Karlsruhe Institute of Technology

Comparative Analysis of Content-based TV Genre Classification and Web Video Categorization

Diploma thesis of

Tomas Semela

At the faculty of Computer Science
Institute for Anthropomatics

Advisor: Hazım K. Ekenel

Duration: 01. January 2012 – 30. June 2012

Computer Vision for Human-Computer Interaction Research Group
Institute for Anthropomatics
Karlsruhe Institute of Technology
Title: Comparative Analysis of Content-based TV Genre Classification and Web Video Categorization
Author: Tomas Semela

Tomas Semela
Morgenstr. 1
76448 Durmersheim
t.semela@gmx.de

statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others

Karlsruhe, 30. June 2012

.....
(Tomas Semela)

Abstract

Efficiently indexing and retrieving multimedia data from huge collections is a very important research field today. This study presents a content-based classification system for TV genre prediction and web video categorization. The approach is to classify TV source material with genre tags and web source with category tags only using content–based cues from the videos without any metadata or other information. Examples for TV genres are *cartoons*, *commercials*, *news* and *sports*. Web videos are grouped into categories like *autos*, *education*, *blog* and *travel*. The classification of web video material proves to be more difficult. Categories do not reflect the same properties as genres and the diversity in web categories is very high. Therefore, it is much more difficult to draw conclusions from content-based features about the category a video should belong to. Also, often videos clearly belong to multiple categories like for example news or a home video about a sports event. This thesis presents a comprehensive related work section, which also covers all research fields affecting video indexing. The presented system for genre or category classification utilizes cues from visual, structural, aural and cognitive properties to predict tags with different machine learning algorithms like SVMs or tree classifiers. The system is enhanced with the possibility to extract and utilize several different SIFT descriptors in bag–of–words fashion. Cognitive and visual features, and SIFT descriptors are extracted from keyframes. The system is evaluated on three datasets from both video domains, two datasets from Italian and French TV channels and one from the video portal YouTube.

The results on the TV domain datasets are excellent and bring classification rates of 98.1% and 94.7% on the Italian and French dataset respectively. The content–based classification on the TV domain is robust enough to handle the small diversity and number of genres. A peak performance of 44.0% on the YouTube dataset confirms the difficulties of the web domain. The high diversity, high number of categories and the categories themselves prove to be a challenge for a content–based system. Nevertheless, the experiments showed several interesting points. Performance of face detection based cognitive features drops significantly when applied on keyframes and not all frames of a video. Specific features work better for some genres than others or a fusion of many features. This may suggest to use individually chosen feature sets or even single features for each genre or category instead of one overall classification framework. Also, visual features are found to be the best working features, achieving the most correct class predicitions for a majority of genres and categories.

Kurzzusammenfassung

Statistiken über das Video Portal YouTube besagen, dass jede Minute 72 Stunden an Videomaterial hochgeladen werden. Diese enorme Datenmenge und das stetige Wachstum erfordern es, diese Daten zu managen. Das bedeutet, die Daten in bestimmte Gruppen zu gliedern um das Archivieren und Suchen zu vereinfachen. Bei YouTube z.B. werden Videos durch User hochgeladen, während diesem Prozess einer vordefinierten Kategorie zugeordnet, und mit verschiedenen Informationen ergänzt. Zu diesen Informationen zählen der Titel, eine Beschreibung und Tags. Mittels textbasierter Suche können die Videos später anhand dieser Daten gesucht und gefunden werden. Beispiele für Kategorien sind Tiere, Autos, Entertainment, Sport, Nachrichten oder Blogs. Dieses Verfahren bringt einige Probleme mit sich. Erstens ist eine gewisse Anstregung seitens des Uploaders erforderlich ein Video hochzuladen und zweitens sind die Informationen des Uploaders möglicherweise unzuverlässig und unvollständig. Um einen robusten, effizienten und einheitlichen Upload der Videos zu gewährleisten, erfordert es ein automatisches System, welches ohne begleitende Informationen auskommt und Videos anhand seines Inhalts kategorisiert. Web Videos sind aber nicht allein verantwortlich für den rasanten Zuwachs an Videodaten im World Wide Web. Auch Rundfunksender haben eine stetig anwachsende Kollektion ihrer Fernsehprogramme, welches es zu archivieren und organisieren gilt. Im Gegensatz zu Web Videos, die nach Kategorien geordnet werden, sind TV Inhalte in Genres gegliedert. Beispiele für Genres sind Cartoons, Werbung, Nachrichten oder Wetterberichte. Auch hier und aus dem selben Gründen wie in der Web Video Domäne ist ein automatisches System wünschenswert.

Im Laufe der letzten Jahre wurden sehr viele Studien über automatische Videoklassifizierung in beiden Domänen durchgeführt. TV–Klassifizierungssysteme arbeiten alle nach ein und demselben grundlegendem Prinzip. Der Inhalt eines Videos wird auf verschiedenen Ebenen analysiert, verschiedene Informationen und Statistiken werden daraus gewonnen und diese werden mittels Maschinenlernverfahren klassifiziert. Seit dem Anfang dieser Forschungsrichtung Mitte der 90er Jahre wurden bereits multimodale Ansätze verfolgt. In der Regel werden Informationen aus den Bildern, dem Audiosignal und anderen Modalitäten gesammelt. Audiosignale werden z.B. in Bereiche wie ”Stille”, ”Geräusche”, ”Musik” oder ”Sprecher” segmentiert. Bildinformationen wie Farbverteilungen oder Texturen werden aus den Bildern gewonnen. Statistiken werden auch über Schnitte und deren Laufzeiten oder die Anzahl der Gesichter pro Bild gesammelt. Diese Informationen, so genannte ”low–level features” werden als Eingabe für Maschinenlernverfahren verwendet. Beispiele für solche Verfahren sind ”Support Vector Machines”, ”Neuronale Netze” oder ”Entscheidungsbäume”. In der Web Video Domäne ist es auch üblich, zusätzliche Informationen wie den Titel, die Beschreibung und die Tags zu verwenden und Dokumentenklassifizierungsver-

fahren darauf anzuwenden. Es ist ersichtlich, dass das Forchschungsgebiet der Videoklassifizierung eine große Anzahl an verwandten Forschungsgebieten zusammenführt.

Diese Arbeit präsentiert ein automatisches Klassifizerungssystem, um Videos in vordefinierte Genres oder Kategorien zu klassifizieren. Als Aufgabe wird dieses System überarbeitet und erweitert, um in beiden Domänen Videos erfolgreich zu kategorisieren. Verwandte Arbeiten befassen sich ausschliesslich mit einer der beiden Domänen und die Leistungsfähigkeit der vorgestellten Systeme wird nur auf einem der beiden Domänen getestet. Diese Arbeit nimmt es sich zum Ziel, ein Domänen übergreifendes System zu präsentieren und es auf Daten beider Domänen zu evaluieren. Des Weiteren wird ein umfangreicher Einblick in die beteiligten Forschungsgebiete und verwandten Arbeiten gewährt.

Das vorgestellte System extrahiert Informationen aus vier verschiedenen Informationsbereichen. Es werden Merkmale wie "Mel–Frequency–Cepstral–Coefficients" oder "Zero–Crossing–Rate" aus dem Audiosignal gewonnen. Mittels Gesichtserkennung werden Statistiken über die durchschnittliche Anzahl an Gesichtern und deren Verteilung gesammelt. Visuelle Informationen über Farben und Texturen werden zusammengetragen und Statistiken über wechselnde Aufnahmen berechnet. Diese Informationen werden als Eingabe für SVMs zur Klassifikation verwendet. Das vorhandene System wird um ein "Schlüsselbild" (Keyframe) Extraktionsmodul erweitert. Informationen aus dem Bildbereich werden nicht mehr auf allen Frames eines Videos sondern nur noch auf ausgewählten Frames berechnet. Als neue Merkmale können verschiedene SIFT–Deskriptoren extrahiert werden. Die Maschinenlernverfahren "Entscheidungsbäume" (Decision Trees) und "Entscheidungswälder" (Random Forests) werden dem System hinzugefügt und evaluiert.

Das System wird auf zwei Datensätzen der TV Domäne und einem YouTube Datensatz evaluiert. Die Ergebnisse zeigen, dass das System in der TV Domäne bereits sehr zuverlässig arbeitet, und 98.1% und 94.7% der Videos korrekt klassifiziert werden. Auf dem YouTube Datensatz wird eine Klassifizierungsrate von 44.0% erzielt. Die Evaluation zeigt weiterhin, dass die besten Ergebnisse mit unterschiedlichen Merkmalen gewonnen werden und bestimmte Merkmale sich besonders leistungsstark bei bestimmten Genres oder Kategorien zeigen. Es wird auch ersichtlich, dass die Fusion verschiedener Merkmale im Gegensatz zu einem Merkmal die Klassifizierungsrate manchmal verringert. Daraus folgt die Annahme, dass individuelle Klassifizierungsverfahren für jedes Genre oder jede Kategorie zuverlässiger arbeiten könnten. Die Klassifizierung mit SVMs übertrifft die beiden Varianten der Entscheidungsbäume bei allen Datensätzen. Bei den Merkmalen heben sich die SIFT–Deskriptoren, gefolgt von den anderen visuellen Merkmalen deutlich positiv von den restlichen Merkmalen ab.

