solchen Unterteilung befinden. Typische Fenstergrößen variieren zwischen *20ms* und *1s*, abhängig welches Ziel mit der anschließenden Analyse verfolgt wird (siehe Abbildung 2.16) [66, 67].

- **Abstand zwischen den Fenstern:**
  Der Abstand wird als *Hop Size (HS)* bezeichnet und beschreibt die Anzahl der Samples, welche zwischen zwei Fenstern liegen. Je nachdem wie der Abstand gewählt wird, werden für eine anschließende Analyse entweder alle Samples (*HS=0*) betrachtet oder gewisse Samples verworfen (*HS>0*). Im Fall von *HS<0* treten bei der Aufteilung der Fenster Überlappungen auf, wodurch Samples öfters ausgewertet werden (siehe Abbildung 2.16) [44].

Die Wahl dieser Parameter ist abhängig vom Anwendungsbereich. Jedoch sind diese entscheidend für eine effiziente Analyse des Audiosignals.

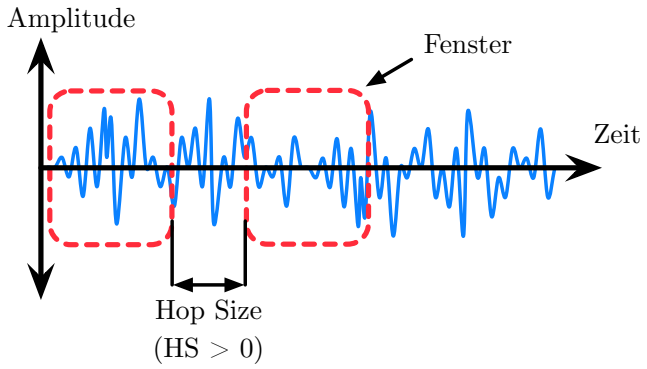


Abbildung 2.16: Unterteilung eines Audiosignals in Fenster.

Zusätzlich ist speziell im Fall von Videos zu beachten, dass mehrere Audiokanäle vorhanden sein können. Es existieren unterschiedliche Strategien wie eine Analyse auf mehreren Signalen durchgeführt wird [66]. Die einfachste Methode ist, nur einen Kanal zu betrachten und die restlichen zu verwerfen. Dies ist sinnvoll wenn alle vorhandenen Audiokanäle nahezu identisch sind. Unterscheiden sich die Signale aus den einzelnen Audiokanälen signifikant, so besteht die Möglichkeit eine unabhängige Analyse für jeden einzelnen Kanal durchzuführen oder die resultierenden Werte des Feature-Vektors statistisch zu kombinieren (z.B. Mittelwert) [44].Im Anschluss werden verschieden auditive Features vorgestellt, die häufig bei einer Segmentierung von Musik, Sprache, Stille und Umgebungsgeräuschen zum Einsatz kommen. Welche Merkmale tatsächlich geeignet sind, ist abhängig vom Anwendungsbereich [68].

2.5.2 Short Time Energy

Die *Short Time Energy (STE)* ist in der Lage die Variation der Amplitude eines Audiosignals zu repräsentieren und ist folgendermaßen definiert:

$$STE = \frac{1}{N}\sum_{m}[x(m) * w(m)]^2 \tag{2.12}$$

wobei $x(m)$ dem Amplitudenwert zum Zeitpunkt m entspricht. Die Funktion $w(m)$ wird herangezogen um das Signal in Fenster mit einer Größe von N Samples zu unterteilen [69]:

$$w(m) = \begin{cases} 1, & 1 \leq m \leq N \\ 0, & sonst \end{cases} \tag{2.13}$$

In anderen Worten, bei der STE handelt es sich um den Mittelwert der quadrierten Amplitudenwerte innerhalb eines gewissen Fensters. Dieses Audiofeature wird beispielsweise bei der Detektion von Stille eingesetzt, wo die STE einen geringen Wert aufweist.

2.5.3 Zero Crossing Rate

Ein weiteres typisches auditives Merkmal wird als *Zero Crossing Rate (ZCR)* bezeichnet. Dieses Feature ist in der Lage die grundlegende Frequenz eines Audiosignals zu repräsentiert, indem die Anzahl der Nulldurchgänge innerhalb eines Fensters gezählt werden:

$$ZCR = \frac{1}{2}\sum_{m=2}^{N} \mid sign(x(m)) - sign(x(m-1)) \mid \quad \text{mit} \quad sign(a) = \begin{cases} 1, & a > 0 \\ 0, & a = 0 \\ -1, & a < 0 \end{cases} \tag{2.14}$$

wobei $x(m)$ dem Amplitudenwert zum Zeitpunkt m entspricht [66]. In Abbildung 2.17 wird der Zusammenhang zwischen Frequenz eines Audiosignals und der ZCR aufgezeigt. Je höher die Anzahl der Nulldurchgänge ist, desto größer wird auch die grundlegende Frequenz angenommen [70]. Dieses Feature wird aufgrund der simplen Berechnung häufig in verschieden Anwendungen der Audioanalyse genutzt [68].

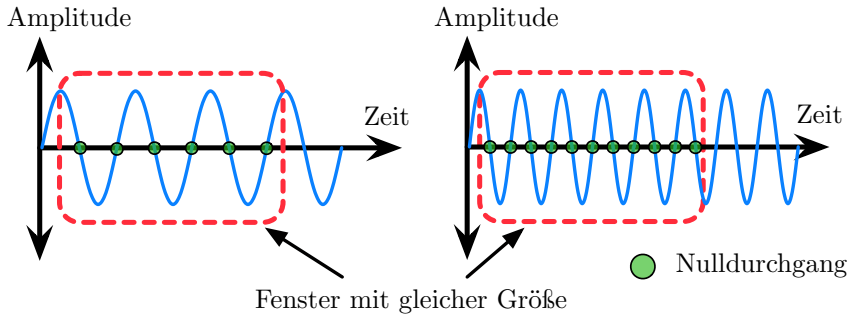


Abbildung 2.17: Zusammenhang zwischen Zero Crossing Rate und grundlegender Frequenz eines Audiosignals.

2.5.4 Root Mean Square

Die Lautstärke eines Audiosignals ist ebenfalls ein auditives Merkmal, welches vor allem für die Segmentierung von Sprache und Musik eingesetzt wird. Der sogenannte *Root Mean Square (RMS)* dient zur Approximation der Lautstärke und ist der STE ähnlich [66, 68]:

$$RMS = \sqrt{\frac{1}{N}\sum_{m} x(m)^2} \tag{2.15}$$

wobei $x(m)$ dem Amplitudenwert zum Zeitpunkt m entspricht und die Anzahl der Samples durch N repräsentiert wird.

2.5.5 Mel-Frequency Cepstral Coefficients

Eines der bekanntesten auditiven Merkmale sind die sogenannten *Mel-Frequency Cepstral Coefficients (MFCC)*, welche häufig in den verschiedensten Anwendungsbereichen der Audioanalyse zu finden sind. Im Allgemeinen sind MFCCs in der Lage das Frequenzspektrum eines Audiosignals zu beschreiben. Die Berechnung dieser Koeffizienten erfolgt in mehreren Schritten (siehe Abbildung 2.18). Für jedes Fenster eines Audiosignals wird zuerst mit Hilfe einer diskreten

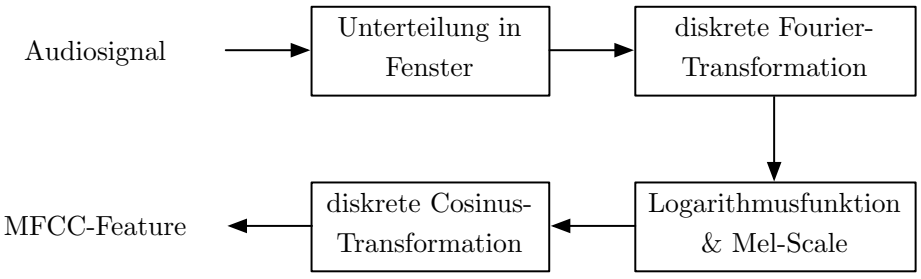


Abbildung 2.18: Struktureller Ablauf zur Berechnung der Mel-Frequency Cepstral Coefficients (in Anlehnung an [14]).

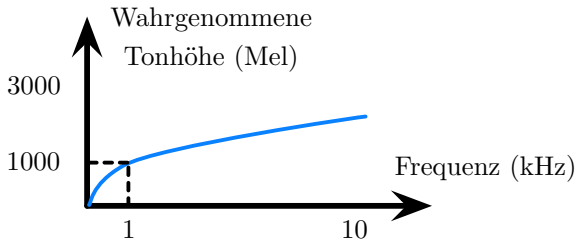


Abbildung 2.19: Menschliche Wahrnehmung der Tonhöhe abhängig von der Frequenz.

Fourier-Transformation (DFT) das Frequenzspektrum des Signals berechnet [14]. Das Spektrum setzt sich aus Real- und Imaginärteil zusammen, wobei es in der Signalanalyse üblich ist, nur einen der beiden Teile weiter zu betrachten [71].

Im nächsten Schritt gilt es, mit Hilfe einer logarithmischen Analyse aussagekräftige Frequenzen aus dem Spektrum hervorzuheben. Dabei wird ein sogenannter *Mel-Scale* eingesetzt, der den in Mel gemessenen Zusammenhang zwischen wahrgenommener Tonhöhe und tatsächlicher Frequenz repräsentiert (siehe Abbildung 2.19). Unter einer Frequenz von *1kHz* wirkt sich eine Veränderung der Frequenz überproportional auf die wahrgenommen Tonhöhe aus. Bei Frequenzen über *1kHz* hat eine Frequenzänderung abnehmenden Einfluss auf die wahrgenommene Tonhöhe [44]. Nachdem die aussagekräftigen Frequenzen aus dem Spektrum ausgewertet worden sind, werden im letzten Schritt einzelnen Koeffizienten berechnet. Mit Hilfe einer diskreten Cosinus-Transformation (DCT), wird der Großteil der Informationen durch wenige Koeffizienten repräsentiert. Abhängig vom Anwendungsbereich werden die ersten *8 – 13* MFCC-Koeffizienten für eine Beschreibung des Audiosignals herangezogen [68].

Die letzten beiden Abschnitte zeigen, wie aus Videos sowohl visuelle als auch auditive Merkmale extrahiert werden können. In Form eines Vektors werden diese Features anschließend für eine Klassifikation herangezogen.

2.6 Klassifikationsmodelle

Im letzten Schritt einer Videoklassifizierung werden die extrahierten Merkmale herangezogen und analysiert, um eine semantische Zuordnung einzelner Objekte (z.B. Shots aus einem Video) zu erlangen. Ziel ist es, mit geeigneten Klassifikationsmodellen unbekannte Objekte gewissen Klassen oder Kategorien zuzuordnen. Abhängig vom Anwendungsbereich unterscheiden sich sowohl die Wahl der Modelle, als auch die Bedeutung der einzelnen Klassen, die oft durch sogenannte Labels in nummerischer Form dargestellt werden.

In den meisten Fällen ist für die Klassifikation eine Trainingsphase erforderlich, um eine geeignete Parametrisierung der gewählten Klassifikationsmodelle zu bestimmen. Dazu wird eine sogenannte *Ground Truth (GT)* benötigt, die sich aus Informationen über Klassenzugehörigkeit und extrahierten Merkmalen von bereits bekannten Objekten zusammensetzt [44]. Die Bildung der GT-Daten kann mit enormem Aufwand verbunden sein. In einem meist manuellen Verfahren werden von einer oder mehreren Personen einzelne Objekte den vorgegebenen Klassen zuge-

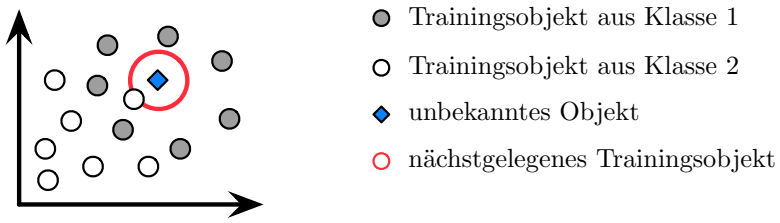


Abbildung 2.20: Funktionsweise der Nearest Neighbour-Klassifikation.

ordnet, aus denen anschließend automatisch aussagekräftige Merkmale extrahiert werden. Diese Informationen werden für den eigentlichen Trainingsprozess herangezogen, um die Klassifikationsmodelle speziell für den Anwendungsbereich zu parametrisieren. Welche Modelle am besten geeignet sind, ist sowohl abhängig von der Anwendung der Klassifikation, als auch von der Verteilung der extrahierten Merkmale. Außerdem spielt bei der Entscheidung die Generalisierungsfähigkeit der verschiedenen Modelle eine wichtige Rolle. Diese Fähigkeit beschreibt, ob ein gewähltes Modell nach dem Trainingsprozess in der Lage ist, unbekannte Objekte korrekt zu klassifizieren [72, 73].In den nächsten Abschnitten werden die verschiedenen Klassifikationsmodelle vorgestellt, die in dieser Arbeit Anwendung finden und ihre Einsatzmöglichkeiten erläutert.

2.6.1 Nearest Neighbour

Eine simple Methode zur Bestimmung der Klasse eines unbekannten Objektes ist das *Nearest Neighbour*-Klassifikationsmodell (*NN*). Das Besondere an diesem Klassifikationsmodell ist, dass kein Trainingsprozess notwendig ist. Die einzelnen Feature-Vektoren der Trainingsdaten werden in einem mehrdimensionalen Raum (*Feature-Space*) dargestellt (siehe Abbildung 2.20) [74], wobei die Dimensionszahl der Länge des Feature-Vektors entspricht. Jeder Punkt im Feature-Space repräsentiert ein Objekt aus dem Trainings-Set und dessen Klassenzugehörigkeit. In weiterer Folge werden diese Punkte als Trainingsobjekte bezeichnet.

Um ein unbekanntes Objekt zu klassifizieren wird der entsprechende Feature-Vektor ebenfalls im Raum betrachtet und es werden die Abstände (z.B. euklidische Distanz) zu allen vorhandenen Trainingsobjekten berechnet. Das unbekannte Objekt wird jener Klasse zugewiesen, die das Objekt mit dem geringsten Distanzwert aus dem Training-Set aufweist [44]. Der Vorteil dieses Klassifikationsmodells ist, dass kein Trainingsprozess durchgeführt werden muss. Jedoch hat die Repräsentation aller Trainingsdaten im Feature-Space einen erhöhten Speicherbedarf zur Folge. Außerdem hat die Berechnung der Distanzen vom unbekannten Objekt zu allen Trainingsobjekten einen Einfluss auf die Rechenzeit.

Eine weitere Einschränkung von Nearest Neighbour ist aufgrund der Verteilung von den Trainingsobjekten im Feature-Space gegeben, die Auswirkungen auf die Qualität der Resultate haben. Es können Überlappungen der Trainingsobjekte mit verschiedenen Klassen auftreten, wodurch geringe Abweichungen der Positionierung des unbekannten Objekts im Feature-Space unterschiedliche Klassifikationsresultate zur Folge haben können. Eine Klassifikation mittels NN ist dann sinnvoll, wenn in lokalen Bereichen jeweils nur eine Klasse vertreten ist [74].

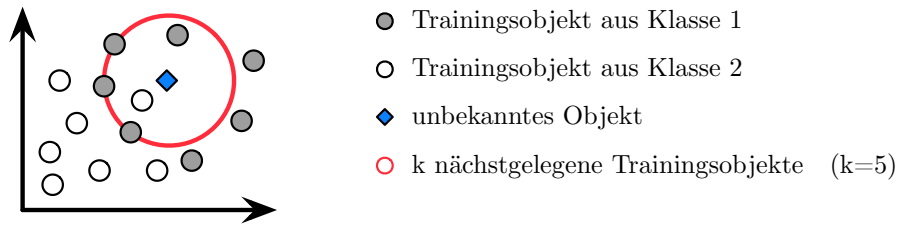


Abbildung 2.21: K-Nearest Neighbour.

2.6.2 K-Nearest Neighbour

Bei *K-Nearest Neighbour (K-NN)* handelt es sich um eine Erweiterung des NN-Klassifikationsmodells. Die grundlegende Vorgangsweise bleibt erhalten, jedoch werden bei der Klassifikation eines unbekannten Objektes die K geringsten Distanzwerte zu den Objekten aus dem Training-Set betrachtet. Dem unbekannten Objekt wird jene Klasse zugewiesen, die bei den K nächstgelegenen Objekten am häufigsten auftritt (siehe Abbildung 2.20) [46].

Die Vor- und Nachteile sind ähnlich wie beim NN-Klassifikationsmodell. Zusätzlich ist die Wahl des Parameters K entscheidend für die Qualität der Klassifikationsresultate. Jedoch können für diese Entscheidung keine bestimmten Kriterien festgelegt werden, da K abhängig vom Anwendungsbereich und der Verteilung der vorliegenden Daten ist.

2.6.3 K-Means

Eine weitere Methode zur Klassifikation ist der *K-Means*-Algorithmus. Im Gegensatz zu NN und K-NN werden hierbei nicht die einzelnen Objekte aus den Trainingsdaten betrachtet (siehe Abbildung 2.22). Sondern es wird für jede Klasse ein Referenzobjekt bestimmt, welches die gesamte Klasse repräsentiert. Die Klassifikation eines unbekannten Objektes verläuft nach dem gleichen Prinzip wie bei NN und K-NN.

Jedoch werden bei K-Means nur die Distanzen zwischen unbekanntem Objekt und den Referenzobjekten im Feature-Space betrachtet. Das unbekannte Objekt wird jener Klasse zugewiesen, die das nächstgelegene Referenzobjekt aufweist [44]. Auch bei K-Means können unterschiedliche Differenzmetriken zum Einsatz kommen. Entscheidend für die Klassifikation ist

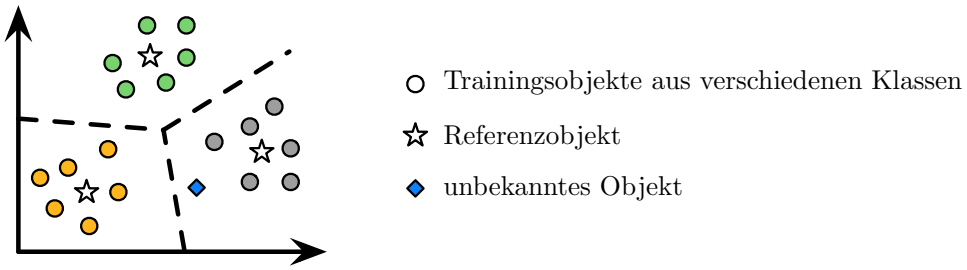


Abbildung 2.22: Klassifikation mittels K-Means.

der Parameter K, der die Anzahl der Referenzobjekte bestimmt. Außerdem hat die Verteilung der Objekte aus den Trainingsdaten und die damit verbundene Lage der Referenzobjekte im Feature-Space Einfluss auf die Qualität der Klassifikationsresultate.

2.6.4 KD-Baum

Wie bereits in Abschnitt 2.6.1 erläutert, ist das Finden des nächstgelegenen Objektes aus den Trainingsdaten ein simples aber auch effektives Klassifikationsmodell. Jedoch ist das NN-Modell aufgrund der Abstandsberechnung zwischen dem unbekannten Objekt und den Trainingsobjekten mit einem enormen Rechenaufwand verbunden, abhängig von der Größe des Trainings-Sets. Eine Alternative zur Bestimmung des nächstgelegenen Objektes ist die Verwendung eines sogenannten KD-Baums [75].