Ansätze für zukünftige Verbesserungen beinhalten die Verwendung von temporalen Merkmalen, "Tracking" basierter Gesichtserkennung, sowie die Segmentierung des Audiosignals und Dokumentklassifizierung extrahierter automatischer Spracherkennung. Die automatische Spracherkennung sollte vor allem in der Web Video Domäne zu einer deutlichen Verbesserung der Klassifizierung führen, da die Kategorien in dieser Domäne mehr dem inhaltlichen Thema nahekommen als bestimmten Genremerkmalen. Oft unterscheiden sich die Videos nur am Gesprächsthema und lassen sich durch Bildmerkmale nicht korrekt klassifizieren.

Contents

List of Figures

List of Tables

1. Introduction

Today's multimedia growth in the world wide web seems without limit. Contributing to the vast amount of digital videos are video portals like *YouTube*. Above all the possibility for users to upload their videos has generated an incredible increase in video data on the World Wide Web. Checking YouTube statistics[1] 72 hours of video material are uploaded every minute. The rapid growth of YouTube data over the years is shown in Figure 1.1 and this quantity of data comes from only one, although the most popular, video portal.

Another big amount of media arises from the collections of broadcast empires like *BBC* and *CNN* as well as other broadcast channels and service providers, which archive their huge amount of broadcast programs accumulated over the years and add their new content to their libraries to make them available digitally for online viewing.

1.1. Motivation

This enormous amount of data enforces the research for automated systems to manage the data collections, making indexing and retrieving simple, efficient and inexpensive, yet consistent and robust. The development of such automated systems plays a crucial role in today's *multimedia analysis and retrieval* research area.

Videos from the YouTube portal are separated by a fixed number of categories. During the upload process the user manually assigns the video to one of several predefined categories, e.g., sports, news or entertainment, and adds information like title, tags and description to it. This category and surrounding metadata is then used for text–based search systems currently employed at these portals. Several problems arise with this course of action. First the process of uploading a video requires manual and time consuming work from the user. Furthermore, the information and category chosen by the uploader is subjective to his opinion, may be inaccurate and incomplete, and inconsistent with the information uploaded for similar videos by different users. For example different users may upload the same video, showing news about a popular movie star, in different categories like news or entertainment.

[1] `http://www.youtube.com/t/press_statistics`, access June 2012

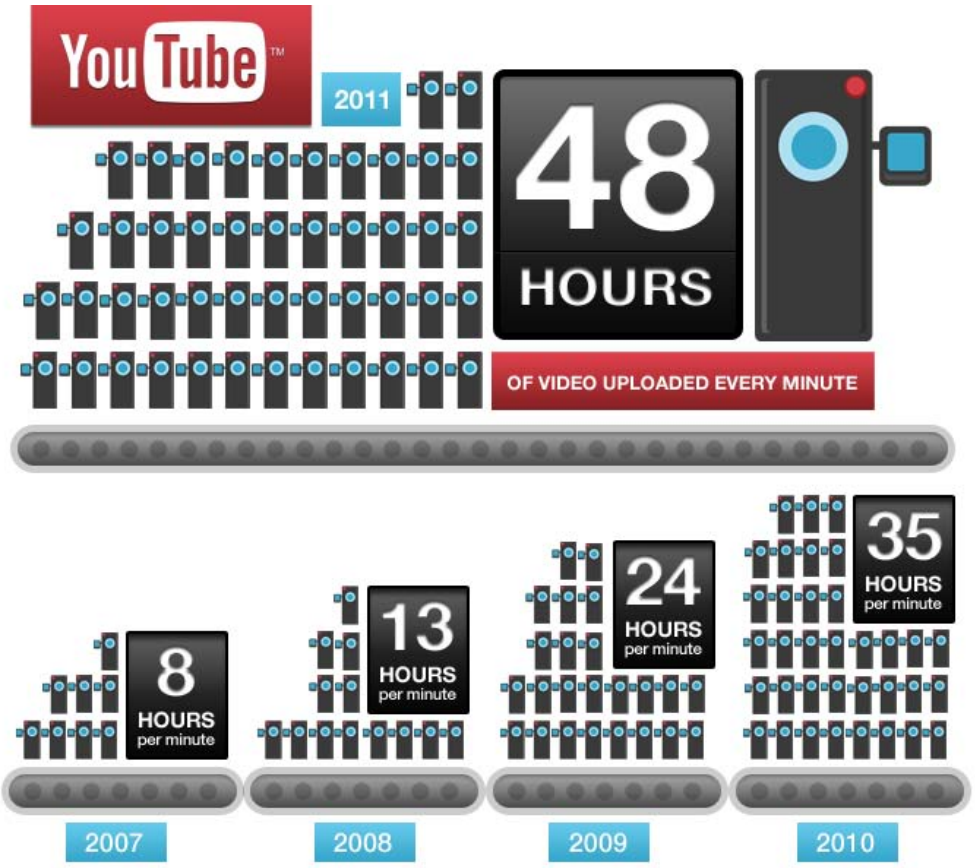


Figure 1.1.: YouTube growth over the past years

The management of broadcast media deals with similar problems. Extra human effort is needed to categorize and archive the data. Manual indexing by human effort is expensive, time consuming and faulty. Uploaders and broadcast channels would benefit greatly from an automated and reliable indexing and retrieval system based on the content of the media alone thus eliminating human error and saving time of the user. In addition, the search systems would benefit from a more consistent and accurate database independent from any outside source.

The content of the video material itself offers a great amount of information from a wide range of modalities, which can be utilized properly by an automated content–based system. Borrowing from different computer vision and audio analysis research fields, many audio–visual cues can be extracted from each video to achieve this task, ranging from visual information like color information, face– and object detection, to audio analysis like audio segmentation and *automatic speech recognition* (ASR) and characteristics of the videos derived from shot detection. In the end these are the same cues we humans apply for this task.

Many aspects of image and audio processing were researched extensively over the past years and they all have the common goal to find a solution for a specific problem, i.e. to bridge the *semantic gap* between extracted low–level features and high-level semantic

concepts.

A lot of research has already been conducted on the topic of genre classification in the music and video domain. Classifying video material into genres began in 1995. Today the research shifts to the more challenging domain of web videos. Due to the possibility to use surrounding metadata as well as additional information like related videos and social information, a lot of studies focus including this information in the classification or retrieval systems.

1.1.1. Difference between the TV and Web Domain

The TV and web video domain distinguish themselves in several different ways. For a better understanding of the separation of these domains and the specific challenges for genre classification in these domains, their characteristics are described next. An overview is available in Table 1.1.

TV domain. The TV domain is the easier one of the two domains. Usually datasets from this domain consist of material from the same TV channel or broadcast group, e.g., the Italian RAI dataset from *Montagnuolo and Messina* [MM07]. Genres for classification are not too difficult and have a lot of distinguishable properties. Example genres are *cartoons*, *commercials*, *news*, *sports*, *movies* or *game shows*. The variety in broadcasts like *news*, *weather forecast* and *talk shows* is very low because the production and style do not change. The same properties apply to *cartoons* and *television series* to a certain degree, too. These shows reoccur in the program of one broadcast channel on a regular basis. The production values and style of TV content is professional and is provided in excellent video quality. In contrast, web video portals contain a lot of amateur content in low quality. The characteristics of TV domain videos lead to an overall small variety in content and make the task of genre classification easier than the one in the web domain.

Web domain. The arrangement of web videos is done in a different way compared to TV content. Even though they share some of the genres like *news* and *sports*, the web videos are usually organized in categories like *blogs*, *educational*, *travel*, *religion* and *fitness*. The web domain unlike the TV domain contains content from many different sources ranging from professional and semi–professional content to amateurishly produced video material, thus the variation in production value, style and quality varies a lot. The main problem is user generated content and the possibility to upload data from every kind of device with a camera. The possibility for the uploaders to add surrounding metadata by themselves is another difference to the TV domain. To better understand the difficulty of classifying videos into categories in the web domain, the following examples are given:

- Weather forecasts from many different sources have many different properties and styles making them harder to identify than the single weather forecast repeating on one TV channel.
- Users upload content filmed with web cams or phones varying greatly in production style and quality. Amateur videos of *weather forecasts*, *news*, *cartoons*, and *talk shows* make the categories even more challenging. Categories like *blogs*

can have many different topics (hobbies, tutorials and reviews), presentations and visual styles.