In Abbildung 2.23 wird der Aufbau eines solchen binären Baumes visualisiert. Die einzelnen Feature-Vektoren der Objekte aus dem Training-Set werden wieder im Feature-Space betrachtet. Anschließend wird in einem iterativen Prozess eine Unterteilung des mehrdimensionalen Raumes vorgenommen und daraus ein Baum aufgebaut. Im ersten Unterteilungsschritt wird der Feature-Space durch eine Hyperebene in zwei Hälften geteilt. Diese Hyperebene liegt orthogonal zu jener Dimensionsrichtung, welche die höchste Varianz bezüglich der Trainingsdaten aufweist. Im Allgemeinen wird der Median in diese Dimensionsrichtung als Anhaltspunkt für die Positionierung der Hyperebene herangezogen [74, 76]. Dieser Unterteilungsschritt ist auch im binären Baum zu erkennen. Der gesamte Feature-Space wird durch den Wurzelknoten repräsentiert und die beiden entstehenden Hälften werden durch zwei nachfolgende Knoten dargestellt. Dieser Unterteilungsvorgang wird rekursiv fortgesetzt, bis keine Aufteilung der Trainingsobjekte mehr möglich ist [75]. Der entstehende Baum kann anschließend für eine Klassifikation herangezogen werden.

Das Prinzip des Klassifikationsprozesses ist ähnlich wie bei NN. Einem unbekanntem Objekt wird jene Klasse zugewiesen, die dem nächstgelegenen Trainingsobjekt im Feature-Space entspricht. Jedoch wird anstelle von einer Menge an Differenzberechnungen der KD-Baum herangezogen und schrittweise, vom Wurzelknoten beginnend, nach unten ausgewertet. Am Ende

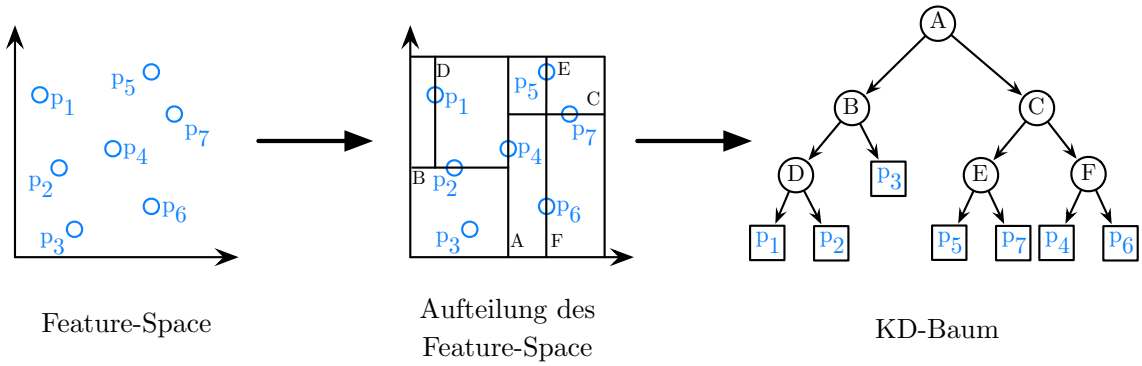
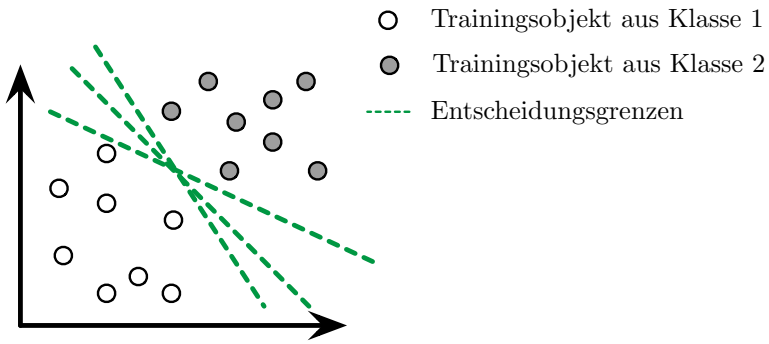
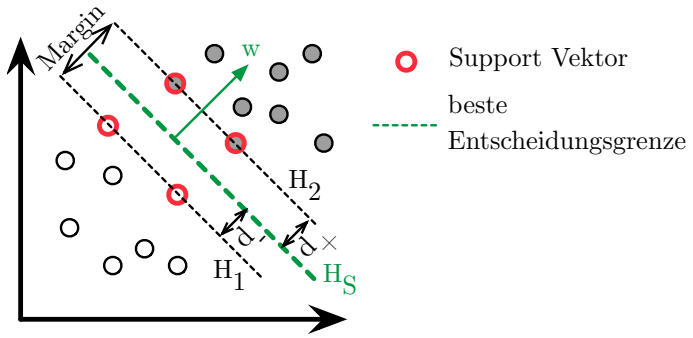


Abbildung 2.23: Aufbau eines KD-Baums in einem 2D-Feature-Space.



(a) Mögliche Entscheidungsgrenzen



(b) Beste Entscheidungsgrenze

Abbildung 2.24: Entscheidungsgrenzen von linear trennbaren Daten im 2D-FeatureSpace.

wird jener Blattknoten bestimmt, der dem nächstgelegenen Trainingsobjekt und der damit verbundenen Klasse entspricht.

2.6.5 Support Vector Machine

Als letztes Klassifikationsmodell wird hier die *Support Vector Machine (SVM)* vorgestellt, deren Effizienz für verschiedene Anwendungsbereiche nachgewiesen wurde [77]. Die grundlegende Funktionsweise der SVM basiert auf linear trennbaren Datenpunkten im Feature-Space, die zwei unterschiedlichen Klassen angehören [78]. Wie in Abbildung 2.24a zu sehen ist, können mehrere Entscheidungsgrenzen existieren, die in der Lage sind, die Daten entsprechend der Klassenzugehörigkeit zu trennen. Ziel der SVM ist es, die bestmögliche Entscheidungsgrenze zu finden (siehe Abbildung 2.24b), um eine hohe Generalisierungsfähigkeit zu erreichen. Die Entscheidungsgrenze wird durch eine Hyperebene H_s beschrieben:

$$H_s: \quad w \cdot x + b = 0 \tag{2.16}$$

Die Hyperebene wird durch den Normalvektor w und dem Normalabstand zum Ursprung des Feature-Spaces $\frac{|b|}{\|w\|}$ definiert, wobei $\|w\|$ der euklidischen Norm von w entspricht. Zur Bestim-

mung der bestmöglichen Entscheidungsgrenze werden zunächst die Datenpunkte x_i aus dem Trainings-Set und die zugehörigen Klassen y_i betrachtet, die im Falle einer linearen Trennbarkeit folgende Gleichungen erfüllen müssen [78, 77]:

$$\begin{aligned} w \cdot x_i + b &\geq +1 \qquad \text{für Klasse 1} \quad (y_i = +1) \\ w \cdot x_i + b &\leq -1 \qquad \text{für Klasse 2} \quad (y_i = -1) \end{aligned} \tag{2.17}$$

Basierend auf linear trennbaren Trainingsdaten x_i und deren Definition werden zwei Grenzebenen H_1 und H_2 definiert, die jeweils die Verteilung der Datenpunkte aus den beiden Klassen eingrenzen (siehe Abbildung 2.24b). Dabei spielen jene Datenpunkte, die auf den Grenzebenen liegen eine wichtige Rolle. Diese Punkte werden als *Support-Vektoren* bezeichnet und sind entscheidend für die Definition und Lage der Grenzebenen [14]:

$$\begin{aligned} H_1 : \;& w \cdot x_i + b = 1 \\ H_2 : \;& w \cdot x_i + b = -1 \end{aligned} \tag{2.18}$$

Die Positionierung der Grenzebene H_1 wird durch ihren Normalvektor w und den Normalabstand zum Ursprung $\frac{|1-b|}{\|w\|}$ beschrieben. Auf gleiche Weise wird die Positionierung der Grenzebene H_2 bestimmt, wobei der Normalabstand $\frac{|-1-b|}{\|w\|}$ beträgt. Diese beiden Grenzebenen und deren unterschiedlichen Normalabstände werden herangezogen, um die bestmögliche Positionierung der Hyperebene zur Trennung der Daten zu ermitteln. Dabei soll die gesuchte Hyperebene in der Mitte zwischen den Grenzebenen liegen, die einen möglichst großen Abstand zu einander aufweisen sollen. Dieser Abstand wird als *Margin* bezeichnet und lässt sich durch die Normalabstände der Grenzebenen und der Hyperebene berechnen [77]:

$$\begin{aligned} d_+ = d_- &= \frac{1}{\|w\|} \\ d_m = d_+ + d_- &= \frac{2}{\|w\|} \end{aligned} \tag{2.19}$$

wobei d_+ und d_- jeweils dem Abstand von einer Grenzebene zur Hyperebene entspricht und d_m die daraus resultierende Margin repräsentiert. Anhand dieser Definition ist zu erkennen, dass eine Minimierung von $\|w\|$ zu dem gewünschten maximalen Margin führt [14]. Auf diese Weise kann die bestmögliche Hyperebene bestimmt werden, die anschließend für eine Klassifikation herangezogen wird.

Jedoch sind die Daten aus unterschiedlichen Klassen meist nicht linear trennbar. Aus diesem Grund wird zunächst mit Hilfe einer Kernelfunktion eine Transformation der Daten vorgenommen. In Abbildung 2.25 wird dieser Vorgang visualisiert. Es werden dabei die Daten in einen höherdimensionalen Raum transformiert, wo eine bestmögliche Hyperebene für eine lineare Trennung gesucht wird. Anschließend wird diese Hyperebene wieder auf den ursprünglichen Raum rückprojiziert, wodurch eine nichtlineare Entscheidungsgrenze bestimmt wird [79]. Die Vorteile der SVM liegen in der hohen Generalisierungsfähigkeit. Selbst bei nichtlinear trennbaren Daten kann, basierend auf der Analyse von wenigen Support-Vektoren, durch die Anwendung einer

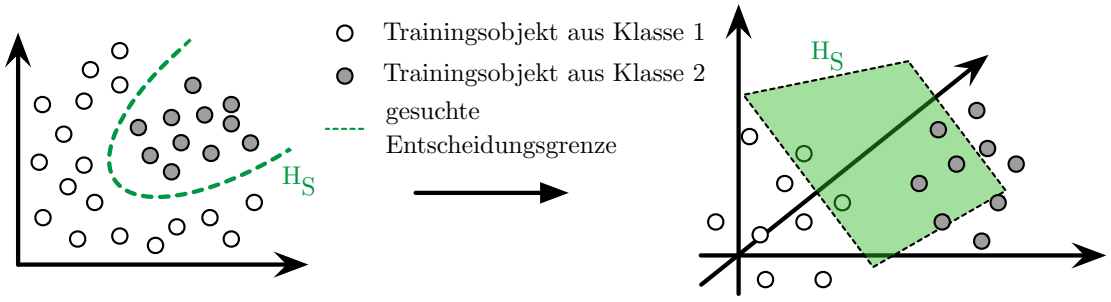


Abbildung 2.25: Finden einer Entscheidungsgrenze bei nicht linear trennbaren Daten.

Kernelfunktion im Trainingsprozess eine bestmögliche Entscheidungsgrenze gefunden werden. Welche Kernelfunktion eingesetzt werden soll, ist abhängig von der Verteilung der Daten und dem Anwendungsbereich.

Zusammenfassend werden aussagekräftige Merkmale für das Trainieren eines Modells benötigt, um eine spätere Klassifikation zu ermöglichen. Die Qualität der Klassifikationsresultate wird anschließend mit Hilfe verschiedener Evaluierungsmethoden bewertet.

2.7 Methoden zur Evaluierung

Der letzte Schritt einer Videoklassifizierung, aber dennoch ein wichtiger, ist die Evaluierung des Systems, um die Qualität der gewählten audiovisuellen Merkmale und die eingesetzten Klassifikationsmodelle zu überprüfen. Dabei können verschiedene Evaluierungsmethoden herangezogen werden. Jedoch werden in den meisten Fällen für eine sinnvolle Auswertung aussagekräftige Ground-Truth-Daten benötigt [44]. In Abbildung 2.26 werden typische Evaluierungsbegriffe vorgestellt, die sich aus vorhanden GT-Daten und den Resultaten des Klassifikationsprozesses

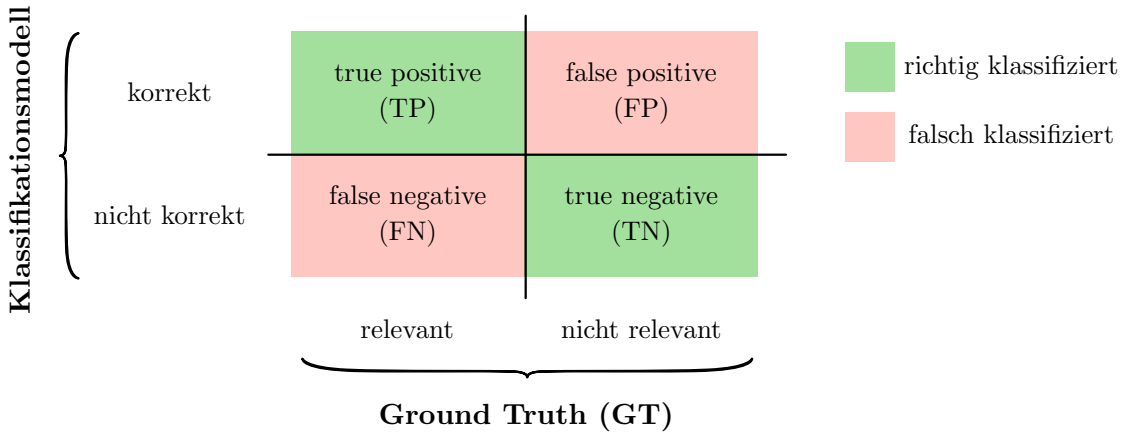


Abbildung 2.26: Mögliche Klassifikationsresultate.

definieren lassen. Dabei wird zwischen korrekt (TP und TN) und nicht korrekt (FP und FN) klassifizierten Ereignissen (z.B. Szenen eines Videos) unterschieden. So wird beispielsweise ein Klassifikationsresultat als *false positive* (FP) bezeichnet, wenn das Ereignis fälschlicherweise als relevant klassifiziert worden ist.

Anhand dieser Begriffe lassen sich weitere Evaluierungen erklären, die häufig in verschiedenen Anwendungsbereichen eingesetzt werden [80, 81]:

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \end{aligned} \tag{2.20}$$

Der *Recall* beschreibt wie viele der relevanten Ereignisse tatsächlich erkannt worden sind. Im Gegensatz dazu repräsentiert die *Precision* die Relevanz der vom System erkannten Ereignisse [44]. Eine alleinige Betrachtung von Recall oder Precision liefert keine aussagekräftige Information über die Performance eines Systems. Beispielsweise kann ein hoher Recall erreicht werden (jedes Ereignis wird als relevant erkannt), aber auf Kosten einer niedrigen Precision (nur wenige der erkannten Ereignisse sind tatsächlich relevant). Aus diesem Grund wird für eine aussagekräftige Evaluierung oft der sogenannte *F1-Score* herangezogen, wo Recall und Precision gleichermaßen einfließen [80]:

$$\textit{F1-Score} = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{2.21}$$

Um die Qualität eines Systems zu ermitteln, gilt es in einem Evaluierungsprozess den F1-Score zu ermitteln. Eine häufig eingesetztes Modell zur Evaluierung ist die Kreuzvalidierung (*cross validation*), deren Ablauf in Abbildung 2.27 visualisiert wird. Im Allgemeinen werden dabei die vorhandenen GT-Daten wiederholt in eine Trainingsmenge und in eine Testmenge aufgeteilt. Die Daten aus dem Training-Set werden für das Training des Klassifikationsmodells herangezogen, während das Test-Set zur Evaluierung und Berechnung von Recall, Precision und F1-Score dient. Dieser Vorgang wird mehrmals wiederholt. Zuletzt werden die ermittelten Evaluierungsdaten über alle Durchläufe ausgewertet, um die Performance des gesamten Systems zu bestimmen [14].

Eine charakteristische Eigenschaft von Cross Validation ist, dass bei jedem Durchlauf die Elemente von Trainings- und Testmenge neu gewählt werden, so dass möglichst viele Variationen in die Evaluierung einfließen können. Eine besondere Form ist dabei die sogenannte *Leave One Out Cross Validation (LOO-CV)* [82]. Dabei wird von jeder Klasse ein Objekt zur Evaluierung herangezogen, während die übrigen Objekte für das Training des Systems verwendet werden. Dieser Vorgang wird wieder mehrmals wiederholt, wobei bei jedem Durchgang unterschiedliche Testdaten gewählt werden [46].

Cross Validation in Verbindung mit Recall, Precision und F1-Score bietet eine Möglichkeit zur aussagekräftigen Evaluierung eines Systems. Besonders die Auswertung der LOO-CV liefert wichtige Informationen über die Qualität und Generalisierungsfähigkeit von Klassifikationsmodellen.

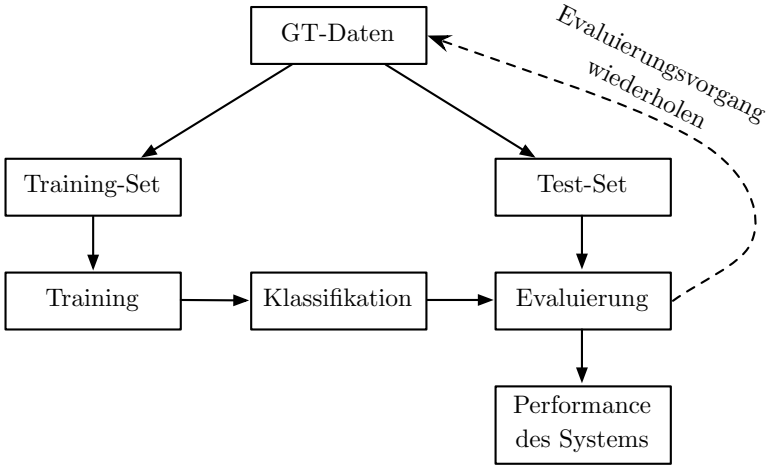


Abbildung 2.27: Ablauf von Cross Validation.

2.8 Verwandte Arbeiten

Zum Abschluss dieses Kapitels werden in diesem Abschnitt ausgewählte Systeme zur Videoklassifikation vorgestellt. Jedoch existiert zurzeit noch kein System zur Segmentierung und Klassifikation von Handpuppen oder Szenen aus der Muppet Show. Aus diesem Grund werden im Folgenden die Funktionsweisen von Prototypen aus ähnlichen Anwendungsbereichen kurz aufgezeigt.

In [83] wird ein System zur inhaltsbasierten Analyse von Nachrichtensendungen beschrieben. Dabei werden nicht nur einzelnen Bestandteile der Sendung segmentiert, sondern es werden auch individuelle Nachrichtenthemen automatisch klassifiziert. Im ersten Schritt wird mit Hilfe von Bewegungsvektoren eine zeitliche Segmentierung durchgeführt, um ein Video in einzelne Shots zu unterteilen. Anschließend werden, basierend auf charakteristischen Eigenschaften bei der Gestaltung von Nachrichtenprogrammen, lokale Regionen und deren räumlicher Zusammenhang ausgewertet. Auf diese Weise können verschieden Arten von Nachrichtensequenzen detektiert werden, wie zum Beispiel Wettervorschau oder Nachrichtenbeiträge.