- Another good example for variety in one category is the entertainment sector in YouTube. Music related videos, a small portion in the entertainment category come in many different types: There are videos from live performances like big concerts or small concerts. The category also includes lessons and tutorials for music instruments and songs. Other videos show people singing and official music videos or videos with the song lyrics. Lyrics videos can be full music videos, still pictures or only showing the lyrics in front of a black background.

domain	TV domain	web domain
diversity in style	low	high
quality	good	bad – good
style	professional	amateur – professional
genre boundaries	easy defined	overlapping and difficult
variation inside genres	little	high

Table 1.1.: TV vs web domain

1.2. Problem definition

Many different terms are used to describe the task of genre tagging in different research fields. TV programs are always classified into *genres*, as for web video content *categories* are used. Further the terms *classifying*, *tagging* and *indexing* are sometimes used interchangeably. They all describe the same task of assigning a label to a video to group videos with similar properties.

Organizing large collections of data in a meaningful way is necessary to manage the data easy and reliable. In case of multimedia data like videos, the approach of arranging the data into genres or categories is common. The task of this study is to extend an automated system for TV genre classification and evaluate its performance in both the TV and web domain. Data from the TV domain will be classified into several predefined genres while web videos will be tagged into different categories. Only content–based information of the video itself will be used to achieve this, dismissing any surrounding metadata, other information and the need of human influence on purpose. This approach is taken to simulate the realistic conditions in the World Wide Web, because the surrounding metadata can be missing or might include faulty information.

The system is based on the classification approach for tagging videos. Several low–level features from different modalities will be extracted to generate a classifiable representation of the input data. Different machine learning algorithms will be used to predict genre or category tags for each sample video. Two different datasets from the TV domain and one dataset from the web video domain will be evaluated on this system, using parts of the datasets for training supervised machine learning techniques on the extracted features. The simple goal is to classify a new video into the correct genre or category for its respective dataset. Groundtruth labeling is based on the information provided with the datasets and will be described in detail in Chapter 5.

1.3. Contribution

This work improves the baseline system developed at the *Institute of Anthropomatics at the KIT*. This system, already able to classify videos into genres from several features from different modalities, is extended for the categorization of web videos. The web video data is collected from YouTube with videos from the actual top level hierarchy categories.

Several new additions, e.g., features and classification methods are added to the system to achieve this goal and investigate their usefulness for this task. Existing features and classifiers are further improved or updated, e.g., the face detection module. New additions include several *SIFT* feature descriptors and *Decision Tree* and *Random Forest* classifiers.

This work will also show the limitations of the content–based genre classification on the web video domain. Due to the high variety in the dataset, the excellent results achieved on the TV domain are not expected to be repeated. This could indicate that *multilabel* and *hierarchical* classification may be unavoidable for better content–based web video tagging systems in the future, compensating the fact that the currently used features are not sufficient enough to distinguish between blurry genre boundaries and high diversity in the web video domain.

Topics like multilabel and hierarchical classification as well as document classification on ASR transcripts will not be part of this work and are beyond the scope of this thesis. However, to get a complete overview of genre classification research field, the "Related Work Chapter" will present all important topics affecting genre classification.

1.4. Outline

Chapter 1, the introduction, deals with the situation of todays multimedia growth, resulting in the motivation of this research. It gives insight into the different domains and individual problems. Finally it establishes the contributions of this study.

Chapter 2 focuses on the related work in the research field of genre and category classification. Important achievements in both the TV and web video domain are presented, respectively. The chapter embraces many different topics that are not part of this thesis but relevant or already used for genre classification.

Chapter 3 describes the methodology of the proposed system for video tagging. The state of the system before this thesis is described first, followed by the additions and enhancements that are part of this thesis.

The implementation of the system is specified in detail in Chapter 4. The chapter is divided into the separate modules of the system, which include shot detection, the C++ framework, the audio analysis module and the SVM classification scripts.

The system is extensively evaluated on different datasets and results are presented in Chapter 5. The chapter contains information on the used datasets and the parameter specifications for the evaluation. The results are compared to the old results of the baseline system as well as related work from Chapter 2.

Chapter 6 presents the conclusions which arise from the results of this thesis. Additionally several ideas for future improvement of the system are introduced.

2. Related Work

This chapter tries to give an insight into the diversified research field of genre classification. First in Section 2.1, TV genre classification related work is presented beginning from the earliest work in 1995 by *Fischer et al.* [FLE95] to the current state–of–the–art work from *Montagnuolo and Messina* [MM09] in 2009. Several related works from the web video domain of category tagging are presented in Section 2.2 followed by an extensive overview over all the research areas in genre classification in Section 2.3.

	Time-Domain				Frequency Domain			
Paper	RMS/Energy	Subband	ZCR	Other	Freq. Centroid	Bandwidth	Pitch/Fund. Freq.	MFCC
Dinh et al. [29]		X	X					
Fan et al. [8]	X						X	
Fischer et al. [35]	X							
Huang et al. [36]	X	X	X		X	X	X	
Jasinschi and Louie [19]			X			X	X	X
Lee et al. [37]		X						
Liu et al. [27]	X	X	X		X	X	X	
Liu et al. [38]	X	X	X		X	X	X	
Moncrieff et al. [5]	X							
Nam et al. [4]	X							
Pan and Faloutsos [39]				X				
Qi et al. [21]				X				
Rasheed and Shah [40]	X							
Roach and Mason [41]								X
Roach et al. [42]								X
Wang et al. [13]			X			X		X
Xu and Li [43]								X

Figure 2.1.: Comparison of audio utilization in the TV domain by *Brezeale and Cook* [BC08].

2.1. TV genre classification

In the past years a lot of research went into classifying TV programs into genres. To give a quick overview of this field only the first and the most recent work is presented in more detail. For more interested readers an extensive overview of the most important work done over the years can be found in the works of *Montagnuolo and Messina* [MM07][MM09] and *Brezeale and Cook* [BC08]. *Montagnuolo and Messina* [MM09] provide an excellent

introduction and work overview in their study and *Brezeale and Cook* [BC08] present a survey of the literature for automatic video classification. The survey presents studies for video classification divided into categories depending on which features they utilize. Examples of these studies are shown in Figure 2.1 for audio features and in Figure 2.2 for visual features. The figures show that many different features were evaluated for video classification for each modality. Information about a third category for text features and studies combining the different modalities can be checked in the survey by *Brezeale and Cook* [BC08].

	Color-Based			Shot-Based		Object-Based			MPEG	Motion-Based			
Paper	Color	Texture	Edge	Trans.	Length	Face	Text	Other	DCT	Motion Vectors	Optical Flow	Frame Diffs	Other
Iyengar and Lippman [59]					X						X	X	
Girgensohn and Foote [74]	X												
Wei et al. [56]				X		X	X						
Dimitrova et al. [66]						X	X						
Truong et al. [60]	X			X	X						X		
Kobla et al. [7]							X			X			
Roach et al. [75]												X	
Roach et al. [76]											X	X	
Pan and Faloutsos [77]									X				
Lu et al. [64]	X												
Jadon et al. [63]				X	X						X		
Hauptmann et al. [2]	X	X	X										
Pan and Faloutsos [39]	X												
Rasheed et al. [62]	X				X								
Gibert et al. [78]	X									X			
Yuan et al. [65]	X				X	X							X
Hong et al. [79]	X	X									X		
Brezeale and Cook [18]									X				
Fischer et al. [35]	X			X	X						X	X	
Nam et al. [4]	X						X						X
Huang et al. [36]	X									X			
Qi et al. [21]	X												
Jasinschi and Louie [19]	X		X			X	X		X				
Roach et al. [42]												X	
Rasheed and Shah [40]	X				X								X
Lin and Hauptmann [20]	X												
Lee et al. [37]			X						X				
Wang et al. [13]					X	X	X			X			
Xu and Li [43]	X	X								X			
Fan et al. [8]	X	X						X					

Figure 2.2.: Comparison of visual utilization in the TV domain by *Brezeale and Cook* [BC08].

TV genre classification was first presented in 1995, when *Fisher et al.* [FLE95] developed a system to classify TV programs into five different genres of *news*, *tennis*, *car race*, *commercials* and *cartoons*, working with a wide range of multimodal information. The three steps of the system include syntactic analysis of the raw video material collecting basic statistics, derivation of style attributes from these statistics and finally predicting a genre from these style attributes. Already in the beginning it was clear to utilize as much information as possible to accomplish this task. *Fisher et al.* [FLE95] derived information from *color statistics*, *cut detection*, *camera motion*, *object motion* and *audio*. These low–level statistics provided a more abstract level of video analysis such as *scene length* and *scene transitions*, *camera panning* and *zooming*, *speech* and *music*.