Merlino u. a. präsentieren in [6] ein ähnliches System zur Segmentierung von Nachrichtensendungen. Ausgehend von einem Video werden sowohl auditive Informationen zur Detektion von Stille als auch eine textuelle Analyse von Untertiteln herangezogen, um einzelne Sequenzen und deren Übergänge zu detektieren. Anschließend wird mit Hilfe einer visuellen Merkmalsextraktion und der Auswertung von auftretenden Schlüsselwörtern (z.B. typische Begrüßungsworte zu Beginn) eine Klassifikation der Nachrichtensequenzen durchgeführt.

Ein weiterer Anwendungsbereich von Videoklassifikation ist die Detektion von Werbung. In [84] werden nach einer zeitlichen Segmentierung audiovisuelle Features extrahiert, um eine aussagekräftige Beschreibung zu ermitteln. Anschließend wird für jeden Shot mit Hilfe der berechneten Features und einer SVM bestimmt, ob es sich dabei um Werbung handelt oder nicht. Abschließend werden in einem Nachbearbeitungsschritt einzelne Shots zu Szenen gruppiert, abhängig vom zeitlichen Auftreten und dem Ergebnis der inhaltsbasierten Analyse. Bei einer

Evaluierung auf mehrstündigem Videomaterial mit unterschiedlichem Inhalt wird eine Klassifikationsrate zwischen *94%* und *96%* erreicht.

Trotz der Ähnlichkeiten, die diese Systeme zu unserem Anwendungsproblem aufweisen, sind die vorgestellten Algorithmen für die Segmentierung und Klassifikation von Videos aus der Muppet Show nicht zwingend geeignet. Zwar sind Nachrichtensendungen und Werbeprogramme auch aus verschiedenen Arten von Sequenzen aufgebaut, jedoch treten im Vergleich zur Muppet Show keine spezielle Übergangseffekte auf, wie beispielsweise das Öffnen des Bühnenvorhangs. Außerdem setzten sich Nachrichtensendungen nur aus wenigen variationsarmen Bestandteilen zusammen. Häufig unterliegen die Sendungen dem gleichen Aufbau, wonach nur wenige Personen durch die Sendung führen, oft das gleiche Szenenbild verwendet wird und die einzelnen Szenenarten in bestimmter Reihenfolge zu sehen sind (z.B. Wettervorschau am Ende der Nachrichtensendung). Im Gegensatz dazu handelt es sich bei der Muppet Show um ein variationsreiches Videomaterial, wo zwar häufig ähnliche Bühnenbilder eingesetzt werden, jedoch die Vielzahl an unterschiedlichen Charakteren und das dynamische Geschehen die Videoklassifikation erschweren. Aus diesem Grund werden aussagekräftige audiovisuelle Merkmale und geeignete Klassifikationsmodelle benötigt. Wie die Analyse des Videomaterials hilfreich sein kann, für die Entscheidung welche Features und Modelle eingesetzt werden sollen, zeigt das folgende Kapitel.

KAPITEL 3

Videomaterial: Die Muppet Show

Die Analyse des Videomaterials ist entscheidend für die Wahl aussagekräftiger audiovisueller Merkmale und geeigneter Klassifikationsmodelle, um eine effiziente Videoklassifikation zu realisieren. In diesem Kapitel wird das Videomaterial vorgestellt, ein kurzer Rückblick auf die Entstehung der Muppet Show gegeben und charakteristische Merkmale erläutert, die für die Videoklassifikation von Bedeutung sein können.

3.1 Entstehungsgeschichte

Die grundlegende Idee und Entwicklung der Muppet Show liegt mehr als *60* Jahre zurück. Bereits *1950* wurde *Jim Henson* (siehe Abbildung 3.1), der Erfinder der Muppets, durch verschiedene Puppenspiele inspiriert. Die zu diesem Zeitpunkt bereits im Fernsehen ausgestrahlten Puppen-Shows weckten sein Interesse für die künstlerische Gestaltung und dem Umgang mit Puppen oder Marionetten [85]. Während der Schulzeit war Jim Henson Mitglied in einem Club für Handpuppenspiel, wo er seine Begeisterung für Handpuppen mit anderen Personen teilen konnte. Ein weiteres großes Interesse von Henson galt dem Fernsehen. Im Verlauf seines Studiums ermöglichte der lokale Sender WTOP Henson, das Medium Fernsehen mit einer eigenen Puppenshow besser kennenzulernen [86, 87]. Jedoch handelte es sich dabei nur um einen Kurzauftritt und nach wenigen Wochen wurde die Sendung bereits abgesetzt. Trotzdem verlor Jim Henson nicht das Interesse. Er war davon fasziniert, welche technischen Möglichkeiten das Puppenspiel in Kombination mit Fernsehen bietet und empfand den Entwurf und Bau von Puppen, die Gestaltung eines ansprechenden Bühnenbildes und das Schreiben von Texten als Herausforderung [88]. Das Besondere an seiner Arbeit war, dass es sich bei den Shows nicht um gewöhnliche Auftritte in Form eines Puppentheaters handelte. Henson versuchte von Anfang an sich auf das Medium Fernsehen zu konzentrieren und nutzte dieses Medium als einzigartige Bühne [89]. Die Puppenspieler konnten während des Spielens weder die Puppen, noch ihre Kollegen sehen, sondern sie verfolgten das Geschehen direkt am Fernsehbildschirm [88]. Diese Spielweise ermöglichte eine besondere Interaktion zwischen den Puppen und durch den Einsatz von damals modernen Kameraeffekten konnten verschiedene Perspektiven hervorgehoben und spezielle Bewegungen dargestellt werden.

Ende *1955* bekam Jim Henson von einer Tochtergesellschaft von NBC eine fünf minütige Sendezeit, wo er mit *Jane Nebel*, einer Studienkollegin, wieder eine Puppen-Show aufführte (siehe Abbildung 3.2). In dieser Show, mit dem Titel *Sam and Friends*, war erstmals das revolutionäre Puppendesign zu sehen, die Kombination von starren Handpuppen (engl. *puppets*) und simplen Marionetten. Daher stammt auch der Name dieser neuartigen Puppen: Muppets [85].

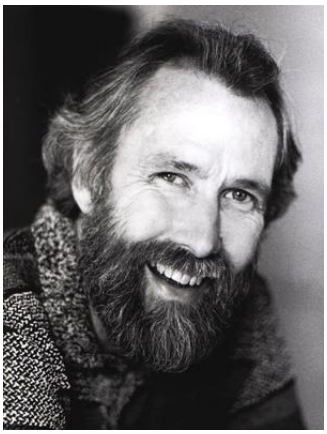


Abbildung 3.1: Jim Henson, Erfinder der Muppets [90].

Abbildung 3.2: Die Puppen aus der Sendung *Sam and Friends* gespielt von Jane Nebel und Jim Henson [91].

Es wird an dieser Stelle nochmal darauf hingewiesen, dass sich der Begriff Muppet nicht nur auf Charaktere der Muppet-Show beschränkt, sondern es sich dabei um eine spezielle Art von Handpuppen handelt, die in mehreren Shows (z.B. *Sam and Friends*, *Sesamstraße*, *Die Fraggles* etc.) zu sehen sind.

Nachdem Henson sein Studium abgeschlossen hatte, machte er sich Gedanken über seine berufliche Zukunft. Erst während einer Reise durch Europa, wo er andere professionelle Puppenspieler traf, überzeugte er sich davon, dass es sich bei Entwurf, Bau und Gestaltung der Puppen um kreative Kunst handelt [88]. In den folgenden Jahren vergrößerte sich das Team um Jim Henson immer weiter und es wurde jener Stil entwickelt, der die Charakteristik der Muppets prägt. Einen maßgeblichen Anteil am Erfolg dieser besonderen Handpuppen hatten die beiden Puppenspieler *Jerry Juhl* und *Frank Oz*, sowie der Puppenhersteller *Don Sahlin* [92]. Während Juhl hauptsächlich für die kreativen Sketch- und Musiktexte verantwortlich war, bildeten Jim Henson und Frank Oz ein gut eingespieltes Team, deren harmonisches Zusammenspiel für eine lebensnahe Darstellung der Puppenshow sorgte (siehe Abbildung 3.3). Außerdem kreierte Jim Henson in Zusammenarbeit mit Sahlin den sogenannten Muppet-Look, welcher sich einerseits durch Einfachheit auszeichnet und andererseits jeder Muppet-Figur ein individuelles charakteristisches Aussehen verleiht [88]. Dies war der Beginn von weiteren Shows, wie beispielsweise die Sesamstraße (Ende *1969*).

Nach dem Erfolg der Sesamstraße wurde *1976* erstmalig die Muppet Show produziert und ausgestrahlt. Die Serie wurde aufgrund der einzigartigen Charakteristik eines der erfolgreichsten Fernsehprogramme aller Zeiten, worauf mehrere Kinofilme folgten [88].

3.2 Charakteristik der Muppet Show

Die Besonderheit der Muppet Show ergibt sich nicht nur aus dam individuellen Aussehen der einzelnen Muppets, sondern umfasst zusätzlich den Aufbau und die Gestaltung des Bühnenbilds, den Einsatz von Kameraeffekten und die spezielle Handhabung der Puppen. Wie bereits im vorherigen Abschnitt erwähnt, handelt es sich bei Muppets um die Kombination von starren Hand-

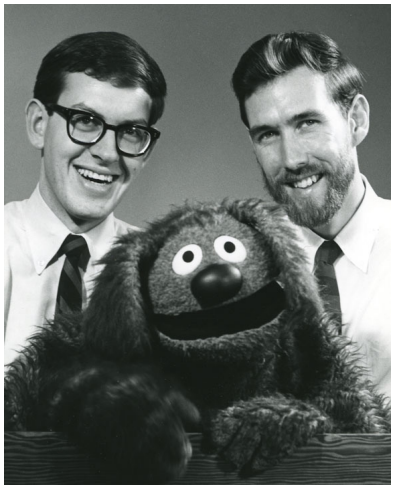


Abbildung 3.3: Das harmonische Zusammenspiel von Frank Oz und Jim Henson führte zum Erfolg der Muppet Show [91].

puppen und typischen Marionetten. Im Allgemeinen ist eine Marionette eine Puppe bestehend aus einzelnen Körperteilen, welche mit Hilfe von dünnen Schnüren bewegt werden können. Im Gegensatz dazu werden Handpuppen nur durch direkte Handbewegungen des Puppenspielers gesteuert. Sowohl Marionetten als auch Handpuppen bestehen meist aus einem starren Material, wodurch beispielsweise nur eine beschränke Ausdrucksweise der Gesichtspartie möglich ist [87]. Da die Muppet Show nur das Medium Fernsehen als Bühne nutzt und verschiedene Kameraeffekte zum Einsatz kommen, werden Muppet-Figuren hauptsächlich aus Schaumstoff hergestellt, um eine flexible Handhabung zu ermöglichen. Somit besitzen Muppets eine größere Bewegungsfreiheit als typische Handpuppen und können zusätzlich verschiedene Gesichtsausdrücke darstellen (z.B. ein sich bewegender Mund während des Sprechens).

Trotz der Verwendung von flexiblen Materialien unterliegen Muppets gewissen Bewegungseinschränkungen. Es wird zwischen zwei verschiedenen Muppet-Typen unterschieden. Die meisten Muppets werden von einer einzigen Person gespielt, wie beispielsweise *Kermit* der Frosch. Dabei setzt der Puppenspieler seine rechte Hand für die Bewegung von Körper, Kopf und Gesichtspartie ein, während er mit der linken Hand Stäbe manövriert, die mit den Händen der Puppe verbunden sind. Im Gegensatz dazu werden für andere Muppets, wie zum Beispiel *Fozzie Bear*, zwei Personen für die Handhabung benötigt. Während ein Puppenspieler sich um die Bewegungen von Oberkörper, Kopf und linkem Arm der Puppe kümmert, ist die andere Person nur für die Darstellung des rechten Arms verantwortlich (siehe Abbildung 3.4) [88].

Eine weitere Besonderheit der Muppet Show ist die spezielle Gestaltung der Charaktere, die nicht nur wie gewöhnliche Puppenfiguren wirken. Während eines langen Entwicklungsprozesses wurde jeder Puppe ein spezieller Charakterzug vermittelt. Wenn ein gewisser Typ fehlte oder noch nicht vorhanden war, wurde eine neue Puppe entwickelt. Die einzelnen Figuren haben zwar ein verschiedenes Aussehen, trotzdem verbindet diese Puppen ein besonderer zugrundeliegender Stil. Dieses besondere Aussehen wird als Muppet-Look bezeichnet. Ein wichtiger Bestandteil ist

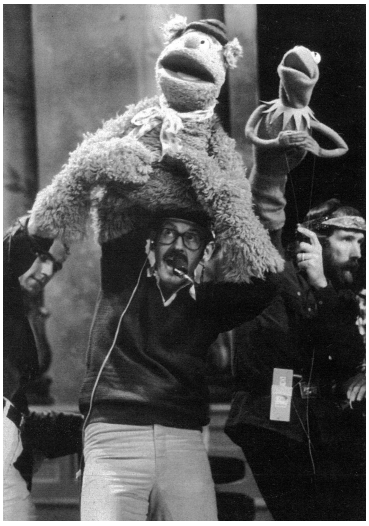


Abbildung 3.4: Frank Oz mit Fozzie und Jim Henson mit Kermit [88].

dabei die Positionierung von Augen, Nase und Mund im Gesicht der Puppen. Zusammen bilden diese Gesichtsteile das sogenannte magische Dreieck, wodurch Muppets auf dem Fernsehbildschirm vom Zuschauer als lebensechte Erscheinung wahrgenommen werden [88].

3.3 Unterteilung der Szenenkategorien

Bei der Muppet Show handelt es sich um eine Show, die sich aus mehreren Sketch-Szenen zusammensetzt. Aus diesem Grund ist es naheliegend, die Kategorisierung basierend auf dem Aufbau der Show vorzunehmen. Jedoch ist eine einheitliche Definition zur Unterteilung der Sketches schwierig zu formulieren. Deshalb sollen folgende Punkte bei der Bildung von Szenenkategorien helfen, um eine effektive Videoklassifikation von Szenen aus der Muppet Show zu ermöglichen:

- **Ähnliches Bühnenbild:**
  In den meisten Fällen wird immer das gleiche oder ein ähnliches Bühnenbild innerhalb einer Szenenkategorie verwendet.

- **Auftretende Muppets:**
  Neben dem Bühnenbild sind vorkommende Muppets und ihr Aussehen (z.B. spezielle Kostümierung) entscheidend für die Definition einer Szenenkategorie.

Basierend auf diesen Definitionsaspekten werden im Anschluss drei bedeutsame Szenenkategorien überblicksartig vorgestellt und kurz beschrieben, die relevant für unser Anwendungsproblem sind:

Kermit's Ansage:

Kermit gehört zu den zentralen Charakteren der Muppet Show. Der lebhafte Frosch führt in jeder Episode durch die gesamte Show und leitet häufig mit herumwirbelnden Armen nachfolgende Sketchszenen ein. Zusätzlich tritt Kermit selbst in einigen Sketchszenen auf und er hat oftmals die Aufgabe in der Muppet Show auftretende Gäste zu interviewen.

Abbildung 3.5: Kermit's Ansage.

Waldorf & Statler:

Die beiden älteren Herren Waldorf und Statler teilen sich eine Loge im Muppet-Theater und äußern gerne ihre harte Kritik über die letzte Sketchszene. Vor allem liefern sie sich mit Fozzie, dem Bären, immer wieder wilde Wortgefechte. Außerdem haben die beiden das letzte Wort am Ende jeder Show.

Abbildung 3.6: Waldorf & Statler.

Tanzszene:

Die Tanzszene spielt in einem glamourösen Ballsaal, wo immer wieder die gleichen Muppets paarweise zu sehen sind. Während sich die Puppen passend zu der Musik tanzend durch das markante Bühnenbild bewegen, amüsieren sie das Publikum mit witzigen Unterhaltungen.

Abbildung 3.7: Tanzszene.

Es existieren noch viele weitere Arten von Sketchszenen [10]. Jedoch werden an dieser Stelle keine weiteren Szenen beschrieben, da in den nächsten Kapiteln zur Erklärung der Funktionsweise und der Evaluierung unseres Prototyps, die Auswahl von wenigen Szenenkategorien ausreichend ist. Abschließend soll nochmals verdeutlicht werden, dass speziell bei der Videosegmentierung von verschiedenen Szenenkategorien die Berücksichtigung der Charakteristik des Videomaterials hilfreich ist, um geeignete audiovisuelle Merkmale und Klassifikationsmodelle zu wählen.

KAPITEL 4

Entwurf & Implementierung

Dieses Kapitel befasst sich mit der Beschreibung des entwickelten Prototyps zur Segmentierung und Klassifikation von Szenen aus der Muppet Show. Zuerst werden die gestellten Anforderungen erläutert und anschließend die Funktionsweise des Systems anhand der einzelnen Phasen erklärt.

4.1 Aufgabenstellung und Anforderungen

Aufgrund des signifikanten Zuwachs von Videoaufnahmen werden Algorithmen benötigt, die in der Lage sind relevante Informationen zu extrahieren und aussagekräftige Beschreibungen von einzelnen Sequenzen zu erstellen. Neben dem Aufzeigen von typischen Featuren, Klassifikationsmodellen und Anwendungsbereichen der Videoklassifikation, ist ein weiteres Ziel dieser Diplomarbeit die Entwicklung eines Prototyps in MATLAB. Videos der Muppet Show sollen in Szenen segmentiert werden, die anschließend basierend auf der Analyse audiovisueller Merkmale in festgelegte Kategorien eingeteilt werden sollen. Dabei werden an den Prototypen folgende Anforderungen gestellt:

- **Segmentierung und Klassifikation von Szenen aus der Muppet Show:**
  Ausgehend vom bereits vorhandenen Videomaterial, soll zunächst eine zeitliche Segmentierung zur Unterteilung von Videos in Sequenzen vorgenommen werden. Anschließend werden aussagekräftige Merkmale benötigt, die in der Lage sind den Inhalt dieser Sequenzen zu erfassen. Dafür können sowohl auditive als auch visuelle Features in Betracht gezogen werden. Abschließend werden für die erfolgreiche Klassifikation geeignete Klassifikationsmodelle benötigt, mit deren Hilfe Sequenzen Kategorien zugeordnet werden können.

- **Möglichst gute Klassifikation:**
  Die zweite Anforderung an den Prototypen bezieht sich auf die Qualität der Resultate. Ziel ist es, eine möglichst hohe Klassifikationsrate zu erreichen. Aus diesem Grund ist es nicht nur notwendig aussagekräftige audiovisuelle Merkmale und Klassifikationsmodelle zu finden, sondern auch geeignete Methoden zur zeitlichen Segmentierung und Bestimmung von Keyframes zu ermitteln.

Für die Realisierung des Prototyps sind die gewählten Merkmale zur Repräsentation des Inhaltes und die Klassifikationsmodelle zur Zuordnung von Szenen in vordefinierte Kategorien entscheidend. In den nachfolgenden Abschnitten wird der Aufbau und die Funktionsweise des entwickelten Prototyps vorgestellt, sowie ein Überblick über jene Features und Klassifikationsmodelle gegeben, die dabei zum Einsatz kommen.

4.2 Aufbau des Prototyps

Die Vorgangsweise der entwickelten Videoklassifikation basiert auf den in Abbildung 4.1 illustrierten Ablauf, wobei zwischen Trainingsphase und Klassifikationsphase unterschieden wird. Während der Trainingsphase werden zuvor manuell ermittelte Ground-Truth-Daten herangezogen, um aus einzelnen Abschnitten von Videos audiovisuelle Merkmale und ihre entsprechende Klassenzugehörigkeit zu extrahieren. Basierend auf diesen Informationen werden verschiedene Klassifikationsmodelle trainiert, die im anschließenden Klassifikationsprozess in der Lage sind, unbekannte Videos korrekt zu klassifizieren.