Two of the most recent and extensive works in TV genre classification come from *Montagnuolo and Messina* [MM07] [MM09] with the 2009 work being an update and new state–of–the–art of the previous work. Their video genre classification research also includes focusing on fuzzy mining and classifiers [MM08a][MM08b]. Since our baseline system is inspired by [MM07], which is a perfect example of the general approach in TV genre

classification, the system is presented in little more detail.

2.1.1. TV Genre Classification Using Multimodal Information and Multilayer Perceptrons [MM07]

This work presented the most extensive experiments in TV genre classification to date. It was the first study to run experiments on complete TV broadcasts from the largest dataset collected for this research. Classifying the highest number of genres, the study achieved an overall best classification accuracy.

The dataset was collected from complete TV programs from three different Italian TV channels, which means that instead of small sample clips, complete TV broadcastings with running times over an hour like football or talk shows were used. The total amount of data is 111 hours. All videos belong to one of the seven genres: *cartoon*, *commercial*, *football*, *music*, *news*, *talk show* or *weather forecast*.

The attention of this work lies on the multimodal information combination and the classification using *Multilayer Perceptrons* (MLPs). The multimodal information is accumulated into four feature vectors, each belonging to a different category. These categories are:

1. **Structural:** Syntactic information from shot boundary detection, e.g., relationships between frames, shots and scenes
2. **Visual:** Physical properties perceived by the users like colors, shapes and motion
3. **Cognitive:** Information related to high–level semantic concepts like faces
4. **Aural:** Audio analysis of noise, speech and music

Visual. The visual feature vector consists of seven sub–features. Color is represented by hue, saturation and value. Luminance is represented by gray scale and textures are described through contrast and directionality features. Temporal activity information is collected based on displaced frame difference. Each feature is modeled by a 10–component *Gaussian Mixture Model*. The final feature vector is 210–dimensional, consisting of the GMMs weight, mean and standard deviation values ($7 \times 10 \times 3 = 210$).

Structural. Automatic shot detection information is gathered to produce two structural features. For one the rhythm of the video is expressed through average shot length and two, shot length distribution is saved in a 65–bin normalized histogram using 64 uniformly distributed bins for shot lengths between 0 and 30 seconds and the last bin for longer shots.

Cognitive. Frontal face detection is applied to model three features. Face distribution along the video with information of the number of faces per frame, face positions in a frame in a 3×3 grid and average number of faces in the video. All features are normalized.

Aural. The audio signal is segmented into seven classes: *speech*, *silence*, *noise*, *music*, *speaker*, *speaker and noise*, *speaker and music*. An ASR system is also applied to the audio signal. Normalized duration values for each audio class and average speech rate over the entire video are computed.

Each of these four feature vectors serve as an input for one *neural network* classifier. The authors use *k–fold cross validation* for training and testing, splitting the dataset into 6 disjoint subsets of equal size, using each subset for testing while the other 5 subsets are used for training. The outputs of the neural network classifiers are combined and averaged for all genres and finally the genre with the highest probability is picked. The classification pipeline is shown in Figure 2.3.

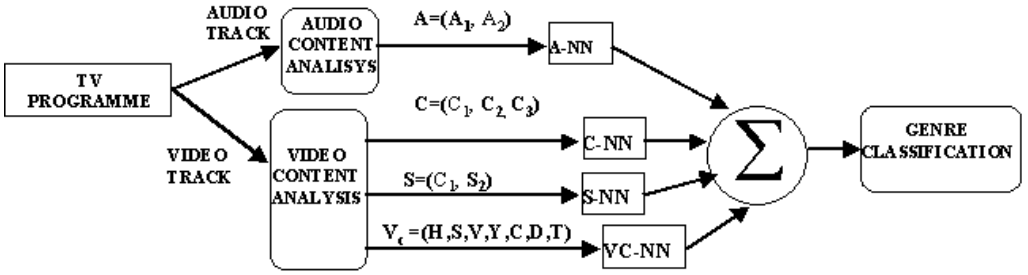


Figure 2.3.: System overview by *Montagnuolo and Messina* [MM07].

An overall classification accuracy of 92% was reached on the evaluation dataset. Improving their system in [MM09] utilizing new and modified features, the accuracy was further boosted to 95% for the same dataset. Improvements include structural and cognitive feature vector improvements.

2.2. Web video categorization

In recent years the focus on genre classification shifted towards the rapidly growing web video domain. The web video domain provides a much more difficult challenge compared to the TV domain. The multimedia data is not arranged through genres but through categories. The number of categories is often very high and even taxonomic. The data itself is very diverse and sometimes not identifiable belonging to one category. The fast growth of web video portals like YouTube demands research for an efficient and automated management system. For example, progress in this domain is currently pushed in the *Tagging Task* of the *MediaEval Benchmarking Initiative for Multimedia Evaluation*[1].

2.2.1. Web Related Work

More diverse content and different kind of categories than in TV domain makes it more difficult for content–based systems to achieve high accuracy rates. Most systems for this domain try to incorporate additional information surrounding the videos like *title*, *tags*, and *description*.

Several studies have been conducted on the web video domain in the past years. Some examples are the works of *Wu et al.* [WZN09], *Ulges et al.* [USKB07], *Song et al.* [SZZ+09], *Zhang et al.* [ZSC+09], *Borth et al.* [BHK+09], *Yang et al.* [YLYH07], *Song et al.* [SZYW10] and *Wang et al.* [WZS+10]. Most of them were studies as part of the *Multimedia Grand Challenge*[2]. A general overview of their characteristics can be viewed in Figure 2.4.

[1] `http://www.multimediaeval.org/mediaeval2012/tagging2012/index.html`
[2] `http://comminfo.rutgers.edu/conferences/mmchallenge/2010/02/10/google-challenge/`

From these seven examples only *Ulges et al.* [USKB07] relies purely on content and only *Wu et al.* [WZN09] on surrounding metadata combined with other sources, while all others include both content and metadata and even additional sources. In the work of *Wu et al.* [WZN09] these additional sources are related videos presented in search results of the online video portals and other videos uploaded by the same user indicating personal interests of the user. *Song et al.* [SZYW10] adapt document classifiers to benefit from large document datasets labeled with the same categories for training their classifiers in the video domain. *Wang et al.* [WZS+10] benefit from both related videos and text–based web pages as additional information sources for training and classification.

	Content	Text	Taxo-nomy	Multi-label	Other sources	LSA	Local / SIFT	Key-frames	Aural	Struct	Cognitive
[WZN09]		X			X						
[USKB07]	X						X	X			
[SZZ+09]	X	X									
[BHK+09]	X	X	X			X	X	X			
[YLYH07]	X	X		X		X		X	X		
[SZYW10]	X	X	X	X	X		X	X	X		X
[WZS+10]	X	X	X		X		X		X	X	X

Figure 2.4.: Comparison of web video related work

SIFT features introduced by *Lowe* [Low04] dominate the content–based visual features and have already proven to be very useful in high–level concept detection like in the *TREC Video Retrieval Evaluation* (TRECVID[3]). But also all different kinds of features from aural, cognitive and structural cues are used as can be seen in Figure 2.4.

The more diverse but also interlaced categories call for hierarchical and multilabel classification as proposed by *Borth et al.* [BHK+09], *Yang et al.* [YLYH07], *Song et al.* [SZYW10] and *Wang et al.* [WZS+10]. Finally, two systems from *Borth et al.* [BHK+09] and *Yang et al.* [YLYH07] tried to bridge the semantic gap of low–level features and high–level concepts with the *Latent Semantic Analysis*.

2.2.2. Web Video Datasets

Several works tried to analyze web video domain data and provide benchmark sets with additional information. Benchmark sets were created by *Zanetti et al.* [ZZMP08], *Loui et al.* [LLC+07] and *Cao et al.* [CZS+09].

[3] `http://trecvid.nist.gov/`

Loui et al. [LLC+07] offer a video benchmark set with a large number of user videos and annotation of concepts. The dataset includes videos from Kodak and web videos from YouTube, extracted visual features for Kodak videos (edge direction histogram, gabor filter, and grid color moment) and download links for YouTube videos. Furthermore the Kodak video dataset comes with keyframes (1 per 10 seconds), while YouTube videos include metadata information like title, tags, category, and author. The videos are annotated with over 100 concepts in seven categories.

Zanetti et al. [ZZMP08] present well established classification algorithms from the TV domain and evaluates them on the web video domain. A dataset from YouTube is created and differences and difficulties compared to the TV domain datasets are presented as well as evaluational results using the TV domain systems. It showcases the difficulties of classification on diverse and challenging video material.

The *MCG–WEBV* benchmark dataset from *Cao et al.* [CZS+09] consists of 80,000 of the most viewed videos from YouTube in a small time period at the beginning of 2009. The data comes along with a wide range of additional information, features and metadata. Information includes low–level visual (color histograms and moments, edge histograms, wavelet texture and co–occurrence texture), textual (textual vector space model) and aural features. Keyframes as well as metadata (ID, uploader, date, length and category) and web features (rating, tagging, title, description, related videos, # of views, # of comments and # of shots) are part of this benchmark set, too. Groundtruth labels come straight from the YouTube categories and 'hot web topics' are annotated with human effort.

2.3. Related Research Fields

By now we established the fact that many research topics influence genre classification. This section covers related work about these topics. Some of it comes directly from genre classification research while others address a topic by itself and most of the work applies to the greater topic of semantic concept detection.

2.3.1. Shot Boundary Detection

Automatic shot boundary detection can be viewed as one of the foundations of genre classification. Not only as a possibility to extract features from shot information like in the work of *Montagnuolo and Messina* [MM07], but also as the initial step to segment a video into useful and meaningful chunks for further processing. Being a building stone in genre classification, a reliable working shot boundary detection system is necessary, not to propagate errors through the whole classification process.

Early work on shot boundary detection was done by *Zhang et al.* in 1993 [ZKS93]. Simple *cuts* and difficult shot transitions like *dissolves* and *fade in/outs* are the focus in this work, trying to successfully distinguish between special transitions, camera zooming and panning, and object movement. A survey of shot boundary detection research up to 2001 is presented by *Lienhart* [Lie01].

Even today the work on shot boundary detection is not finished. Researchers are trying to perfect the methods on various of different and difficult shot transition types for difficult video material. A more recent overview of shot detection research conducted as part of the TRECVID evaluation during the past years is given by *Smeaton et al.* [SOD10].

2.3.2. Content–based Feature Extraction

Content–based features play an important role in genre classification besides the classification algorithms. This is especially true in TV domain, where additional information is not common. Figure 2.4 shows that studies utilize content–based features in standalone fashion or in combination with additional sources. This subsection highlights related work to all kind of different content–based features used in genre classification or similar semantic concept detection tasks.

Audio Analysis

Audio material or the audio source of a video plays a crucial role in human perception to recognize different programs and presents a rich information source for any kind of work related to media indexing. Research areas include music genre classification, audio segmentation, automatic speech recognition and music modeling.

Each of this research areas provide information or ideas that can be adopted in video genre classification. For example segmenting audio into different categories like silence, noise, music and speech and using these statistics to distinguish genres like proposed by *Montagnuolo and Messina* [MM07] [MM09]. Early work began 1996 with *Saunders* [Sau96] where radio broadcasts were discriminated into speech and music parts using features like the *Zero Crossing Rate* (ZCR). The ASR system used by *Montagnuolo and Messina* is described in the work of *Brugnara et. al.* [BCFG00].

One of the most popular and useful features in audio analysis was presented by *Beth Logan* in [Log00], namely the *Mel Frequency Cepstral Coefficients* (MFCCs). The MFCCs provide rich information and proved their usefulness in many audio related tasks over the years. Audio segmentation using MFCCs, ZCR and many different features with SVM classification into different categories is presented by *Lu et al.* [LLZ01].

The similar task of musical genre classification is presented by *Tzanetakis and Cook* [TC02], where music clips are classified into different genres like *classic*, *jazz*, *rock* and *hip hop*. The goal in music genre classification is the same, to provide robust automatic data annotation that is usually done manually. In the proposed classification hierarchy, data is first distinguished between speech and music and further into a wide range of sub–categories for music.

Video Analysis

Most of the important visual features in genre classification will be presented in the methodology chapter as they are part of our genre classification system. Examples are color histograms, color moments, co–occurrence matrix, color correlograms, edge and wavelet textures, and local features like SIFT.

2.3.3. Semantic Concept Detection

Semantic concept detection in general and event detection in particular are evaluated each year extensively in the TRECVID program. For the state–of–the–art results interested readers can, for example, look at the Media Mill studies by *Snoek et. al* [SvdSdR$^+$10]

[SvdSL+11]. Their combination of many different SIFT descriptors and other color features combined with SVM classification is proved to be very successful in the past years.

One research area in computer vision engages with task of detecting and recognizing actions in scenes as a part of high–level concept detection. Actions are like *walking*, *running*, *hand shaking* or *opening a door*. These actions can prove to be useful semantic concepts and high–level features in both video domains for genre tagging. Typical low–level features in this line of work are *spatio–temporal* features using *Histogram of Optical Flow* (HoF) or *Histogram of Gradient Orientations* (HoG) descriptors. An up–to–date example of this research comes from *Sadanand and Corso* [SC12].

Another approach to bridge the semantic gap is the *Latent Semantic Analysis* (LSA). Through LSA the relationship between semantic concepts and feature vectors can be analyzed. It produces a set of topics related to the concepts and features. In the order of appearance, LSA was introduced 1990 by *Deerwester et al.* [DDF+90], followed by *Landauer et al.* [LFL98] in 1998 and extended to *Probabilistic Latent Semantic Analysis* by *Hofmann* [Hof99] in 1999.

An interesting study conducted by *Yang and Hauptmann* [YH08] looks critically at the work of high–level semantic concept detection. The authors raise attention to the fact that concept detectors are more likely copy detectors instead of capturing the essence of the concepts they try to detect. They argue by showing the poor cross domain results of these systems.

2.3.4. Document Classification

Useful for annotation of large textual datasets, like with audio and video multimedia data, or for indexing and searching web pages on the web, document classification precedes multimedia classification and has served as a forerunner for multimedia classification methods. Like in the multimedia domain very large datasets of text or websites are needed to be annotated for indexing and retrieval. A popular example for this is the *bag–of–words* model adapted in computer vision. Also the original document classification systems are applied to web video surrounding information like title, tags and description.

Chen and Ho [CH00] offer a perfect example for typical processing steps in document classification systems. After acquiring a dataset for processing, important decision have to be made about feature selection and extraction, pre–processing and classification method. In document classification features often measure word frequency in documents and the whole corpus. Important for such features are dimensionality reduction methods, since the feature space is often very large and sparse, for example if one wants to count the occurrence of every word in a text as a feature.

Steps for feature dimensionality reduction and feature vector creation include segmentation and transformation of text into *lowercase*, *stopword removal*, *stemming*, *term selection* and feature extraction.

Stopwords. Stopwords are terms that have no semantic meaning in documents and can be removed without losing any information. Typical stopwords are *and*, *or*, *the*, *are* and *again*, and are removed simply according to a stopword list.

Stemming. Stemming reduces terms to their roots downsizing the feature space, e.g., *ask*, *asking*, *asked*, *asks* are all reduced to *ask*.

Term selection. An important step is not only to remove high occurrence terms like stopwords but also low occurrence terms that appear only occasionally and carry not enough information for the classification. A comparison of feature selection methods was conducted by *Yang and Pedersen* [YP97]. Typical term selection models are document frequency (DF), information gain (IG), mutual information(MI) and χ^2–Test (CHI). Results indicate that the computationally most affordable term selection is document frequency with a threshold. It simply counts the number of documents the term appears in and is removed from the feature space if a threshold is not reached.

Feature extraction. The final feature vector is computed using a vector model representation computing the *term frequency – inverse document frequency (tf–idf)* measure. The *tf–idf* measure is a statistical measure to evaluate how important a term is to a document in a collection of documents. The *term frequency* counts the number of times a term appears in the current document. The *inverse document frequency* computes the importance of the term for the whole document collection. The more often a term occurs in documents, the less it is suitable for classification. Thus the inverse document frequency increases the weight of terms that occur rarely in the document collection:

$$(tf - idf)_{i,j} = tf_{i,j} \times idf_i \tag{2.1}$$

The final feature vector is ready as input for several different classifiers like *k–Nearest Neighbor* (kNN), *Decision Trees* or *Random Forests* or *Support Vector Machines* (SVMs). Examples of document classification studies utilizing these machine learning techniques are the works by *Chen and Ho* [CH00] and *Leopold and Kindermann* [LK02].

2.3.5. Multilabel Classification & Hierarchical Taxonomy

Much research was done on the topic of labeling data with two or more categories. This was done in different research areas like document and multimedia classification. Especially in the task of topic prediction or tag assignment often a single topic or tag to describe some data is not enough. Tagging such data with multiple labels can make unusually hard classification tasks simpler and cover the topics better for searching and indexing. Another way to organize data and information is to use taxonomic hierarchies, which provide a more straightforward and understandable view on the data. Many exist on both topics separately. Interested readers can look into work from *Santos and Rodrigues* [SR09] and *Punera and Rajan* [PR09] for systems combining these two research areas for *multilabel taxonomic classification* of textual data. As already shown in Figure 2.4, multilabel, taxonomic classification and the combination of both have been already utilized for web video categorization.