Innerhalb der Klassifikationsphase wird zunächst eine zeitliche Segmentierung durchgeführt, um Videos in einzelne Shots zu unterteilen. Anschließend gilt es, aus jedem Shot sowohl

visuelle Merkmale zu extrahieren, wie die Analyse der Farbverteilung oder die Beschreibung von lokalen Bereichen aus Bildern, als auch eine Segmentierung des Audiosignals in Musik, Sprache und Umgebungsgeräusche vorzunehmen. Danach werden Shots anhand der berechneten Features und der trainierten Klassifikationsmodelle den entsprechenden Szenenkategorien zugeordnet. Im letzten Schritt werden Shots basierend auf ihren Inhalten zu Szenen zusammengefasst. Wie die einzelnen Phasen im Detail ablaufen, wird in den nächsten beiden Abschnitten erläutert.

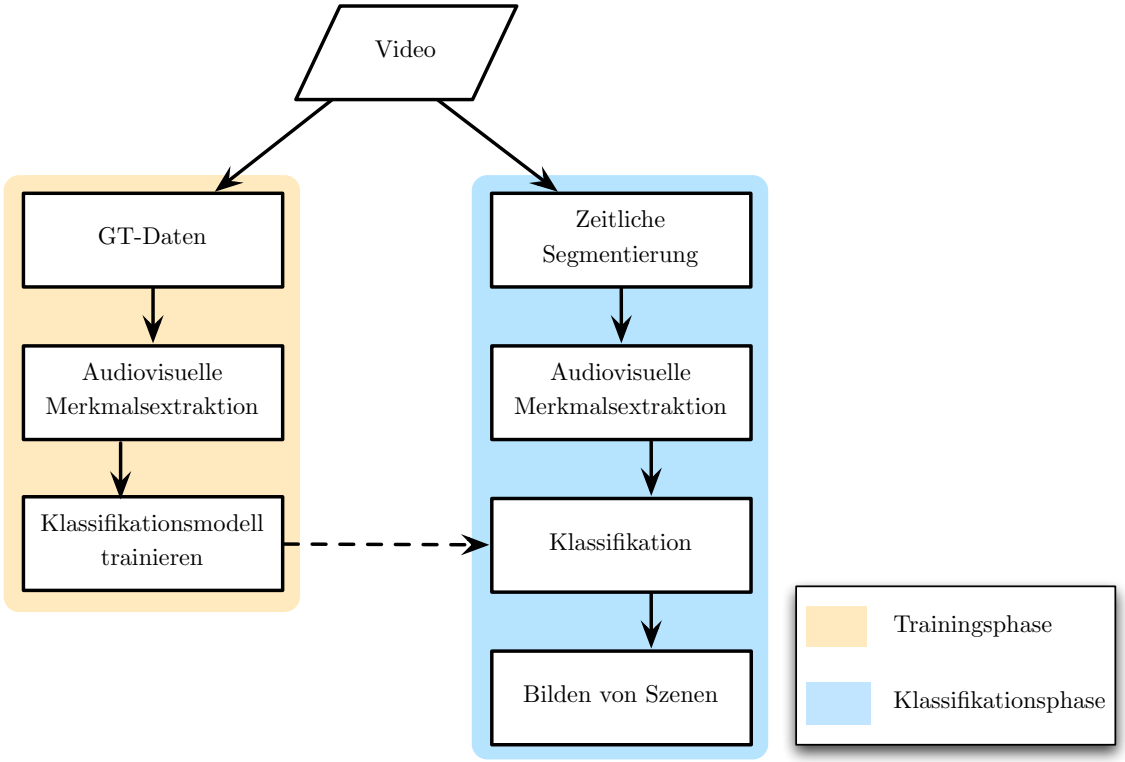


Abbildung 4.1: Aufbau des Prototyps zur Klassifikation von Videos.

4.3 Trainingsphase

Um eine effektive Videoklassifikation von Szenen aus der Muppet Show zu ermöglich, werden zunächst in einem iterativen Trainingsprozess aussagekräftige Merkmale aus bereits bekannten Videosequenzen extrahiert, die anschließend für das Trainieren von Klassifikationsmodellen herangezogen werden. In Abbildung 4.2 wird der Ablauf der Trainingsphase dargestellt. Ausgehend von Videos werden zunächst Ground-Truth-Daten ermittelt. Diese Informationen werden danach für den eigentlichen Trainingsprozess verwendet, wobei sich dieser Prozess in folgende drei Teile gliedern lässt:

- Visuelles Training der einzelnen Shots

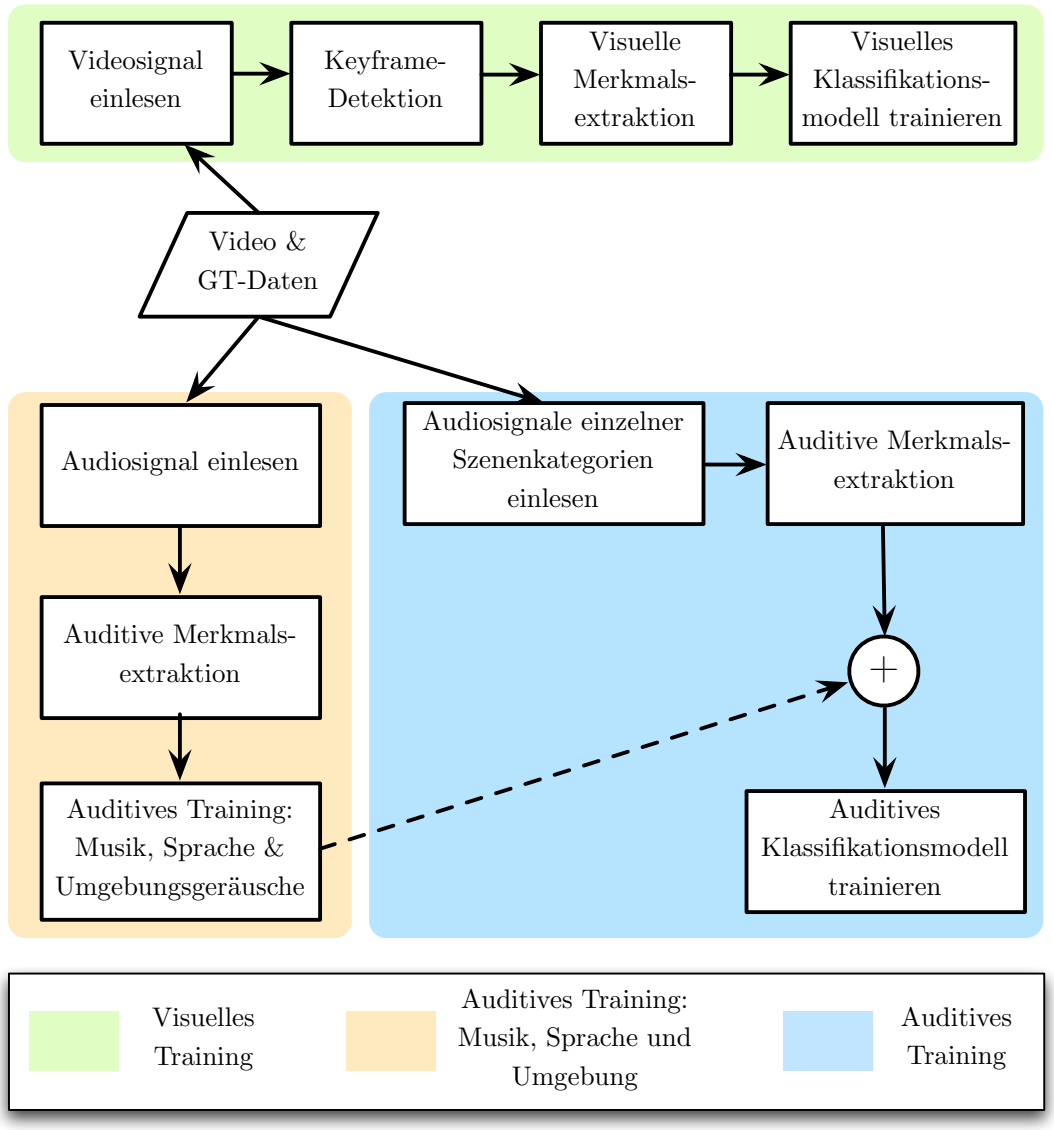


Abbildung 4.2: Ablauf der Trainingsphase.

- Auditives Training zur Unterscheidung von Musik, Sprache und Umgebungsgeräuschen
- Auditives Training der einzelnen Shots

Für das visuelle Training werden für die einzelnen Shots aussagekräftige Keyframes bestimmt. Aus diesen Frames werden verschiedene Merkmale extrahiert, die anschließend für das Training der visuellen Klassifikationsmodelle herangezogen werden (siehe Abschnitt 4.3.2). Im Gegensatz dazu setzt sich das auditive Training aus zwei separaten Schritten zusammen. Zunächst wird mit Hilfe auditiver Features ein Modell zur Klassifikation von Musik, Sprache, Stil-

le und Umgebungsgeräuschen trainiert (siehe Abschnitt 4.3.3).Im zweiten Schritt wird dieses Klassifikationsmodell herangezogen, um für alle Szenenkategorien den prozentuellen Anteil der vier Geräuscharten zu ermitteln (siehe Abschnitt 4.3.4).

4.3.1 Ground-Truth-Daten

Zunächst werden für das Training und später für die Evaluierung Informationen über die Klassenzugehörigkeit von einzelnen Abschnitten der Videos benötigt. Diese Informationen werden in einem manuellen Prozess extrahiert und sollen möglichst exakt sein, da am Ende sowohl die Qualität der Klassifikationsresultate, als auch die Aussagekraft der Evaluierung abhängig von der Genauigkeit der Ground-Truth-Daten sind. Für die angestrebte Videoklassifikation werden folgende GT-Daten ermittelt (siehe Abbildung 4.3):

- **Informationen über einzelne Shot-Übergänge (*Shot Transitions*):**
  - Startpunkt (in Frames)
  - Endpunkt (in Frames)
  - Dauer des Überganges (in Frames)
  - Art des Überganges (*cut, fade, wipe oder dissolve*)
- **Informationen über einzelne Shots (*Shot Information*) und ihre Inhalte:**
  - Startpunkt (in Frames)
  - Endpunkt (in Frames)
  - Dauer der Shots (in Frames)
  - Zugehörige Szenenkategorie
- **Informationen über einzelne Szenen (*Scene Information*) und ihre Inhalte:**
  - Startpunkt (in Frames)
  - Endpunkt (in Frames)
  - Dauer der Szenen (in Frames)
  - Zugehörige Szenenkategorie

Zusätzlich werden für die spätere Analyse des Audiosignals einzelne Bereiche eines Videos entnommen, die eindeutig einer auditiven Kategorie, wie Musik, Sprache oder Umgebungsgeräusch, zugeordnet werden können. Diese Bereiche und ihre Kategorien werden ebenfalls in den GT-Daten festgehalten, um später ein auditives Klassifikationsmodell trainieren zu können. Die ermittelten Ground-Truth-Daten repräsentieren das tatsächliche Auftreten von relevanten Szenenkategorien und dienen sowohl für das audiovisuelle Training, als auch zur späteren Evaluierung des Prototyps.

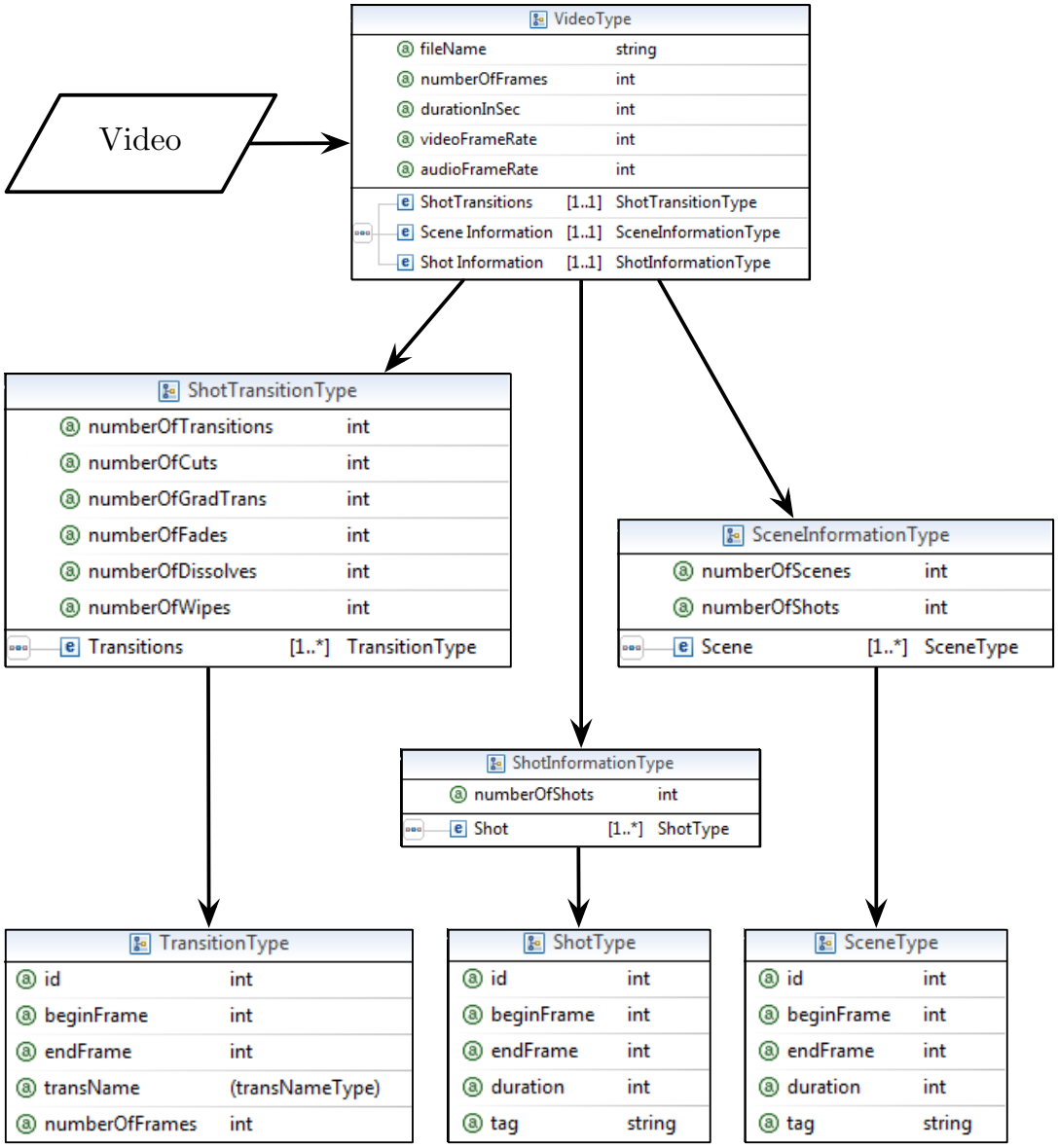


Abbildung 4.3: Struktureller Aufbau der Ground-Truth-Daten.

4.3.2 Visuelles Training einzelner Shots

Für das visuelle Training werden mit Hilfe der Informationen aus den GT-Daten einzelne Shots der vorhandenen Videos betrachtet und die entsprechenden Frames eingelesen. Dabei können abhängig von der Dauer der Shots enorme Datenmengen anfallen, da jede Sekunde des Videos aus *25* Frames besteht. Aufgrund des entstehenden Speicherbedarfs sind mehrere Lesevorgänge mit einer geringeren Anzahl an Frames notwendig. Jedoch können bei Frames visuelle Artefakte auftreten, die sich am Anfang eines solchen Lesevorgangs befinden, wodurch die weitere

Abbildung 4.4: Vergleich zwischen tatsächlichem Frame und Frame mit fehlerhaftem Bildinhalt.

Analyse nicht möglich wäre (siehe Abbildung 4.4). Um diese Bildfehler zu vermeiden, werden zusätzliche Frames geladen, die sich vor dem entsprechenden Leseabschnitt befinden und somit nicht weiter benötigt werden.

Nach Abschluss des Lesevorgangs wird für jeden Frame ein Farbhistogramm berechnet (siehe Abschnitt 2.4.2), wobei jeder Farbkanal in *16* Bins unterteilt wird. Insgesamt ergibt sich daraus ein Histogramm mit *16x3* Farbwerten. Zusätzlich wird jeder Frame zu einem Grauwertbild transformiert und die vorhandene Variabilität der Intensitätswerte bestimmt. Diese beiden Informationen (Farbhistogramm und Variabilität der Grauwerte) werden anstelle der Frames gespeichert und anschließend für die Keyframe-Detektion herangezogen. Für eine framebasierende visuelle Merkmalsextraktion werden aussagekräftige Frames aus einem Shot benötigt. Wie bereits in Abschnitt 2.3.1 erklärt, ist neben der Forderung, möglichst aussagekräftige Frames zu bestimmen, vor allem die Anzahl der Keyframes entscheidend. Zur Bestimmung der Keyframes wird die Ähnlichkeit von Frames innerhalb eines Shots analysiert. Dazu werden die zuvor berechneten Farbhistogramme betrachtet und ihre Differenzen berechnet. Anstelle der Histogram Intersection wird die in der Praxis oft eingesetzte χ^2-Distanz $d_{\chi^2}(h_1, h_2)$ herangezogen [93, 32]:

$$d_{\chi^2}(h_1, h_2) = \sum_{i=1}^{n} \frac{(h_1(i) - h_2(i))^2}{h_1(i) + h_2(i)} \tag{4.1}$$

wobei $h_1(i)$ dem i-ten Bin aus dem Histogramm des ersten Frames entspricht und n die Anzahl der Bins repräsentiert. Zur Approximation der Ähnlichkeit $a(h_1, h_2)$ wird der Kehrwert der Differenz $d_{\chi^2}(h_1, h_2)$ gebildet:

$$a(h_1, h_2) = d_{\chi^2}(h_1, h_2)^{-1} \tag{4.2}$$

Auf diese Weise werden die Ähnlichkeitswerte zwischen allen möglichen Framepaaren innerhalb eines Shots berechnet und in Form einer Kostenmatrix abgespeichert. Je höher der Wert eines Eintrages in dieser Matrix ist, desto ähnlicher sind sich die entsprechenden Frames. Aus der Kostenmatrix wird anschließend ein gerichteter Graph konstruiert, der sich aus den einzelnen Frames als Knoten und den Ähnlichkeitswerten als zugehörigen Kantengewichten zusammensetzt (siehe Abbildung 4.5).