2.3.6. Classification

Many different *supervised* and *unsupervised* classification methods were already used and analyzed for genre classification, while *retrieval* approaches like *Content–based Image Re-*

trieval (CBIR) become very popular for web video categorization. These, however, are beyond the scope of this thesis. Some popular choices of classifiers and examples can be found in their related work sections of *Montagnuolo and Messina* [MM07] [MM09]. They include support vector machines, artificial neural networks, multilayer perceptrons, decision trees, k–nearest neighbors, HMMs and Gaussian mixture models. Genre classification performed with hierarchical SVMs is presented by *Liu et al.* [LYCR05] and *Yuan et al.* [YLM+06]. Decision Trees, random forests and SVM will be discussed in detail in the next chapter as part of our system.

3. Methodology

In this chapter the system for TV genre classification and web video categorization is presented. The chapter is roughly divided into two parts, the *baseline system* before the thesis and the additions and improvements, which are the main part of this thesis. The system itself is in development since 2009 and was upgraded and improved gradually over the past years. But since all parts come together for the overall classification system and the improvements are better understandable in comparison, the baseline system is described in detail first. After this, the actual work of this thesis is presented in Section 3.2. Figure 3.1 gives an overview of the whole system. For easy understanding and plug–and–play purposes the system is assembled in modules, inspired from *Montagnuolo and Messina* [MM07]. Red modules are newly added to the system as part of the thesis. Blue modules are modified or improved by this work. However, this chapter focuses on the theoretical foundation and specifications of the system. Details of the implementation are presented in Chapter 4.

3.1. Baseline

The baseline system describes the genre classification system before this thesis. In the following sections the single modules are described separately and in the case of modified modules in their original state. Over the years the modules have undergone several changes, while some new modules were added, others were adapted or removed completely. To stay within the scope of this thesis the baseline system is described in the final state before this thesis. Earlier evaluation results are presented in Chapter 5 for comparison.

3.1.1. Shot Detection

The main purpose of the shot detection module before this thesis was the computation of the structural features. The shot detection module is from *Ekenel et al.* [EFG+07]. The shot detection is capable of detecting simple *cuts*, *fade–in* and *fade–outs* as well as *dissolves*, with output of all shot change types and frame numbers.

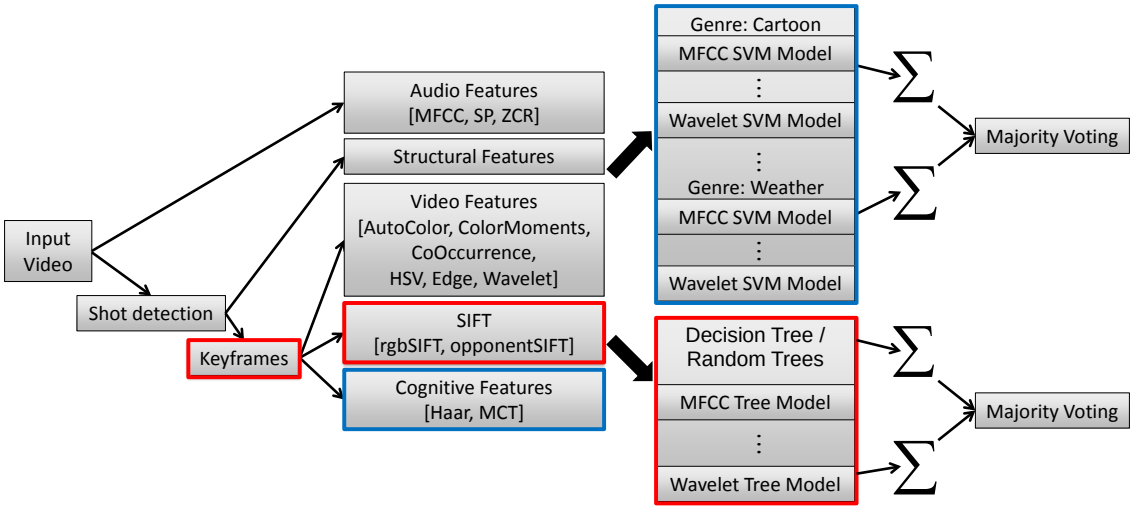


Figure 3.1.: Overview of the classification framework highlighting processing steps, modules and additions by this thesis. Blue symbolizes improved modules, red new ones.

3.1.2. Low–Level Visual Features

Six different low–level visual features, which represent both color and texture information in the video, are extracted in the visual features module. These are also the features that were utilized for content–based analysis to detect high–level features and semantic concepts in videos, as part of the TRECVID evaluation by *Ekenel et al.* [EFG+07] [EGS08].

Color Descriptors

Color features are one of the most popular visual features in the area of image retrieval, since color features are less dependent on the size, direction, and view point of images compared to other visual features. The three color features used are:

HSV histogram. Color histograms presented by *Swain and Ballard* [SB91] are applied in many image and video retrieval systems. From the HSV color space a histogram is built with 162 bins utilizing all three channels. Hue (H) values, which represent the color information are quantized more precisely, 18 bins for the ”H” channel, 3 bins for the saturation (S) channel, and 3 bins for the value (V) channel.

Color moments. The first three color moments from *Stricker and Orengo* [SO95], *mean*, *variance* and *skewness* are used. The image is divided into $k \times k$ blocks, and color moments are extracted from each image block. The final feature vector is obtained by concatenating the color moments extracted from the blocks, which results in a $9 \times k \times k$ feature vector. k is set to $k = 5$ resulting in a 225–dimensional feature vector.

Autocolorcorrelogram. The color correlogram was proposed by *Huang et al.* [HKM+97] to characterize not only the color distributions of pixels, but also the spatial correlation between pairs of colors. A color correlogram is a table indexed by color pairs, where the k-th entry for (i, j) specifies the probability of finding a pixel of color j at a

distance k from a pixel of color i in the image. Let I represent the entire set of image pixels and $Ic(i)$ represent the set of pixels whose colors are $c(i)$. If we would consider all the possible combinations of color pairs, the size of the color correlogram will be very large. Therefore, a simplified version of the feature called the color autocorrelogram is often used instead. The color autocorrelogram only captures the spatial correlation between identical colors and thus reduces the dimension to $O(Nd)$. 64 quantized color bins and five distances are used for this representation.

Texture Descriptors

Texture features are also an important group of image descriptors. Three different types of texture descriptors are used:

Co-occurrence texture. The implemented algorithm is based on the description given by *Campbell et al.* [CHE+06]. Five types of features are extracted from the gray level *co-occurrence* matrix (GLCM): *entropy*, *energy*, *contrast*, *correlation*, and *local homogeneity*. Those features are extracted from 24 different GLCMs, in our case with 8 gray level bins, at different orientations and distances. The resulting vector is $24 \times 5 = 120$–dimensional.

Wavelet texture grid. The implementation follows the description given in the work of *Campbell et al.* [CHE+06], obtaining the variances of the high–frequency sub–bands of the wavelet transform of each grid region. We used 12 sub-bands (4–level analysis). The used wavelet basis function is the simple Haar wavelet while the grid has $4 \times 4 = 16$ regions. Thus, the resulting vector is $16 \times 12 = 192$–dimensional.

Edge histogram. For the edge histogram, five filters as proposed in the *MPEG–7* standard are used to extract the kind of edge in each region of 2×2 pixels. Then, those small regions are grouped in a certain number of areas (4 rows $\times$ 4 columns in our case) and the number of edges matched by each filter (vertical, horizontal, diagonal 45°, diagonal 135° and non-directional) are counted in the region's histogram. Thus, the resulting vector is $4 \times 4 \times 5 = 80$–dimensional.

3.1.3. Audio Features

As already pointed out in Section 2.3.2 audio is very important in genre classification. Therefore, it is of paramount importance to utilize audio information in our genre classification system. Three features are computed from the audio signal of each video, whereas additional features can be integrated easily in future work. These features are *Mel Frequency Cepstral Coefficients*, *Signal Energy* and *Zero Crossing Rate*. All features are computed from a mono–channel audio signal with a 16kHz sample rate and a 256 kbit/s bit rate.

The single features are computed over small overlapping windows of $N = 400$ samples and 160 sample–shifts using the Hamming window function. In the following equations m is the index of the window and $s_a(n)$ is the signal at the time index n.

Mel Frequency Cepstral Coefficients. MFCCs are coefficients that make up the mel frequency cepstrum. In the mel scale the frequency bands are logarithmically spaced,

which approximates the human auditory system and can lead to a better representation of the signal. In a nutshell, short excerpts of the signal are Fourier transformed and the log–amplitudes of the power–spectrum are taken. These are mapped onto the mel scale and *DCT* is performed. Typically 13 coefficients are used in speech recognition, while the first five coefficients provide the best genre classification performance in the work of *Tzanetakis and Cook* [TC02]. The 8^{th} order mel frequency cepstral coefficients are computed in this system.