Ähnlich wie in Abschnitt 2.3.2 bereits erklärt, werden die gesuchten Keyframes durch jene Knoten im Graph repräsentiert, die Teil des kürzesten Pfades sind. Abhängig von den auftretenden Änderungen innerhalb des Shots wird eine geeignete Anzahl an repräsentativen Keyframes ermittelt, wobei der erste und letzte Frame immer Teil des kürzesten Pfades sind. Diese graphenbasierte Methode ermöglicht das Hinzufügen von weiteren Keyframes, indem der Graph

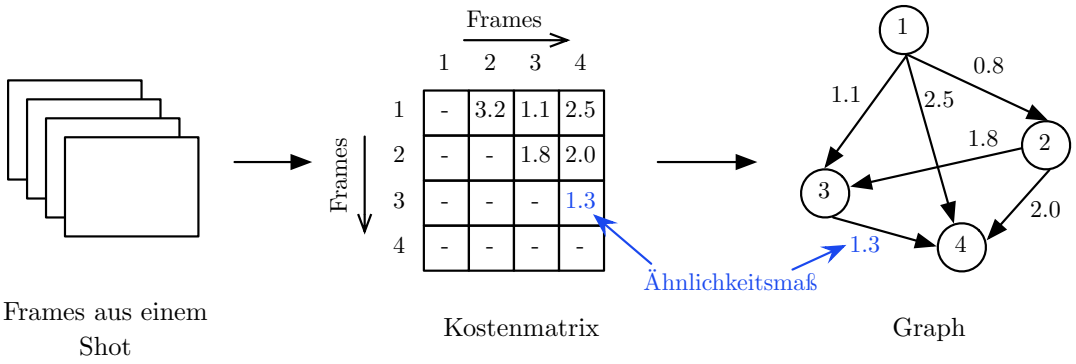


Abbildung 4.5: Ablauf der Keyframe-Detektion.

aufgeteilt wird, sobald die Knotenmenge einen zuvor festgelegten Schwellwert übersteigt. In diesem Fall wird der Graph jeweils nach *125* Knoten aufgeteilt, wobei dies der Dauer von fünf Sekunden entspricht. Abschließend werden für alle Teilgraphen die kürzesten Pfade und die somit verbundenen Keyframes bestimmt.

Ausgehend von Keyframes sind sowohl globale als auch lokale visuelle Features geeignet, um aussagekräftige Beschreibungen des Bildinhaltes zu erstellen. Nach der Anwendung eines Gauß-Filters zur Entfernung von Bildrauschen, werden visuelle Merkmale berechnet (siehe Abbildung 4.6). Im Anschluss wird erklärt, wie sich diese Features zusammensetzen.

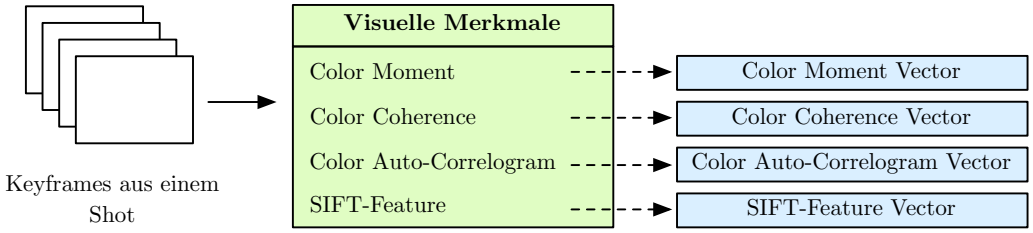


Abbildung 4.6: Überblick der visuellen Merkmalsextraktion.

Color Moments (CM):
Für die globale Analyse von Farbinformationen werden für jeden Farbkanal die ersten drei statistischen Momente ausgewertet (siehe Abschnitt 2.4.2), da diese in der Lage sind, die Charakteristik der Farbverteilung zu repräsentieren.

Color Coherence Vector (CCV):
Wie bereits in Abschnitt 2.4.3 aufgezeigt, ist die alleinige Betrachtung von Farbinformationen nicht immer ausreichend für eine inhaltsbasierte Analyse. Aus diesem Grund wird für jeden Frame der CCV ermittelt. Für die Berechnung des Features wird zunächst ein Farbhistogramm gebildet, welches aus *512* Bins besteht. Anschließend wird für jeden Bin der prozentuelle Anteil von kohärenten und nicht-kohärenten Pixeln berechnet, wobei die kohärenten Regionen eine Mindestgröße von *10%* der Bildgröße aufweisen müssen.

Color Auto-Correlogram (CAC):
Als weiteres visuelles Feature wird das Color Auto-Correlogram eingesetzt. Zunächst werden aus Frames wieder Farbhistogramme bestehend aus *512* Bins gebildet. Anschließend wird das CAC berechnet, wie es in Abschnitt 2.4.4 erklärt wurde. Um dabei eine bessere Beschreibung des visuellen Inhaltes zu erlangen, werden mehrere Distanzwerte (*1*, *3*, *5* und *7*) betrachtet. Die vier daraus resultierenden CACs werden abschließend in Form eines eindimensionalen Feature-Vektors zusammengefasst.

SIFT-Features:
Neben den bisherigen globalen Merkmalen werden mit Hilfe von SIFT-Features auch kleine lokale Bereiche von Frames analysiert. Bereits in verschiedenen Bereichen der Bildanalyse hat sich gezeigt, dass SIFT-Features in Verbindung mit einem sogenannten *Bag-of-Words*-Modell (BoW) aussagekräftige Beschreibungen des Bildinhaltes erzeugen [17, 14]. Dabei werden die *128*-dimensionalen SIFT-Features zu visuellen Wörter gruppiert, basierend auf der Verteilung im Feature-Space. Eine anschließende Auswertung der auftretenden Wörter liefert die Beschreibung des Bildinhaltes. Der dafür benötigte Trainingsprozess setzt sich aus folgenden zwei Schritten zusammen:

(1) **Extrahieren von SIFT-Features aus Keyframes aller Videos und Bilden des visuellen Vokabulars:**
Für die Bildung des visuellen Vokabulars werden aus allen Keyframes SIFT-Features extrahiert, unabhängig davon welchen Szenenkategorien die Frames angehören. Dabei wird anstelle der Detektion von interessanten Punkten (siehe Abschnitt 2.4.5), der ganze Frame gleichmäßig in x- und y-Richtung abgetastet. An jeder Stelle werden SIFT-Deskriptoren berechnet, die anschließend im Feature-Space betrachtet werden. Mit Hilfe von K-Means (siehe Abschnitt 2.6.3) werden die SIFT-Features zu Referenzvektoren gruppiert, die als visuelle Wörter bezeichnet werden (siehe Abbildung 4.7). Zuletzt wird aus dem erstellten visuellen Vokabular, welches hier aus *1000* Wörtern besteht, ein KD-Baum erstellt (siehe Abschnitt 2.6.4), um im späteren Verlauf einzelne SIFT-Features den entsprechenden visuellen Wörtern zuweisen zu können.

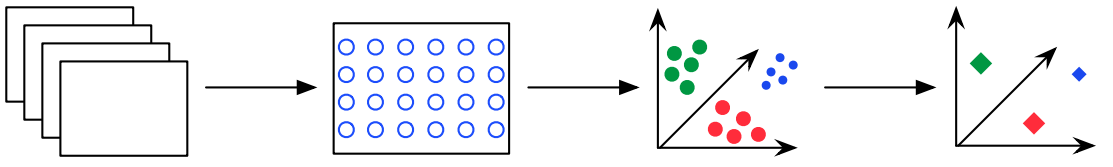


Abbildung 4.7: Ablauf zur Bildung eines visuellen Vokabulars.

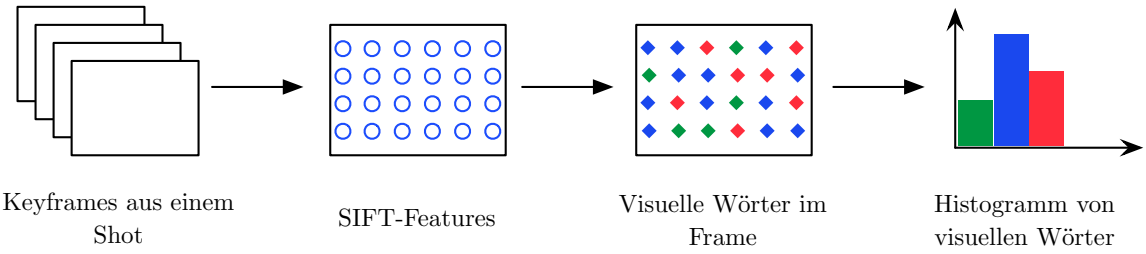


Abbildung 4.8: Beschreibung des Bildinhaltes mit Hilfe von SIFT-Features.

(2) **Bestimmen der Häufigkeit des Auftretens von visuellen Wörtern innerhalb von Keyframes:**
Nach der Erstellung des visuellen Vokabulars werden die SIFT-Features einzelner Keyframes erneut betrachtet. Diese Features werden mit Hilfe des zuvor erstellten KD-Baums in visuelle Wörter umgewandelt und die auftretende Häufigkeit wird in einem Histogramm festgehalten (siehe Abbildung 4.8). Für alle Keyframes eines Shots werden solche Histogramme berechnet, die zur Beschreibung des Bildinhaltes dienen.

Die extrahierten visuellen Features werden abschließend für das Training von verschiedenen Klassifikationsmodellen eingesetzt. Wie bereits erläutert, wirkt sich die Wahl der eingesetzten Features und Klassifikationsmodelle auf die Qualität der Klassifikationsresultate aus. Programme wie WEKA (*Waikato Environment for Knowledge Analysis*) sind in der Lage, die Aussagekraft einzelner Features zu analysieren und helfen bei der Wahl von geeigneten Klassifikationsmodellen [94]. Basierend auf dem Einsatz von WEKA und den draus gewonnenen Erkenntnissen werden folgende Kombinationen von Features und Klassifikationsmodellen trainiert, die im späteren Klassifikationsprozess verwendet werden (siehe Tabelle 4.1). Neben dem Einsatz von visuellen Informationen ist die Analyse des Audiosignals für eine erfolgreiche Klassifikation entscheidend.

Bezeichnung der Features	Gewähltes Klassifikationsmodell	Bemerkung
Color Moments	Nearest Neighbour	Verwendung der euklidischen Distanz
Color Coherence Vector	Nearest Neighbour	Verwendung der euklidischen Distanz
Color Auto-Correlogram	Support Vector Machine	mit linearer Kernelfunktion
SIFT-Features	Support Vector Machine	mit linearer Kernelfunktion

Tabelle 4.1: Visuelle Feature und Klassifikationsmodelle.

4.3.3 Auditives Training zur Klassifikation von Musik, Sprache und Umgebungsgeräuschen

Ähnlich wie bei der visuellen Trainingsphase werden mit Hilfe der GT-Daten einzelne Audiosequenzen aus einem Video eingelesen. Bei diesem Lesevorgang werden, wie beim visuellen Lesevorgang, zusätzliche Daten geladen, um mögliche Signalstörungen zu vermeiden. Jedoch können aus Sequenzen, die kürzer als eine Sekunde sind, keine verwendbaren Daten ausgelesen werden. Es ist experimentell festgestellt worden, dass extrahierte Features aus solchen kurzen Audiosignalen zu wenige diskriminative Informationen enthalten, die für die Klassifikation nicht ausreichend sind. Aus diesem Grund werden kurze Abschnitte nicht weiter betrachtet. Das geladene Audiosignal wird anschließend in überlappende Fenster mit einer Größe von *0,2* Sekunden (entspricht *5* Frames) unterteilt. Bevor aus den einzelnen Fenstern aussagekräftige Features berechnet werden, wird überprüft, ob der Signalanteil für eine Merkmalsextraktion ausreichend ist (siehe Abbildung 4.9).

Für diese Entscheidung wird ein Schwellwert t_{sil} bestimmt. Wie experimentelle Analysen gezeigt haben, kann t_{sil} basierend auf den Amplitudenwerten des gesamten Audiosignals definiert werden:

$$t_{sil} = mean(x^2) * \left(1 - \frac{mean(x^2)}{max(x^2)}\right) \qquad \text{mit} \quad x \in [-1, 1] \tag{4.3}$$

wobei x den Amplitudenwerten des Audiosignals entspricht. Wird dieser Schwellwert nicht überschritten, so wird das entsprechende Fenster als Stille definiert und es wird keine weitere Merkmalsextraktion durchgeführt. Aus den übrigen Audiofenstern werden auditive Merkmale berechnet, wobei für die Klassifikation von Musik, Sprache und Umgebungsgeräuschen folgende Features betrachtet werden:

- **Zero Crossing Rate (ZCR)**
- **Root Mean Square (RMS)**
- **Mel-Frequency Cepstrum Coefficients (MFCC)**

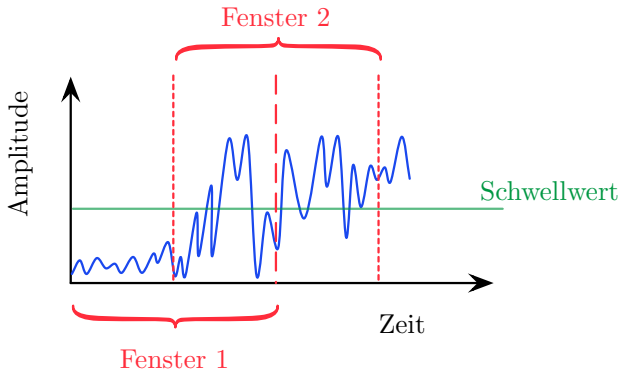


Abbildung 4.9: Unterteilung des Audiosignals in Fenster und Überprüfen auf Stille.

- **Verhältnis von RMS und ZCR:**
  Ein Audiofenster wird erneut in *20* nicht überlappende Abschnitte unterteilt und für jeden dieser Abschnitte wird RMS und ZCR berechnet. Anschließend wird das Verhältnis von RMS und ZCR über das gesamte Audiofenster ermittelt [66]:

$$R_{RZ} = \frac{\sum_{i=1}^{20} RMS(i) * ZCR(i)}{2 * RMS_x - RMS_n - RMS_m} \tag{4.4}$$

  wobei RMS_x den Maximal-, RMS_n den Minimal- und RMS_m dem Medianwert aller RMS-Werte $RMS(i)$ der Teilfenster i repräsentieren.

- **Verhältnis von Mittelwert und Varianz des RMS:**
  Aus den *20* nicht überlappenden Abschnitten wird der RMS berechnet und das Verhältnis von Mittelwert und Varianz über das gesamte Audiofenster ausgewertet [66]:

$$R_{RMS_{mv}} = \frac{var(RMS_{sub})}{mean(RMS_{sub})} \tag{4.5}$$

  wobei RMS_{sub} die RMS-Werte aus den *20* Abschnitten darstellt.

Abschließend werden die einzelnen auditiven Merkmale mit Hilfe von WEKA analysiert und auf ihre Entscheidungskraft überprüft. Zuletzt werden die Features für das Trainieren der SVM herangezogen, um eine Klassifikation von Musik, Sprache und Umgebungsgeräuschen zu ermöglichen. Dabei wird eine lineare Kernelfunktion eingesetzt, da ähnliche Klassifikationsraten erreicht werden können wie bei Verwendung komplexer polynomieller Kernelfunktionen.

4.3.4 Auditives Training einzelner Shots

Nach abgeschlossener Trainingsphase zur Klassifikation von Musik, Sprache und Umgebungsgeräuschen werden erneut die GT-Daten herangezogen, um Audiosignale aus einzelnen Shots einlesen zu können. Diese Signale werden wieder in überlappende Fenster unterteilt und für die auditive Merkmalsextraktion herangezogen. Dabei werden dieselben Features wie zuvor in Abschnitt 4.3.3 verwendet. Anschließend wird eine erste Klassifikation der Audiosignale durchgeführt. Für jedes Fenster wird bestimmt, ob es sich dabei um Stille, Musik, Sprache oder Umgebungsgeräusche handelt. Auf diese Weise wird für jeden Shot und somit für jede Szenenkategorie der prozentuelle Anteil der vier Geräuscharten bestimmt. Jede Szenenkategorie wird durch ein Histogramm repräsentiert, welches die gemittelten Anteile von Stille, Musik, Sprache und Umgebungsgeräusche enthält. Es ist nicht nötig, ein spezielles Klassifikationsmodell zu trainieren, da die spätere auditive Analyse eines unbekannten Shots auf dem Vergleich von Histogrammen basiert.

Nach Abschluss der visuellen und auditiven Trainingsphase ist das System in der Lage, innerhalb der Klassifikationsphase aus Videos der Muppet Show einzelne Szenenkategorien zu segmentieren. Im nächsten Abschnitt werden die einzelnen Schritte der Klassifikationsphase näher erläutert.

4.4 Klassifikationsphase

In der Klassifikationsphase werden aus einzelnen Videos audiovisuelle Merkmale extrahiert und analysiert. Diese Phase besteht aus mehreren Schritten, die ähnlich jenen aus dem Training sind (siehe Abbildung 4.10). Zunächst wird die zeitliche Segmentierung durchgeführt und anschließend eine Keyframe-Extraktion vorgenommen, um aussagekräftige Frames zu bestimmen. Im nächsten Schritt werden audiovisuelle Merkmale extrahiert und mit Hilfe der trainierten Klassifikationsmodelle wird für jeden Shot die entsprechende Szenenkategorie bestimmt. Zuletzt werden jene Shots, die der gleichen Kategorie angehören und innerhalb eines Zeitbereiches liegen, zu einer Szene zusammengefasst. Im Anschluss werden die einzelnen Schritte näher beschrieben.

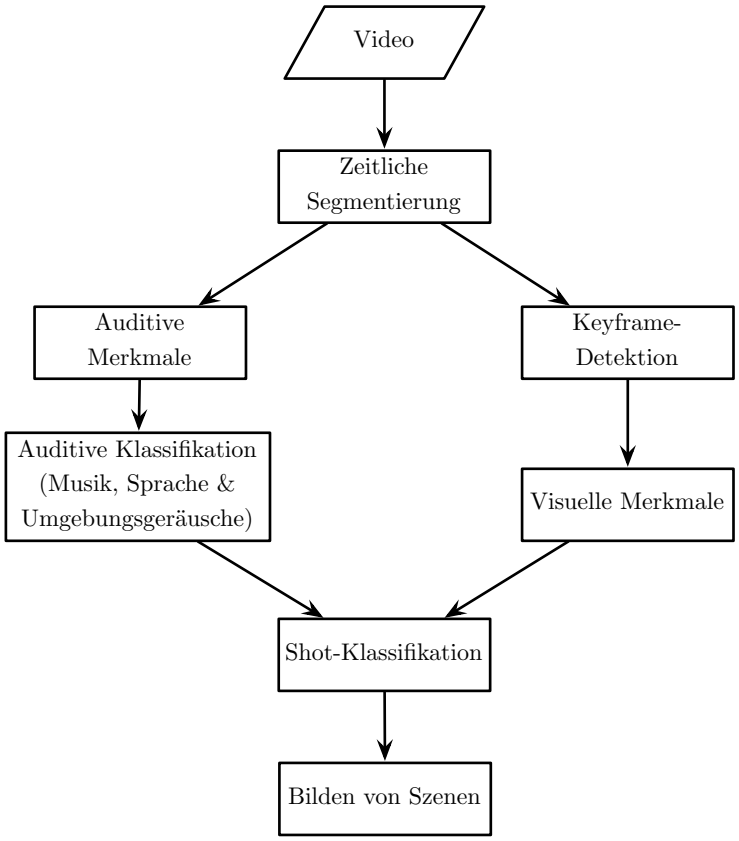


Abbildung 4.10: Aufbau der Klassifikationsphase.

4.4.1 Zeitliche Segmentierung

Zunächst werden Videos mit Hilfe einer zeitlichen Segmentierung in einzelne Shots unterteilt. Für diese Segmentierung wird ein simpler Twin-Comparison-Ansatz verwendet. Dabei werden aus aufeinanderfolgenden Frames Farbhistogramme berechnet, die sich aus *16* Bins pro Farb-

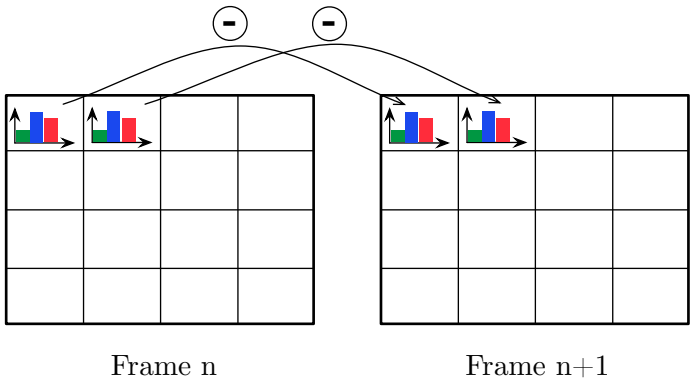


Abbildung 4.11: Aufteilen zweier Frames in Zellen und Berechnung der Differenzen.

kanal zusammensetzen. Wie in Abschnitt 2.2.2 erklärt, werden anschließend die Unterschiede zwischen diesen Histogrammen berechnet, wobei für die Differenzbildung in erster Linie Histogram Intersection eingesetzt wird.

Zusätzlich besteht die Möglichkeit, eine alternative Methode zur zeitlichen Segmentierung einzusetzen. Der Unterschied zum bereits vorgestellten Ansatz liegt in der Bildung der Differenzen von Farbhistogrammen. Dabei werden Frames nicht global betrachtet, sondern zunächst in *4x4* Zellen aufgeteilt. Anschließend wird für jede Zelle ein Farbhistogramm gebildet (siehe Abbildung 4.11). Mit Hilfe der χ^2-Distanz werden die Unterschiede von korrespondierenden Zellen aus zwei benachbarten Frames berechnet [95]. Zur Kompensation von auftretender Bewegung innerhalb der Shots werden die acht größten Differenzwerte verworfen. Die restlichen Werte werden aufsummiert und dienen als Differenzmaß zweier benachbarter Frames.

Unabhängig davon welche der beiden Methoden zur Differenzbildung zum Einsatz kommt, werden alle auftretenden Differenzwerte innerhalb eines gesamten Videos betrachtet, um die für Twin Comparison benötigten Schwellwerte T_h und T_l zu bestimmen. Nachdem aus den Videos einzelne Shots ermittelt worden sind, werden audiovisuelle Merkmale für die bevorstehende Klassifikation extrahiert.

4.4.2 Audiovisuelle Merkmalsextraktion

Nach der Bestimmung der Shots aus den Videos gilt es, aussagekräftige audiovisuelle Merkmale zu extrahieren. Dieser Vorgang teilt sich wieder in mehrere Schritte auf, welche bereits in den vorherigen Abschnitten erläutert worden sind:

- **Visuell:**

  (1) Bestimmen von Keyframes

  (2) Extrahieren von visuellen Merkmalen

- **Auditiv:**

  (1) Unterteilung des Audiosignals in überlappende Fenster

  (2) Extrahieren von auditiven Merkmalen

  (3) Bestimmen der Anteile von Musik, Sprache, Stille und Umgebungsgeräuschen

Die extrahierten Merkmale dienen zur Repräsentation des Inhaltes innerhalb eines Shots und ermöglichen die anschließende Klassifikation.

4.4.3 Klassifikation von Shots

Die Ermittlung der Kategorien von Shots erfolgt durch ein Voting, basierend auf den extrahierten Merkmalen. Für jedes trainierte Klassifikationsmodell werden die entsprechenden Features herangezogen und unabhängig voneinander klassifiziert. Dabei wird zwischen folgenden zwei Arten der Klassifikation unterschieden:

- **Visuelle Klassifikation:**
  Die entsprechenden visuellen Merkmale der Keyframes aus dem aktuellen Shot werden für jedes visuelle Klassifikationsmodell einzeln herangezogen und analysiert. Aus den jeweiligen Klassifikationsergebnissen wird der prozentuelle Anteil bezüglich aller Szenenkategorien berechnet und in einer Voting-Matrix abgespeichert (siehe Abbildung 4.12). Jeder Eintrag in dieser Matrix repräsentiert die Wahrscheinlichkeit, dass der aktuelle Shot zu einer gewissen Szenenkategorie gehört, bei Verwendung eines einzigen spezifischen Klassifikationsmodells.

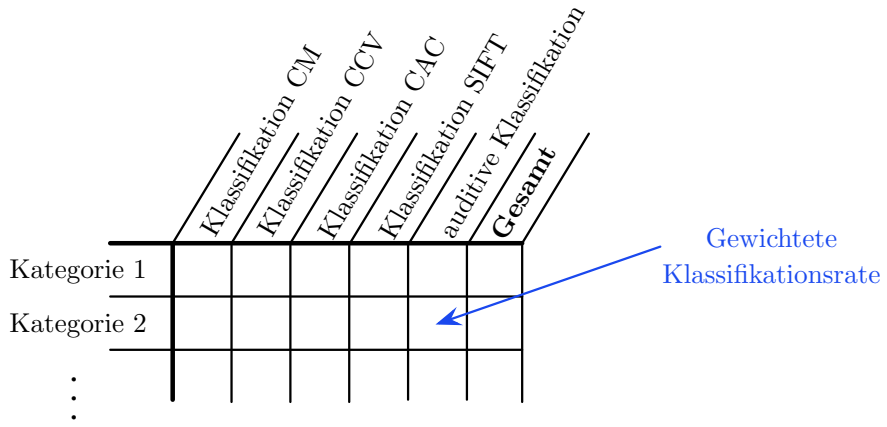


Abbildung 4.12: Aufbau der Voting-Matrix.

- **Auditive Klassifikation:**
  Für jeden Shot ist bereits der prozentuelle Anteil von Musik, Sprache, Stille und Umgebungsgeräuschen bekannt (siehe Abschnitt 4.4.2). Ausgehend von diesen Informationen,

wird bei der endgültigen auditiven Klassifikation die Verteilung der vier Audiokategorien ausgewertet. Mit Hilfe der Histogram Internsection werden die Ähnlichkeiten zwischen der Verteilung des aktuellen Shots und den Verteilungen aller Szenenkategorien aus der Trainingsphase berechnet. Abschließend werden die Ergebnisse wieder in der Voting-Matrix festgehalten.

Die Resultate in der Voting-Matrix werden anschließend einer Gewichtung unterzogen, die auf Erkenntnisse heuristischer Analysen weniger Videos basiert, um eine möglichst hohe Generalisierungsfähigkeit zu erreichen. So wird den visuellen Klassifikationsmodellen ein größeres Vertrauen ausgesprochen, da die Ermittlung der korrekten Szenenkategorie basierend auf auditive Information, wie Musik, Sprache und Umgebungsgeräuschen, nicht immer möglich ist. Im letzten Schritt werden die gesammelten Resultate über die einzelnen Klassifikationsmodelle aufsummiert. Der aktuelle Shot wird jener Szenenkategorie zugewiesen, die den größten Gesamtwert aufweist. Jeder Shot wird innerhalb der Klassifikationsphase einer Szenenkategorie zugeordnet und mehrere Shots bilden eine gemeinsame Szene, abhängig vom zeitlichen Auftreten der Kategorien.

4.4.4 Gruppierung von Shots zu Szenen

Nach der Klassifikation von einzelnen Shots werden diese zu gemeinsamen Szenen zusammengefasst. Wie bereits zuvor erwähnt, können aus Sequenzen mit kurzer Dauer keine aussagekräftigen Features berechnet werden, weshalb keine Klassifikation möglich ist. In diesem Fall werden den kurzen Shots jene Szenenkategorien zugewiesen, die der nächste vorkommende Shot aufweist. Anschließend gilt es für die Bildung von Szenen folgende zwei Regeln zu beachten (siehe Abbildung 4.13):

- Aufeinanderfolgende Shots, die zur selben Szenenkategorie gehören, werden zu einer gemeinsamen Szene gruppiert.
- Werden zwei Shots derselben Kategorie durch eine andere Szenensequenz unterbrochen, so werden alle betroffenen Shots unabhängig von der Szenenkategorie zu einer gemeinsamen Szene gruppiert, wenn es sich um eine geringe Unterbrechung handelt. Dieser Fall wird im weiteren Verlauf als Dialogszene bezeichnet.

In die Entscheidung, ob die Unterbrechung geringfügig ist oder nicht, fließt sowohl die Anzahl der dazwischenliegenden Shots, als auch die Dauer der Unterbrechung in Sekunden ein. In

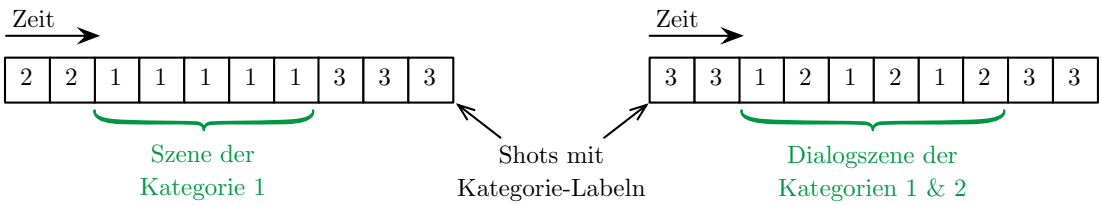


Abbildung 4.13: Zusammenfügen von Shots zu einer simplen Szene und Dialogszene.

Abbildung 4.14 wird dieser Vorgang dargestellt. Ausgehend von einem Shot (rot) werden jeweils die drei vorhergehenden und nachfolgenden Shots (blau) betrachtet und zu einer Szene gruppiert, falls beide Kriterien erfüllt werden. Auf diese Art werden aus einem Video die einzelnen Szenen verschiedener Kategorien gebildet und abschließend in der graphischen Benutzeroberfläche des Prototyps visualisiert.

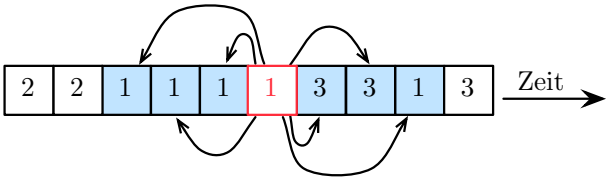


Abbildung 4.14: Betrachten von umliegenden Shots zur Bildung von Szenen.

KAPITEL 5

Ergebnisse

In diesem Kapitel werden die Ergebnisse der entwickelten Videoklassifikation präsentiert. Neben der Vorstellung der graphischen Benutzeroberfläche werden verschiedene Auswertungen erklärt, um die Wahl der audiovisuellen Merkmale zu bewerten und die Qualität der Klassifikationsresultate zu bestimmen.

5.1 Graphische Benutzeroberfläche

Basierend auf der Aufgabenstellung und den Anforderungen (siehe Abschnitt 4.1) wurde im Verlauf dieser Arbeit ein Prototyp zur Segmentierung und Klassifikation von Szenen aus der Muppet Show entwickelt. In Abbildung 5.1 wird die graphische Benutzeroberfläche des Systems dargestellt, wobei die Kreise der Identifikation einzelner Elemente dienen.

Nach dem Start des Prototyps wird zunächst überprüft, ob sich Videos in einem zuvor gewählten Ordner befinden ①. Anschließend kann zwischen verschiedenen Szenenkategorien gewählt werden, die aus den Videos segmentiert werden sollen ②. Als zusätzliche Option kann dabei angegeben werden, ob alle vorhandenen Videos herangezogen werden sollen oder ob sich die Analyse nur auf eine spezifische Videodatei beschränken soll ③. Nachdem der Klassifikationsvorgang gestartet worden ist, werden die einzelnen Schritte des Klassifikationsprozesses (siehe Abschnitt 4.4) durchgeführt.

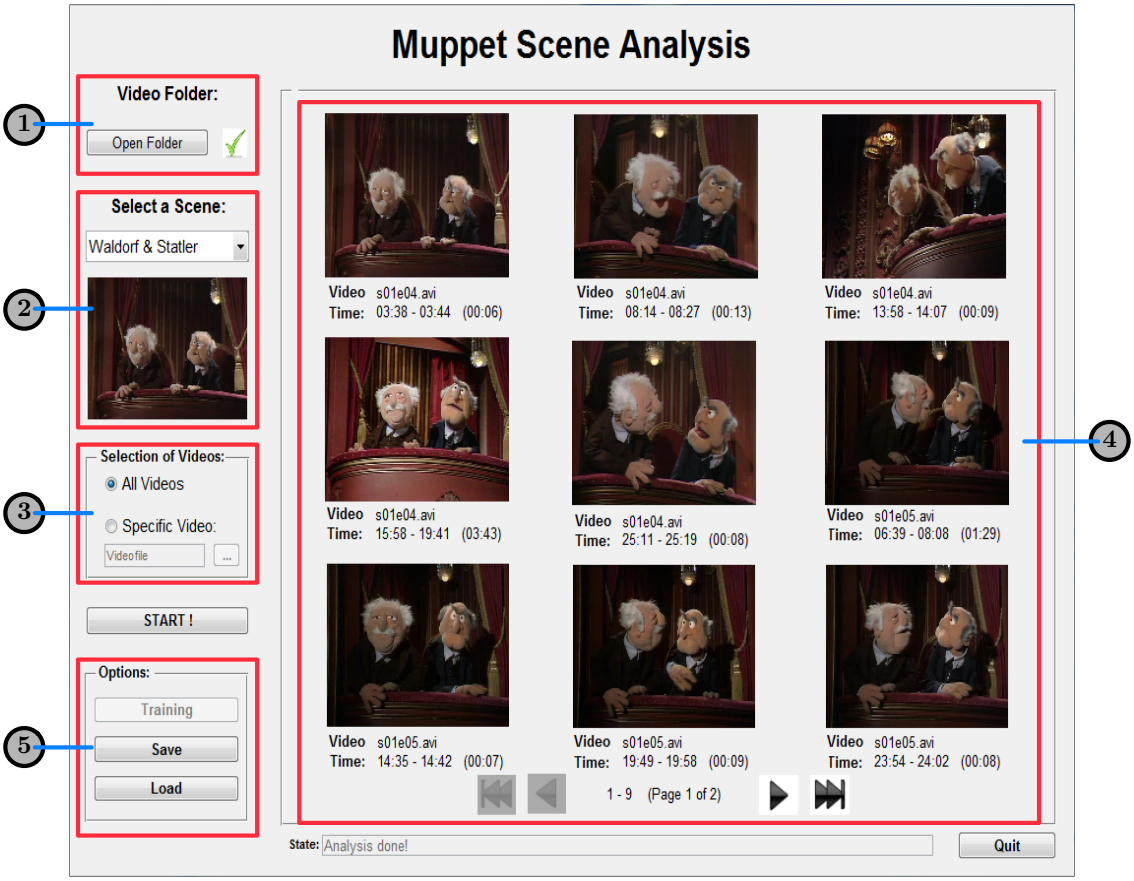


Abbildung 5.1: Graphische Benutzeroberfläche des Prototyps.

Sobald die Analysen und Berechnungen abgeschlossen sind, die jedoch aufgrund der großen Datenmengen einige Zeit in Anspruch nehmen können, werden die Ergebnisse durch Vorschaubilder visualisiert ④. Im Durchschnitt erfolgt die Analyse in *33* Minuten bei Videos mit Dauer

von ca. *25* Minuten. An dieser Stelle wird erwähnt, dass die segmentierten Szenen nicht separat abgespeichert werden. Nur Anfangszeitpunkt, Endzeitpunkt, Dauer und Kategorie der Szenen sind relevant. Außerdem unterliegen die visualisierten Klassifikationsresultate keiner speziellen Ordnung nach Bedeutsamkeit, sondern sie werden nach Videoname und Anfangszeitpunkt sortiert.

Zusätzlich steht noch eine weitere Funktion zur Verfügung, die das Abspielen von gewünschten Szenen erlaubt. In Abbildung 5.2 ist die graphische Benutzeroberfläche des Abspielmodus dargestellt. Abhängig von der gewählten Szene wird in diesem Modus nur der entsprechende Abschnitt des Videos wiedergegeben. Weitere optionale Funktionen sind das Speichern und Laden von Klassifikationen, sowie das Starten des Trainingsprozesses ⑤. Dabei werden Videos mit entsprechenden GT-Daten herangezogen und die Trainingsphase eingeleitet, die bereits in Abschnitt 4.3 erklärt worden ist.

Der restliche Teil des Kapitels befasst sich mit der Evaluierung der entwickelten Videoklassifikation. Welche Qualität die Klassifikationsresultate aufweisen, wird durch die Auswertung einzelner Teilschritte des Prototyps, als auch durch die Evaluierung des gesamten Systems aufgezeigt. In diesem Prozess kommen die verschiedenen Evaluierungsmethoden zum Einsatz, die bereits in Abschnitt 2.7 erklärt worden sind. Dabei werden insgesamt fünf Videos aus der ersten Staffel der Muppet Show und *15* verschiedene Szenenkategorien betrachtet. In Abbildung 5.3 ist ein Überblick aller Kategorien zu sehen. Im folgenden Abschnitt werden zunächst die Resultate der zeitlichen Segmentierung ausgewertet.

Abbildung 5.2: Abspielen einer segmentierten und klassifizierten Szene.

Abbildung 5.3: Überblick der Szenenkategorien des Prototyps.

5.2 Zeitliche Segmentierung

Für die Auswertung der zeitlichen Segmentierung werden Daten aus der Ground Truth herangezogen, die Informationen über Shot-Übergänge innerhalb der Videos enthalten. Neben dem Zeitpunkt des Auftritts ist die Dauer von Übergängen entscheidend. Es ist für unsere Anwendung aber nicht notwendig zu bestimmen, um welche Übergangsform (Schnitt, Ein-, Aus- oder Überblendung) es sich dabei handelt.

Wie bereits in Abschnitt 4.4.1 erwähnt, sind zwei verschiedene Methoden zur zeitlichen Segmentierung implementiert worden. Für die Evaluierung werden für alle Videos Recall, Precision und der F1-Score basierend auf GT-Daten ermittelt. Dabei beschreibt der Recall wie viele der Shot-Übergänge tatsächlich erkannt worden sind, während die Precision die Relevanz der erkannten Übergänge repräsentiert. In Tabelle 5.1 werden zunächst die Resultate präsentiert, die unter Verwendung von Histogram Intersection erreicht worden sind. Es wird ein durchschnittlicher F1-Score von *87,54%* erzielt, wobei in allen Fällen eine höhere Precision (durchschnittlich *91,25%*) als Recall (durchschnittlich *83,71%*) erreicht wird. Dieser Umstand lässt sich durch die Problematik bei der Detektion von Ein-, Aus- und Überblendungen erklären. Wie in Abschnitt 2.2.2 erläutert, werden bei Twin Comparison zwei Schwellwerte (T_h und T_l) benötigt, um aus Differenzwerten benachbarter Frames Shot-Übergänge zu detektieren. Basierend auf der Annahme, dass fortlaufende Übergänge zwar geringere Unterschiede als Schnitte aber größere Differenzwerte als Frames innerhalb eine Shots aufweisen, ist vor allem T_l für die Detektion von Ein-, Aus- und Überblendungen entscheidend. Jedoch wird das Festlegen des Schwellwertes er-

Zeitliche Segmentierung (Histogram Intersection)

	Recall	**Precision**	**F1-Score**
Video 1	85,86%	94,44%	89,95%
Video 2	85,71%	89,48%	87,55%
Video 3	87,20%	90,83%	88,98%
Video 4	73,87%	94,25%	82,83%
Video 5	85,92%	91,04%	88,41%
Gesamt	**83,71%**	**91,25%**	**87,54%**

Tabelle 5.1: Evaluierung der zeitlichen Segmentierung bei Verwendung von Histogram Intersection.

schwert, da diese Annahme im Falle des vorliegenden Videomaterials nicht immer zutrifft. Die Differenzwerte von Frames innerhalb eines Shots und von fortlaufenden Übergängen weisen nur geringe Unterschiede auf, wodurch diese Übergänge häufig nicht erkannt werden. Besonders in *Video 4* wirkt sich dieses Problem auf die Detektionsrate aus, da hier mehr solcher Übergänge vorkommen, als in den übrigen Videos.