Signal Energy Signal energy is defined as the mean square of the amplitude in the current window:

$$SP(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} s_a(n)^2 \tag{3.1}$$

Zero Crossing Rate The ZCR measures the rate of zero crossings in the amplitude of the signal, averaged by the length of the frame and is another widely used feature in speech recognition and music information retrieval. It is easy to compute and, for example, most indicative and robust to recognize speech in audio signals as shown by *Saunders* [Sau96]:

$$ZCR(m) = \frac{1}{N} \sum_{n=m-N+1}^{m} \frac{|sign(s_a(n) - sign(s_a(n-1))|}{2} \tag{3.2}$$

Each of these features are used in three different feature vector representations:

1. A single 20–dimensional feature vector consisting of the mean and standard deviation for each feature over the whole audio signal was chosen for the first representation, inspired by the work of *Lu et al.* [LLZ01].

2. One feature vector for each feature is created separately. Mean and standard deviation of each feature (2x SP, 2x ZCR, $8 \times 2 = 16$x MFCC) are computed over a time frame of one second for the whole audio signal. These values are used as an input to compute a 10–component Gaussian mixture model. Each Gaussian in the model is represented by a mean, standard deviation and weight parameter itself. Therefore each Gaussian has 5 parameters since the mean and standard deviation of the features have both separate Gaussian mean and standard deviation values. In case of the MFCC each of the 8 coefficients have its own mean and standard deviation values which again have their own Gaussian mean and standard deviation values. This leads to 33 parameters for each Gaussian: $(8 \times 2 \times 2) + 1 = 33$. For SP and ZCR this leads to a 50–dimensional feature vector and in case of MFCC to a 330–dimensional feature vector.

3. A third representation is computed with a 10–component Gaussian mixture model, without the intermediate step of computing mean and standard deviation over one second frames of each feature. Therefore, the feature vectors are 30– and 170–dimensional consisting of only three parameters for SP and ZCR and 17 parameters for MFCC features.

3.1.4. Cognitive Features

Cognitive features are implemented as proposed by *Montagnuolo and Messina* [MM09]. A frontal–face detector based on the object–detection framework by *Viola and Jones* [VJ01] is used to detect faces represented by rectangles. It is a 22–dimensional feature vector that consists of four separate statistics about the distribution of the faces in the video. All statistics are normalized over the duration of the input video measured in frames. N_f represents the total number of faces, D the duration of the video and W and H represent the width and height of the frame:

Avg number of faces per frame. The ratio between the total number of faces and the video duration:

$$AvgF = \frac{N_f}{D} \tag{3.3}$$

Distribution of the faces per frame. A 11–bin histogram that saves the distribution of the number of faces per frame. That is, the i^{th} $(i = 0, 1, 2, \ldots, 9)$ bin represents the amount of frames that contain i faces. The amount of frames that contain more than 10 faces are represented in the 11^{th} bin.

Location of the faces in a frame. The third component is a 9–bin histogram that corresponds to the location of the faces in each frame. The frame is divided into 3×3 blocks and each bin represents the amount of faces that block contains.

Face covering percentage. The face covering percentage is the ratio between the space covered with faces and total image space. The face covered space is approximated with the detected face rectangles. w_i and h_i represent the width and height of the face rectangle:

$$FCP = \frac{100}{D \cdot H \cdot W} \sum_{i=1}^{N_f} (w_i \cdot h_i) \tag{3.4}$$

3.1.5. Structural Features

The structural feature vector is related to shot editing of the video in terms of duration and rhythm. It is composed of sub–features proposed by *Montagnuolo and Messina* [MM07] [MM09] and 15–dimensional overall:

Avg. shot length. The average shot length gives information about the average rhythm of the video and is measured in seconds. F_r is the frame rate, N_s the total number of shots and $\triangle l_i$ the individual shot length:

$$AvgShotLength = \frac{1}{F_r \cdot N_s} \sum_{i=1}^{N_s} (\triangle l_i) \tag{3.5}$$

Shot length distribution. A 9–bin normalized histogram models the shot length distribution. Bins 1 to 8 represent the range of 0 to 30 seconds uniformly. The last bin collects shots longer than 30 seconds.

Shot temporal activity. Indicates the shot distribution along the video. This feature is represented by a 5–bin histogram and a cumulative function, which uniformly collects the fraction of shots occurring up to duration time t.

3.1.6. Classification & Fusion

Classification is performed using support vector machine (SVM) classifiers [Vap95] [Bur98]. The one–vs–all strategy is employed to train a binary SVM for each feature and each genre. The radial basis function (RBF) is used as the kernel:

$$RBF : K(x_i, x_j) = \exp\left(-\gamma \parallel x_i - x_j \parallel^2\right), \gamma > 0 \tag{3.6}$$

With SVM classification it is important to determine the optimal parameters for the chosen kernel function and for the classifier to avoid unsatisfactory results. The first step in this process is to scale the whole data to a common range of $[-1, +1]$ or $[0, 1]$ depending on the feature. The RBF function parameter γ and the penalty parameter $C > 0$ of the error term need to be optimized for high performance. This step is conducted using a cross–fold validation scheme on the training data to perform training and classification with different combinations of these two parameters (for example, $C = 2^{-5}, 2^{-3}, ..., 2^{15}, \gamma = 2^{-15}, 2^{-13}, ..., 2^{3}$). Finally the parameter combination with the highest classification accuracy is used to train a model on the whole training set. This process is conducted to train all SVMs. No weighting on the training samples is performed during this process. For classification the SVMs offer the possibility of binary or probabilistic output for each testing sample, enabling different fusion techniques for classification.

All feature vectors are extracted for one video and fed multiple times to the corresponding SVM model for each genre, which produce the desired classification output. To get an overall genre prediction for a video, the single outputs have to be combined. One possibility is to accumulate all the outputs for each feature in one genre category with both the binary or probability output and perform majority voting. The genre category with the highest probability or highest number of features is chosen.

Classification is performed with k–fold cross–validation. The dataset is divided into k folds. $k - 1$ folds are dedicated for training and the left out fold for testing. Every fold is used one time for testing. Final scores are averaged over the folds.

3.2. Extensions and Improvements

This section shows the extensions and improvements to the baseline system. It is divided into general improvement on the framework, keyframe extraction, new and improved features like the SIFT and cognitive modules, and newly added and updated classification possibilities, the Decision Tree/Random Forest and SVM module.

3.2.1. Framework

Before upgrading the baseline system with concrete improvements for better classification results, the system was first transferred to an almost completely new framework. The reason was to benefit from the new library versions of OpenCV[1] and OKAPI[2] and the improvements and modifications made about them. Since this is part of the implementation, details about the changes and improvements will follow in Chapter 4.

[1] `http://opencv.willowgarage.com/wiki/`
[2] `http://cvhci.anthropomatik.kit.edu/okapi/trac/`

3.2.2. Keyframe Extraction

Before the re–implementation of the framework for this thesis, the visual, structural and cognitive features were computed on every frame of an input video. The final feature vectors were global statistics normalized over the number of frames and were not influenced by the number of processed frames. The only disadvantage proceeding this way was the high computation time on video data with long durations and high frame rates.

It is well established in the semantic concept research that single or multiple keyframes are representative enough for robust classification, decreasing the computational time by several magnitudes. This argument also applies to the newly added SIFT feature extraction. Since the SIFT features generate feature vectors for each processed frame and not one for the whole video, processing videos on all frames becomes computationally and memory wise unfeasible. Also no temporal features are used by the system.

Looking at video material, it is easy to understand that visible change to the image source happens mostly during shot alteration, excluding the fact of long scenes with a lot of camera movement. Therefore, it is reasonable to extract keyframes on a shot basis for good representation of a video. In the works of *Snoek et al.* [SWG+05] [SvdSdR+08] the MediaMill team showed that multiple keyframes help a great deal for classifying concepts. Features normalized by the duration of the video in frames will be now normalized by the number of keyframes.

The keyframe module uses the shot detection output to extract any number of keyframes for each detected shot. Choosing only one keyframe per shot, the keyframe in the middle of the shot is extracted. With more than one keyframe per shot the keyframes are distributed linearly across the shot.

A second option is added to extract any number of keyframes over the whole video without shot information. Very large videos with thousands of shots make the number of keyframes still too high for processing. A maximum number of keyframes can be chosen for extraction and if the number of shots and keyframes per shot exceeds this maximum number, the maximum number of keyframes are uniformly extracted over the whole video.