Ein weiterer Grund für die fehlerhafte Detektion liegt in der Verwendung von Farbhistogrammen und den damit verbundenen Quantisierungsfehlern vor. Geringfügige Beleuchtungsänderungen wirken sich auf die Intensitätswerte der Pixel in den einzelnen Farbkanäle aus, wodurch Pixel anderen Bins zugewiesen werden als im Frame zuvor. Daraus resultiert eine große Differenz der Farbhistogramme und es wird somit ein Schnitt erkannt. Solche Beleuchtungsänderungen treten zum Beispiel während einer Ein- oder Ausblendung auf. Wie bereits zuvor erläutert, wird die Erkennung der Art des Überganges vom Prototypen nicht gefordert, weshalb es nicht relevant ist, ob anstelle eines fortlaufenden Überganges ein Schnitt erkannt wird. Jedoch können solche Beleuchtungsänderungen auch andere Gründe haben, wie beispielsweise Abschattung verursacht durch schnelle Objektbewegungen. In diesem Fall kommt es wieder zu Änderungen der Intensitätswerte und der damit verbundenen fälschlichen Detektion eines Shot-Überganges. Um den Einfluss von bewegenden Objekten auf die Resultate der zeitlichen Segmentierung zu minimieren, werden Frames in mehrere Zellen aufgeteilt und die χ^2-Distanz zur Differenzberechnung eingesetzt, wie bereits in Abschnitt 4.4.1 präsentiert. In Tabelle 5.2 sind die Resultate dieser alternativen Methode zur Detektion von Shot-Übergängen zu sehen. Im Vergleich zur ersten Methode wird ein ähnlicher durchschnittlicher F1-Score von *87,62%* erreicht. Dabei wird eine höhere Precision erzielt, jedoch auf Kosten des Recalls. Aufgrund der Aufteilung der Frames in Zellen und der besonderen Differenzbildung, wo die acht größten Differenzwerte verworfen werden, wird der Einfluss von lokalen Beleuchtungsänderungen minimiert. Jedoch wird gleichzeitig die Unterscheidbarkeit basierend auf Differenzwerten zwischen Frames erschwert, die Teil von Shots oder fortlaufenden Übergängen sind. Dadurch lassen sich die niedrigen Recall-Werte erklären. Basierend auf diesen Erkenntnissen und dem höheren

Aufwand bei der Berechnung der χ^2-Distanz, erfolgen die weiteren Evaluierungen unter Verwendung der Histogram Intersection.

Zeitliche Segmentierung (χ^2-Distanz)			
	Recall	**Precision**	**F1-Score**
Video 1	81,82%	95,29%	88,04%
Video 2	72,86%	100%	84,30%
Video 3	80,80%	97,12%	88,21%
Video 4	79,28%	96,70%	87,13%
Video 5	86,62%	94,62%	90,44%
Gesamt	**80,28%**	**96,75%**	**87,62%**

Tabelle 5.2: Evaluierung der zeitlichen Segmentierung bei Verwendung der χ^2-Distanz.

5.3 Evaluierung der Klassifikation von Musik, Sprache und Umgebungsgeräuschen

Als nächstes wird die Qualität der Resultate der Klassifikation von Musik, Sprache und Umgebungsgeräuschen aufgezeigt. Für die Auswertung wird Cross Validation eingesetzt, wobei GT-Daten in eine Trainings- und Testmenge unterteilt werden (siehe Abschnitt 2.7). Um eine aussagekräftige Evaluierung zu erreichen, wird die sogenannte zehnfache Kreuzvalidierung eingesetzt. Dabei werden die Daten der einzelnen Kategorien in zehn möglichst gleichgroße Teilmengen gegliedert. Anschließend werden zehn Testläufe durchgeführt, wobei jeweils eine Teilmenge als Testmenge betrachtet wird und die restlichen Daten für das Training herangezogen werden. Die Gesamtresultate der Evaluierung lassen sich als Durchschnitt der Ergebnisse aus den einzelnen Durchläufen erklären. In Tabelle 5.3 werden die Resultate der Evaluierung zusammengefasst.

Wie zu sehen ist, werden bei der Klassifikation von Sprache und Musik ähnliche Resultate erzielt, während bei Umgebungsgeräuschen mehr Fehlklassifikationen auftreten. Jedoch hat dies nur einen geringen Einfluss auf die durchschnittliche Gesamtklassifikationsrate von *95,66%*, da es sich dabei um einen gewichteten Durchschnittswert des F1-Scores handelt und innerhalb von Videos weniger Umgebungsgeräusche vorkommen als Musik oder Sprache. Der Grund für die Schwierigkeiten bei der auditiven Klassifikation sind auftretende Überlappungen der drei Audiotypen. Vor allem Umgebungsgeräusche, wie beispielsweise das Klatschen oder Lachen des Publikums, sind häufig in Verbindung mit Musik oder Sprache zu hören.

Auditive Klassifikation			
	Recall	**Precision**	**F1-Score**
Musik	95,73%	97,11%	96,42%
Sprache	98,66%	96,90%	97,77%
Umgebung	80,84%	83,33%	82,07%
Gewichteter Durchschnitt	**95,69%**	**95,65%**	**95,66%**

Tabelle 5.3: Evaluierung der Klassifikation von Musik, Sprache und Umgebungsgeräuschen.

5.4 Klassifikation einzelner Shots

In diesem Abschnitt werden die erzielten Resultate der Klassifikation von einzelnen Shots dargestellt, um die Qualität der gewählten audiovisuellen Merkmale und Klassifikationsmodelle zu überprüfen. Es ist zu beachten, dass bei diesem Evaluierungsprozess keine zeitliche Segmentierung vorgenommen wird, sondern notwendige Informationen über Anfangs- und Endzeitpunkt von Shots werden den GT-Daten entnommen. Ausgehend von einzelnen Shots gilt es zunächst audiovisuelle Features zu extrahieren. Anschließend werden diese Merkmale für den eigentlichen Evaluierungsprozess herangezogen, wobei LOO-CV (siehe Abschnitt 2.7) zum Einsatz kommt. Dabei werden wiederholt vier der fünf Videos für das Training der Klassifikationsmodelle herangezogen, während das übrige Video als Testvideo dient. Die Klassifikationsresultate werden abschließend für jedes Video mit Hilfe der Genauigkeitsrate Acc ausgewertet, die folgendermaßen definiert ist [96]:

$$Acc = \frac{TP + TN}{FP + FN + TP + TN} \tag{5.1}$$

wobei TP, TN, FP und FN den Evaluierungsbegriffen entsprechen, welche bereits in Abschnitt 2.7 vorgestellt wurden. In unserem Anwendungsfall gibt es jedoch keine *True Negatives* und dementsprechend ist dieser Wert immer null. In Tabelle 5.4 werden die Ergebnisse der Evaluierung präsentiert, wobei der Vergleich der Genauigkeitsraten basierend auf visueller und audiovisueller Merkmale dargestellt wird. Die nur minimalen Unterschiede der Resultate lassen sich durch weniger diskriminative auditive Informationen erklären. Im Durchschnitt wird bei Verwendung audiovisueller Merkmale eine Genauigkeitsrate von *87,60%* erreicht und es ist ersichtlich, dass vor allem die ersten drei Videos ähnliche Werte aufweisen. Währenddessen werden in Video *4* und Video *5* deutlich geringere Genauigkeitswerte erreicht, wofür hauptsächlich Shots mit selten vorkommenden Szenenkategorien verantwortlich sind. Aufgrund der Seltenheit sind nur wenige Daten vorhanden, um die Klassifikationsmodelle zu trainieren. Außerdem weisen speziell diese Szenen variationsreiche Bühnenbilder und unterschiedliche auftretende Muppet-Figuren auf, wodurch eine Klassifikation zusätzlich erschwert wird.

Im Folgenden werden die Klassifikationsresultate von drei bedeutenden Szenenkategorien präsentiert, welche bereits in Abschnitt 3.3 vorgestellt worden sind. Diese Kategorien sind vor

Shot-Klassifikation	**nur visuelle Merkmale**	**visuelle & auditive Merkmale**
	Genauigkeitsrate	**Genauigkeitsrate**
Video 1	90,22%	90,22%
Video 2	94,37%	94,37%
Video 3	92,56%	93,24%
Video 4	76,15%	77,06%
Video 5	81,76%	83,11%
Gesamt	**87,01%**	**87,60%**

Tabelle 5.4: Evaluierung der Klassifikation von Shots.

allem aufgrund des wiederholten Vorkommens und der Erfüllung aller Anforderungen an Szenenkategorien für die Evaluierung des Prototyps geeignet. Dabei wird wie zuvor bei der Auswertung der Shot-Klassifikation LOO-CV eingesetzt. In den Tabellen 5.5, 5.6 und 5.7 werden die Resultate der Kategorien *Kermit's Ansage*, *Waldorf & Statler* und *Tanzszene* präsentiert.

Die Klassifikationsresultate zeigen, dass sowohl die audiovisuellen Merkmale als auch die Klassifikationsmodelle in der Lage sind, den semantischen Inhalt von einzelnen Shots zu erfassen und eine aussagekräftige Beschreibung zu erzeugen. Trotzdem treten Fehlklassifikationen auf, wobei das Ausmaß abhängig vom Video und den vorkommenden Szenenkategorien ist. Vor allem wenn zwei Kategorien visuell ähnlich sind, können fehlerhafte Resultate auftreten, da den viuellen Modellen mehr Vertrauen zugesprochen wird als der auditiven Klassifikation. Wie bereits in Abschnitt 4.4.3 erklärt, sind in unserem Anwendungsfall auditive Informationen weniger diskriminativ als visuelle Beschreibungen, weshalb die Gewichtung entscheidend für die Qualität der Resultate ist.

Kermit's Ansage	**(insgesamt 28 Shots)**		
	Recall	**Precision**	**F1-Score**
Video 1	100%	100%	100%
Video 2	75,00%	100%	85,71%
Video 3	100%	100%	100%
Video 4	100%	80,00%	88,89%
Video 5	100%	90,00%	94,74%

Tabelle 5.5: Auswertung der Szenenkategorie *Kermit's Ansage*.

Waldorf & Statler	**(insgesamt 50 Shots)**		
	Recall	**Precision**	**F1-Score**
Video 1	93,75%	100%	96,78%
Video 2	100%	100%	100%
Video 3	100%	100%	100%
Video 4	80,00%	100%	88,89%
Video 5	100%	100%	100%

Tabelle 5.6: Auswertung der Szenenkategorie *Waldorf & Statler*.

Tanzszene	**(insgesamt 31 Shots)**		
	Recall	**Precision**	**F1-Score**
Video 1	100%	100%	100%
Video 2	100%	100%	100%
Video 3	89,60%	100%	94,52%
Video 4	100%	100%	100%
Video 5	100%	100%	100%

Tabelle 5.7: Auswertung der Szenenkategorie *Tanzszene*.

5.5 Evaluierung segmentierter Szenen

Zuletzt werden die Resultate der segmentierten und klassifizierten Szenen ausgewertet, wo auch die bereits evaluierten Teilschritte einfließen (Abschnitte 5.2 bis 5.4). Für die Evaluierung ganzer Szenen werden zunächst wieder GT-Daten herangezogen, um für jeden Zeitpunkt der Videos die tatsächlich auftretenden Kategorien zu entnehmen. Anschließend wird mit Hilfe von LOO-CV die Qualität der Klassifikationsresultate bewertet, wobei ausgehend von Videos die gesamte Klassifikationsphase des Prototyps durchlaufen wird.

In Tabelle 5.8 werden die Ergebnisse der Evaluierung in Form von Genauigkeitsraten für alle fünf Videos präsentiert, wobei nicht immer die zeitliche Segmentierung eingesetzt wird. Die Resultate der Szenenklassifikation ohne zeitlicher Segmentierung werden erst am Ende des Abschnittes näher erläutert. Zunächst werden die Resultate der Klassifikation unter Verwendung der zeitlicher Segmentierung betrachtet. Dabei ist zu erkennen, dass die einzelnen Videos

Szenenklassifikation	mit zeit. Seg.	ohne zeit. Seg.
	Genauigkeitsrate	**Genauigkeitsrate**
Video 1	56,82%	79,07%
Video 2	70,46%	89,19%
Video 3	67,39%	84,44%
Video 4	59,53%	71,43%
Video 5	55,32%	68,18%
Gesamt	**61,90%**	**78,46%**

Tabelle 5.8: Evaluierung der Klassifikation von Szenen.

schwankende Klassifikationsraten aufweisen. Der Grund dafür lässt sich abhängig vom Inhalt der Videos durch folgende Fehlerfortpflanzung erklären:

(1) **Fehlerhafte zeitliche Segmentierung:**
Ausgehend von Testvideos wird zunächst eine zeitliche Segmentierung durchgeführt. Wie bereits aus Abschnitt 5.2 ersichtlich, können vor allem bei fortlaufenden Übergängen fehlerhafte Resultate auftreten. Außerdem stellen besondere Übergänge ein weiteres Problem dar, wie beispielsweise das Öffnen des Bühnenvorhangs, wodurch zwei unterschiedliche Szenenkategorien zu einer verschmelzen.

(2) **Fehlerhafte Klassifikation einzelner Shots:**
Abhängig von den Ergebnissen der zeitlichen Segmentierung werden aus den einzelnen Shots audiovisuelle Merkmale extrahiert, die anschließend mit den entsprechenden Klassifikationsmodellen klassifiziert werden. Neben möglichen Fehklassifikationen von Shots bei exakten Ergebnissen der zeitlichen Segmentierung, führen fehlerhafte Shot-Grenzen zu ungenauen Beschreibungen der Inhalte durch audiovisuelle Merkmale. Dadurch lassen sich weitere Fehlklassifikationen erklären.

(3) **Fehlerhafte Bildung von Szenen:**
Basierend auf den Ergebnissen der Shot-Klassifikation werden mehrere Shots zu Szenen zusammengefasst, abhängig von der Szenenkategorie und dem zeitlichem Auftreten (siehe Abschnitt 4.4.4). Unterliegen einzelne Shots einer fehlerhaften Klassifikation, so hat dies auch Auswirkungen auf das Bilden von Szenen.

Die schwankenden Ergebnisse der Videoklassifikation lassen sich abhängig vom Video durch diese drei Punkte erklären. Vor allem die zeitliche Segmentierung hat großen Einfluss auf die Klassifikationsrate des Prototyps. Aus diesem Grund werden in Tabelle 5.8 zusätzlich die Genauigkeitswerte des Klassifikationsprozesses aller Videos präsentiert, wo die benötigten Informationen über Begin und Ende von Shots aus den GT-Daten entnommen werden. Der deutliche

Anstieg der durchschnittlichen Genauigkeitsrate von *61,9%* auf *78,46%* unterstreicht sowohl die Problematik der zeitlichen Segmentierung, als auch die Performance des Prototyps. Zusammenfassend zeigt die Evaluierung, dass sowohl die gewählten audiovisuellen Merkmale als auch die Klassifikationsmodelle in der Lage sind, einzelne Szenen aus der Muppet Show zu segmentieren und weitgehend korrekt zu klassifizieren. Jedoch ist die Performance des Prototyps neben den Ergebnissen der zeitlichen Segmentierung abhängig vom Variationsreichtum einzelner Szenenkategorien und der entsprechend benötigten Menge an Trainingsdaten.

KAPITEL 6

Schlussfolgerung & Ausblick

Die Ergebnisse der Evaluierung des entwickelten Prototyps zeigen, dass aussagekräftige audiovisuelle Merkmale und geeignete Klassifikationsmodelle in der Lage sind, die semantische Bedeutung von Videoszenen mehrheitlich korrekt zu extrahieren und zu beschreiben. Basierend auf der Analyse des Videomaterials können charakteristische Eigenschaften festgestellt werden, die sowohl bei der Definition von Szenenkategorien unterstützen, als auch bei der Wahl von Merkmalen hilfreich sein können. Speziell bei Videos aus der Muppet Show, wo sich die verschiedenen Kategorien hauptsächlich durch individuelle Bühnenbilder und unterschiedlich auftretenden Muppet-Figuren erklären lassen, sind sowohl visuelle Merkmale, wie Farbmomente, Color Coherence Vector, Color Auto-Correlogram und SIFT-Features, als auch die Segmentierung von Audiosignalen in Musik, Sprache, Stille und Umgebungsgeräusche für die erfolgreiche Klassifikation von Szenen geeignet. Jedoch ist die Qualität der Klassifikationsresultate abhängig vom Variationsreichtum einzelner Szenenkategorien und der Menge an vorhandenen Trainingsdaten.

Um die Videoklassifikation zu verbessern, können in zukünftigen Arbeiten weitere audiovisuelle Merkmale in Betracht gezogen werden, welche einer höheren semantischen Bedeutung unterliegen. Beispielsweise bieten sowohl die Detektion und Erkennung von Gesichtern, als auch das Identifizieren von Stimmen weitere Möglichkeiten den Inhalt von Videosequenzen zu ermitteln. Speziell bei der Klassifikation von Videos aus der Muppet Show könnten dadurch das Auftreten von spezifischen Charakteren detektiert und entsprechende Szenen segmentiert werden, jedoch ist dieses Vorhaben mit enormen Aufwand während der Trainingsphase verbunden.

Außerdem können die charakteristischen Besonderheiten des Videomaterials besser ausgenutzt werden, um aussagekräftige Beschreibungen des Inhalts zu erlangen. Vor allem das annähernd gleichbleibende Bühnenbild als auch das starre Verhalten der Muppet-Figuren können hilfreich sein, um beispielsweise mit Hilfe von *Template Matching* einzelne Objekte zu detektieren. Es existiert bereits eine Vielzahl an unterschiedlichen Methoden, wo ausgehend von Musterbildern, sogenannte Templates, Objekte in Bildern oder Videos detektiert und lokalisiert werden [97, 98, 99]. Dabei können Informationen aus der Analyse des Videomaterials herangezogen werden, um Bildbereiche einzugrenzen, wo sich das gesuchte Objekt möglicherweise befinden kann. Beispielsweise sind Muppet-Figuren kaum im oberen Drittel des Bildes zu sehen. Es ist

jedoch zu beachten, dass die Resultate von Template Matching abhängig von den gewählten Templates sind. Außerdem müssen sowohl unterschiedliche Skalierungen und Blickwinkeln als auch das Auftreten von Verdeckungen und Veränderungen der Lichtverhältnisse berücksichtigt werden, um eine erfolgreiche Detektion und Lokalisierung von Objekten zu ermöglichen. Aus diesem Grund ist es notwendig mehrere unterschiedliche oder verformbare Templates zu verwenden.

Eine weitere Möglichkeit zur Steigerung der Performance des Prototyps bietet neben der Verbesserung der zeitlichen Segmentierung, die individuelle Gewichtung der einzelnen Klassifikationsmodelle. Abhängig von den Szenenkategorien kann eine erneute Analyse des Videomaterials Aufschluss darüber geben, unter welchen Umständen eine Adaptierung der Gewichtungen sinnvoll ist, um bessere Klassifikationsresultate zu erreichen. Durch die Umsetzung der vorgestellten Verbesserungsvorschläge kann die Performance der Videoklassifikation gesteigert werden, wobei es als realistisch angenommen wird, *85%* bis *90%* der Szenen korrekt klassifizieren zu können.

Abbildungsverzeichnis

Tabellenverzeichnis

Quellen- und Literaturverzeichnis

[1] SEBE, Nicu ; LEW, Michael S. ; ZHOU, Xiang ; HUANG, Thomas S. ; BAKKER, Erwin M.: The state of the art in image and video retrieval. In: *Image and Video Retrieval*. Springer, 2003, S. 1–8

[2] *Statista - YouTube*. http://de.statista.com/statistik/daten/studie/207321/umfrage/upload-von-videomaterial-bei-youtube-pro-minute-zeitreihe/, Abruf: 23. April 2013

[3] BORTH, Damian ; ULGES, Adrian ; SCHULZE, Christian ; BREUEL, Thomas M.: Keyframe extraction for video tagging and summarization. In: *Proc. Informatiktage*, 2008, S. 45–48

[4] *YouTube - The Big Bang Theory*. http://www.youtube.com/watch?v=riLDqY42RWo, Abruf: 24. April 2013

[5] PATEL, BV ; MESHRAM, BB: Content based video retrieval systems. In: *arXiv preprint arXiv:1205.1641* (2012)

[6] MERLINO, Andrew ; MOREY, Daryl ; MAYBURY, Mark: Broadcast news navigation using story segmentation. In: *Proceedings of the Fifth ACM International Conference on Multimedia* ACM, 1997, S. 381–391

[7] SIDIROPOULOS, Panagiotis ; MEZARIS, Vasileios ; KOMPATSIARIS, Ioannis ; MEINEDO, Hugo ; BUGALHO, Miguel ; TRANCOSO, Isabel: Video scene segmentation system using audio visual features. In: *Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS*, 2010

[8] WANG, Hualu ; DIVAKARAN, Ajay ; VETRO, Anthony ; CHANG, Shih-Fu ; SUN, Huifang: Survey of compressed-domain features used in audio-visual indexing and analysis. In: *Journal of Visual Communication and Image Representation* 14 (2003), Nr. 2, S. 150–183

[9] NGAN, King N.: *Video segmentation and its applications*. Springer Science+ Business Media, 2011

[10] *Muppet Wikia*. http://muppet.wikia.com/wiki/Category:The_Muppets_Characters, Abruf: 24. April 2013

[11] IDRIS, F ; PANCHANATHAN, S: Review of image and video indexing techniques. In: *Journal of Visual Communication and Image Representation* 8 (1997), Nr. 2, S. 146–166

[12] ZHANG, Hong J. ; WU, Jianhua ; ZHONG, Di ; SMOLIAR, Stephen W.: An integrated system for content-based video retrieval and browsing. In: *Pattern Recognition* 30 (1997), Nr. 4, S. 643–658

[13] BRUNELLI, Roberto ; MICH, Ornella ; MODENA, Carla-Maria: A Survey on the Automatic Indexing of Video Data. In: *Journal of Visual Communication and Image Representation* 10 (1999), Nr. 2, S. 78–112

[14] EIDENBERGER, Horst: *Professional Media Understanding: The Common Methods of Audio Retrieval, Biosignal Processing, Content-Based Image Retrieval, Face Recognition, Music Classification, Speech Recognition, Text Retrieval and Video Surveillance*. BoD–Books on Demand, 2012

[15] VERMAAK, Jaco ; PEREZ, Patrick ; GANGNET, Michel ; BLAKE, Andrew: Rapid summarization and browsing of video sequences. In: *British Machine Vision Conference, BMVC* Bd. 1, 2002

[16] ULGES, Adrian ; SCHULZE, Christian ; KEYSERS, Daniel ; BREUEL, Thomas M.: Content-based video tagging for online video portals. In: *MUSCLE/Image-CLEF Workshop*, 2007

[17] JIANG, Yu-Gang ; NGO, Chong-Wah ; YANG, Jun: Towards optimal bag-of-features for object categorization and semantic video retrieval. In: *Proceedings of the 6th ACM International Conference on Image and Video Retrieval* ACM, 2007, S. 494–501

[18] SATO, Toshio ; KANADE, Takeo ; HUGHES, Ellen K. ; SMITH, Michael A. ; SATOH, Shin'ichi: Video OCR: indexing digital news libraries by recognition of superimposed captions. In: *Multimedia Systems* 7 (1999), Nr. 5, S. 385–395

[19] LIENHART, Rainer ; EFFELSBERG, Wolfgang: Automatic text segmentation and text recognition for video indexing. In: *Multimedia systems* 8 (2000), Nr. 1, S. 69–81

[20] MINAMI, Kenichi ; AKUTSU, Akihito ; HAMADA, Hiroshi ; TONOMURA, Yoshinobu: Video handling with music and speech detection. In: *MultiMedia, IEEE* 5 (1998), Nr. 3, S. 17–25

[21] LIU, Zhu ; WANG, Yao ; CHEN, Tsuhan: Audio feature extraction and analysis for scene segmentation and classification. In: *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* 20 (1998), Nr. 1-2, S. 61–79

[22] LIN, Tong ; ZHANG, Hong-Jiang: Automatic video scene extraction by shot grouping. In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on* Bd. 4 IEEE, 2000, S. 39–42

[23] PORTER, Sarah ; MIRMEHDI, Majid ; THOMAS, Barry: Temporal video segmentation and classification of edit effects. In: *Image and Vision Computing* 21 (2003), Nr. 13, S. 1097–1106

[24] SAKARYA, Ufuk ; TELATAR, Ziya: Graph-based multilevel temporal video segmentation. In: *Multimedia Systems* 14 (2008), Nr. 5, S. 277–290

[25] LIENHART, Rainer W.: Comparison of automatic shot boundary detection algorithms. In: *Electronic Imaging'99* International Society for Optics and Photonics, 1998, S. 290–301

[26] BORECZKY, John S. ; ROWE, Lawrence A.: Comparison of video shot boundary detection techniques. In: *Journal of Electronic Imaging* 5 (1996), Nr. 2, S. 122–128

[27] *Elements of Cinema*. `http://www.elementsofcinema.com/editing/types-of-transition.html`, Abruf: 30. April 2013

[28] *Media College*. `http://www.mediacollege.com/video/editing/transition/types.html`, Abruf: 30. April 2013

[29] PORTER, SV ; MIRMEHDI, M ; THOMAS, BT: Video cut detection using frequency domain correlation. In: *Pattern Recognition, 2000. Proceedings. 15th International Conference on* Bd. 3 IEEE, 2000, S. 409–412

[30] KÜÇÜKTUNÇ, Onur ; GÜDÜKBAY, Uğur ; ULUSOY, Özgür: Fuzzy color histogram-based video segmentation. In: *Computer Vision and Image Understanding* 114 (2010), Nr. 1, S. 125–134

[31] PORTER, Sarah ; MIRMEHDI, Majid ; THOMAS, Barry: Detection and classification of shot transitions. In: *British Machine Vision Conference (BMVC)* IEEE Computer Society Washington, DC, 2001, S. 73–82

[32] GARGI, Ullas ; KASTURI, Rangachar ; STRAYER, Susan H.: Performance characterization of video-shot-change detection methods. In: *Circuits and Systems for Video Technology, IEEE Transactions on* 10 (2000), Nr. 1, S. 1–13

[33] KOPRINSKA, Irena ; CARRATO, Sergio: Temporal video segmentation: A survey. In: *Signal Processing: Image Communication* 16 (2001), Nr. 5, S. 477–500

[34] ZHANG, HongJiang ; KANKANHALLI, Atreyi ; SMOLIAR, Stephen W.: Automatic partitioning of full-motion video. In: *Multimedia Systems* 1 (1993), Nr. 1, S. 10–28

[35] ZHUANG, Yueting ; RUI, Yong ; HUANG, Thomas S. ; MEHROTRA, Sharad: Adaptive key frame extraction using unsupervised clustering. In: *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on* Bd. 1 IEEE, 1998, S. 866–870

[36] WOLF, Wayne: Key frame selection by motion analysis. In: *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on* Bd. 2 IEEE, 1996, S. 1228–1231

[37] PORTER, Sarah ; MIRMEHDI, Majid ; THOMAS, Barry: Video indexing using motion estimation. In: *The British Machine Vision Conference*, 2003

[38] RASHEED, Zeeshan ; SHAH, Mubarak: Detection and representation of scenes in videos. In: *Multimedia, IEEE Transactions on* 7 (2005), Nr. 6, S. 1097–1105

[39] DUFAUX, Frederic: *Keyframe selection to represent a video*. März 23 2004. – US Patent 6,711,587

[40] AHUJA, Ravindra K. ; MEHLHORN, Kurt ; ORLIN, James ; TARJAN, Robert E.: Faster algorithms for the shortest path problem. In: *Journal of the ACM (JACM)* 37 (1990), Nr. 2, S. 213–223

[41] DIJKSTRA, Edsger W.: A note on two problems in connexion with graphs. In: *Numerische Mathematik* 1 (1959), Nr. 1, S. 269–271

[42] CHORAS, Ryszard S.: Image feature extraction techniques and their applications for CBIR and biometrics systems. In: *International Journal of Biology and Biomedical Engineering* 1 (2007), Nr. 1, S. 6–16

[43] UMBAUGH, Scott E. ; WEI, Y-S ; ZUKE, Mark: Feature extraction in image analysis. A program for facilitating data reduction in medical image classification. In: *Engineering in Medicine and Biology Magazine, IEEE* 16 (1997), Nr. 4, S. 62–73

[44] EIDENBERGER, Horst: *Fundamental Media Understanding*. BoD–Books on Demand, 2011

[45] SONKA, Milan ; HLAVAC, Vaclav ; BOYLE, Roger u. a.: Image processing, analysis, and machine vision. (1999)

[46] NIXON, Mark ; AGUADO, Alberto S.: *Feature extraction & image processing*. Academic Press, 2008

[47] HAN, Ju ; MA, Kai-Kuang: Fuzzy color histogram and its use in color image retrieval. In: *Image Processing, IEEE Transactions on* 11 (2002), Nr. 8, S. 944–952

[48] STRICKER, Markus A. ; ORENGO, Markus: Similarity of color images. In: *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology* International Society for Optics and Photonics, 1995, S. 381–392

[49] RAO, Aibing ; SRIHARI, Rohini K. ; ZHANG, Zhongfei: Spatial color histograms for content-based image retrieval. In: *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on* IEEE, 1999, S. 183–186

[50] PASS, Greg ; ZABIH, Ramin: Histogram refinement for content-based image retrieval. In: *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on* IEEE, 1996, S. 96–102

[51] PASS, Greg ; ZABIH, Ramin ; MILLER, Justin: Comparing images using color coherence vectors. In: *Proceedings of the Fourth ACM International Conference on Multimedia* ACM, 1997, S. 65–73

[52] MA, Wei-Ying ; ZHANG, Hong J.: Benchmarking of image features for content-based retrieval. In: *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on* Bd. 1 IEEE, 1998, S. 253–257

[53] HUANG, Jing ; KUMAR, S R. ; MITRA, Mandar ; ZHU, Wei-Jing ; ZABIH, Ramin: Spatial color indexing and applications. In: *International Journal of Computer Vision* 35 (1999), Nr. 3, S. 245–268

[54] HUANG, Jing ; KUMAR, S R. ; MITRA, Mandar ; ZHU, Wei-Jing ; ZABIH, Ramin: Image indexing using color correlograms. In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on* IEEE, 1997, S. 762–768

[55] SIVIC, Josef ; ZISSERMAN, Andrew: Video Google: A text retrieval approach to object matching in videos. In: *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* IEEE, 2003, S. 1470–1477

[56] WANG, JINQIAO ; LU, HANQING ; DUAN, LINGYU ; JIN, JESSE S.: Commercial Video Retrieval with Video-based Bag of Words. In: *Fifth International Conference on Intelligent Multimedia Computing and Networking*, 2007

[57] MIKOLAJCZYK, Krystian ; SCHMID, Cordelia: A performance evaluation of local descriptors. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27 (2005), Nr. 10, S. 1615–1630

[58] LOWE, David G.: Distinctive image features from scale-invariant keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, S. 91–110

[59] LOWE, David G.: Object recognition from local scale-invariant features. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on* Bd. 2 Ieee, 1999, S. 1150–1157

[60] SZELISKI, Richard: *Computer vision: algorithms and applications*. Springer, 2010

[61] SUNDARAM, Hari ; CHANG, Shih-Fu: Video scene segmentation using video and audio features. In: *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on* Bd. 2 IEEE, 2000, S. 1145–1148

[62] HASAN, Md R. ; JAMIL, Mustafa ; RAHMAN, Md Golam Rabbani Md S.: Speaker identification using Mel frequency cepstral coefficients. In: *Variations* 1 (2004), S. 4

[63] EZZAIDI, Hassan ; ROUAT, Jean ; O'SHAUGHNESSY, Douglas: Towards combining pitch and MFCC for speaker identification systems. In: *Proc. Seventh European Conf. on Speech Communication and Technology (Eurospeech 2001), Aalborg, Denmark* Citeseer, 2001, S. 2825–2828

[64] JIANG, Hao ; LIN, Tony ; ZHANG, Hongjiang: Video segmentation with the support of audio segmentation and classification. In: *Proc. IEEE ICME*, 2000

[65] ZHANG, Tong ; KUO, CC J.: *Content-based audio classification and retrieval for audio-visual data parsing*. Bd. 606. Kluwer Academic Pub, 2001

[66] PANAGIOTAKIS, Costas ; TZIRITAS, George: A speech/music discriminator based on RMS and zero-crossings. In: *Multimedia, IEEE Transactions on* 7 (2005), Nr. 1, S. 155–166

[67] EL-MALEH, Khaled ; KLEIN, Mark ; PETRUCCI, Grace ; KABAL, Peter: Speech/music discrimination for multimedia applications. In: *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on* Bd. 6 IEEE, 2000, S. 2445–2448

[68] MITROVIĆ, Dalibor ; ZEPPELZAUER, Matthias ; BREITENEDER, Christian: Features for content-based audio retrieval. In: *Advances in Computers* 78 (2010), S. 71–150

[69] ZHANG, Tong ; KUO, C-C J.: Content-based classification and retrieval of audio. In: *SPIE's International Symposium on Optical Science, Engineering, and Instrumentation* International Society for Optics and Photonics, 1998, S. 432–443

[70] KEDEM, Benjamin: Spectral analysis and discrimination by zero-crossings. In: *Proceedings of the IEEE* 74 (1986), Nr. 11, S. 1477–1493

[71] LOGAN, Beth u. a.: Mel frequency cepstral coefficients for music modeling. In: *International Symposium on Music Information Retrieval* Bd. 28, 2000, S. 5

[72] BISHOP, Christopher M.: *Neural networks for pattern recognition*. Oxford University Press, 1995

[73] WOOD, Jeffrey: Invariant pattern recognition: a review. In: *Pattern Recognition* 29 (1996), Nr. 1, S. 1–17

[74] HASTIE, Trevor ; TIBSHIRANI, Robert: Discriminant adaptive nearest neighbor classification. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18 (1996), Nr. 6, S. 607–616

[75] SILPA-ANAN, Chanop ; HARTLEY, Richard: Optimised KD-trees for fast image descriptor matching. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* IEEE, 2008, S. 1–8

[76] PHILBIN, James ; CHUM, Ondrej ; ISARD, Michael ; SIVIC, Josef ; ZISSERMAN, Andrew: Object retrieval with large vocabularies and fast spatial matching. In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* IEEE, 2007, S. 1–8

[77] BURGES, Christopher J.: A tutorial on support vector machines for pattern recognition. In: *Data Mining and Knowledge Discovery* 2 (1998), Nr. 2, S. 121–167

[78] IVANCIUC, Ovidiu: Applications of support vector machines in chemistry. In: *Reviews in Computational Chemistry* 23 (2007), S. 291

[79] XU, Yun ; ZOMER, Simeone ; BRERETON, Richard G.: Support vector machines: a recent method for classification in chemometrics. In: *Critical Reviews in Analytical Chemistry* 36 (2006), Nr. 3-4, S. 177–188

[80] LANDGREBE, Thomas C. ; PACLIK, Pavel ; DUIN, Robert P. ; BRADLEY, Andrew P.: Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In: *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* Bd. 4 IEEE, 2006, S. 123–127

[81] CRESTANI, Fabio ; LALMAS, Mounia ; VAN RIJSBERGEN, Cornelis J.: *Information Retrieval: Uncertainty and Logics: Advanced Models for the Representation and Retrieval of Information*. Bd. 4. Kluwer Academic Pub, 1998

[82] BRERETON, Richard G.: *Applied chemometrics for scientists*. Wiley, 2007

[83] LOW, Chien Y. ; TIAN, Qi ; ZHANG, Hongjiang: An automatic news video parsing, indexing and browsing system. In: *Proceedings of the Fourth ACM International Conference on Multimedia* ACM, 1997, S. 425–426

[84] HUA, Xian-Sheng ; LU, Lie ; ZHANG, Hong-Jiang: Robust learning-based TV commercial detection. In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* IEEE, 2005, S. 4–pp

[85] *Jim Henson's Fantatstic World - A Teacher's Guide; Michener Art Museum*. http://learn.michenerartmuseum.org/wp-content/uploads/2010/05/Henson-Curriculum-FINAL-WEB.pdf, Abruf: 28. Mai 2013

[86] STOESSNER, Jennifer K.: *Building American Puppetry on the Jim Henson Foundation Dissertation*, The Ohio State University, Diss., 2008

[87] *Jennifer Kate Stuller; Ink-Stained Amazons and Cinematic Warriors: Superwomen in Modern Mythology*. http://ink-stainedamazon.com/geek-monthly-jim-hensons-fantastic-world-a-retrospective/, Abruf: 27. Mai 2013

[88] BERGER, Jürgen ; DILLMANN, Claudia ; GEHR, Herbert ; FRANKFURT AM, Deutsches F.: *Muppets, Monster & Magie: die Welt von Jim Henson*. Deutsches Filmmuseum, 1987

[89] DIRCKS, Phyllis T.: *American Puppetry: Collections, History and Performance*. Bd. 23. McFarland, 2004

[90] *Goodreads - Jim Henson*. http://www.goodreads.com/author/show/4427.Jim_Henson, Abruf: 03. Juni 2013

[91] *The Jim Henson Project*. http://jimhensonproject.blogspot.co.at/, Abruf: 28. Mai 2013

[92] *Jim Henson's Red Book*. http://www.henson.com/jimsredbook/2011/08/04/8-1961/, Abruf: 28. Mai 2013

[93] LIU, Xiaowen ; OWEN, Charles B. ; MAKEDON, Fillia: Automatic video pause detection filter. (1997)

[94] HALL, Mark ; FRANK, Eibe ; HOLMES, Geoffrey ; PFAHRINGER, Bernhard ; REUTEMANN, Peter ; WITTEN, Ian H.: The WEKA data mining software: an update. In: *ACM SIGKDD Explorations Newsletter* 11 (2009), Nr. 1, S. 10–18

[95] NAGASAKA, Akio ; TANAKA, Yuzuru: Automatic video indexing and full-video search for object appearances. (1992)

[96] OLSON, David L. ; DELEN, Dursun: *Advanced data mining techniques*. Springer, 2008

[97] SCHWEITZER, Haim ; BELL, JW ; WU, Feng: Very fast template matching. In: *Computer Vision—ECCV 2002*. Springer, 2006, S. 358–372

[98] BRIECHLE, Kai ; HANEBECK, Uwe D.: Template matching using fast normalized cross correlation. In: *Aerospace/Defense Sensing, Simulation, and Controls* International Society for Optics and Photonics, 2001, S. 95–102

[99] NGUYEN, Hieu T. ; SMEULDERS, Arnold W. M.: Fast occluded object tracking by a robust appearance filter. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 26 (2004), Nr. 8, S. 1099–1104