3.2.3. SIFT Descriptor

The *Scale Invariant Feature Transform* (SIFT) descriptor was introduced by *Lowe* [Low04]. It quickly became one of the most popular features for all kind of image based research. In combination with bag–of–visual–words representation presented by *Sivic and Zisserman* [SZ03], SIFT descriptors showed very promising results and improvement in state–of–the–art research.

Today many variations of the initial proposed SIFT descriptor exist, varying on the underlaying color space, different kind of local sampling point strategies and codebook assignment. An extensive overview and evaluation of different SIFT descriptors are presented by *van de Sande et. al* [vdSGS10]. SIFT descriptors in many different combinations proven to be successful in achieving state–of–the–art performance in the TRECVID evaluation in the last several years. Variations, description and usage of the SIFT descriptors can be found in work by *Snoek et al.* [SvdSdR+08] [SvdSdR+10] [SvdSL+11]. An overview of their SIFT descriptor processing pipeline is given in Figure 3.2.

SIFT descriptors belong to the category of local features, because they describe a small local region, for example a local interest point. Among other things they are *scale* and *rotation* invariant as will be clear in the following general description for generating and using SIFT descriptors presented by *Lowe* [Low04] and *van de Sande et al.* [vdSGS10]:

1. **Scale–invariant keypoint detection.** To compute SIFT descriptors first local interest points have to be detected. Multiple techniques can be applied. Two of them are the *Harris–Laplace* point detector or the *dense sampling* method. While the Harris–Laplace detector actually detects interest points in the image (Figure 3.2(b)), the image is partitioned into small slightly overlapping interest points (Figure 3.2(a)) with dense sampling.

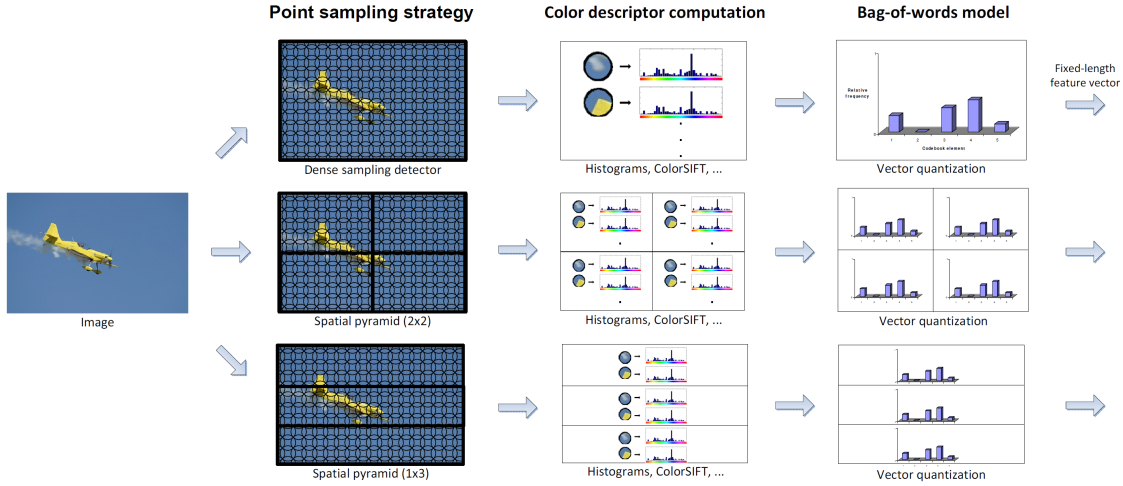


Figure 3.2.: (a) SIFT processing pipeline with dense sampling and spatial–pyramid strategy (b) SIFT processing pipeline with Harris–Laplace interest point detector. Taken from [vdSGS10].

2. **Assigning orientation to the keypoints.** Based on local image gradient directions, orientations are assigned to each keypoint. A 36–bin histogram for 360° is used while the orientation with the top magnitude as well as magnitudes up to 80% of the top magnitude influence the overall orientation of the keypoint.

3. **SIFT descriptor generation.** The local region around the keypoint is divided into 16×16 sample arrays, in which gradient magnitudes and orientations are calculated. A Gaussian window is used to weight the gradient magnitudes. The size is set by the scale of the keypoint. These orientations are saved in a histogram of 8–bins over 4×4 regions, see Figure 3.3. The final descriptor is rotation invariant, because the

keypoint orientation is subtracted from the descriptor. The final representation is normalized and 128–dimensional ($4 \times 4 \times 8 = 128$).

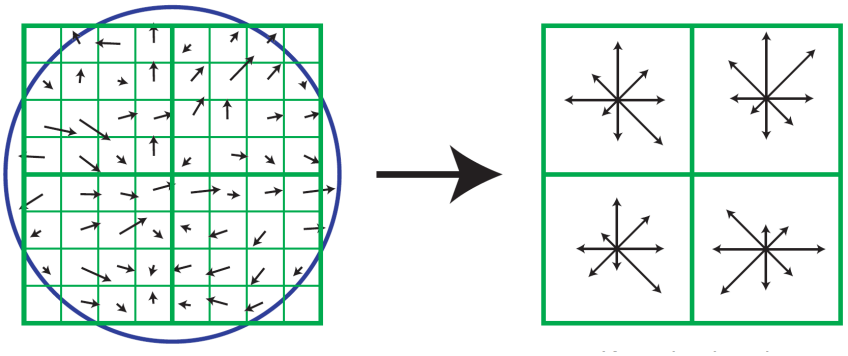


Figure 3.3.: Keypoint descriptor creation displaying the small sample arrays on the left and the sub–region orientations and magnitudes on the right. The circle symbolizes the weighting of the magnitudes of the sample arrays with a Gaussian window. Taken from [Low04].

4. **Bag–of–visual–words model.** One way to use SIFT descriptors is to generate bag–of–words histograms using a codebook of SIFT descriptor clusters like *Yang et al.* [YJHN07], representing an image as an histogram of codewords. Visual–bag–of-words were introduced in 2003 by *Sivic and Zisserman* [SZ03]. Codebook clusters are computed using the *k–means* clustering algorithm with 500, 1,000 or more clusters. The final histogram is normalized over the number of codewords.

5. **Classification.** The final feature vector representation is fixed in length for each image and can be used as input for supervised machine learning algorithms like, for example, SVMs.

Many parameters influence the overall SIFT descriptor generation process. Possible options and parameter choices as part of this thesis are presented next:

Keypoint detector. The focus of this thesis lies on the dense sampling detector. Possible Harris–Laplace interest point detection could be investigated in the future.

SIFT descriptor. *van de Sande et al.* [vdSGS10] evaluated many different color and SIFT descriptors. *OpponentSIFT* and *RGB–SIFT* achieved the best results. OpponentSIFT and RGB–SIFT descriptors are normal SIFT descriptors computed in the respective color space channels and concatenated into a single feature vector ($128 \times 3 = 384$). The original SIFT descriptor is computed from gray images. SIFT, OpponentSIFT and RGB–SIFT are the choices for this genre classification system.

Visual Codebooks. Images will be represented as codeword histograms using the bag–of–words model. Codebooks will be generated from keyframes available for the respective datasets used in the evaluation. Details will be presented in Chapter 5.

Codebook Assignment. Typically codewords are assigned to one cluster only (*hard assignment*). The distance to the cluster centers is measured with the Euclidean distance. *van Gemert et al.* [vGVSG10] proposed *soft codebook assignment* of code-

words to more than one cluster. The focus in this thesis will be on hard assignment of codewords.

Spatial Pyramid. For natural scene classification *Lazebnik et. al.* [LSP06] proposed *spatial pyramid* image partitioning. As shown in Figure 3.2(a) an image can be divided into different regions where codeword histograms are computed separately and concatenated in the end. This method is also applied in the works of *Snoek et al.* For this thesis image partitioning into 1×1, 2×2 and 1×3 segments will be included.

The RGB color space is a three channel color space based on the RGB color model. It can produce any color by combining the three additive base colors *red*, *green* and *blue*. In the gray color space only one channel is representing the intensity information of each pixel. It is computed in the following manner from the RGB color space:

$$Grey = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \tag{3.7}$$

The opponent color space has three channels. The intensity channel O_3 and the color information channels O_1 and O_2. Transformation from RGB to opponent color space is as follows:

$$O_1 = \frac{R - G}{\sqrt{2}} \tag{3.8}$$

$$O_2 = \frac{R + G - 2B}{\sqrt{6}} \tag{3.9}$$

$$O_3 = \frac{R + G + B}{\sqrt{3}} \tag{3.10}$$

The two main functions of the SIFT module will be described shortly one final time for a better overview:

Codebook generation. Provided input images (keyframes), a codebook can be generated extracting SIFT descriptors of every image and performing k–means clustering. Different kind of SIFT descriptors from the options presented above can be used to generate different codebooks.

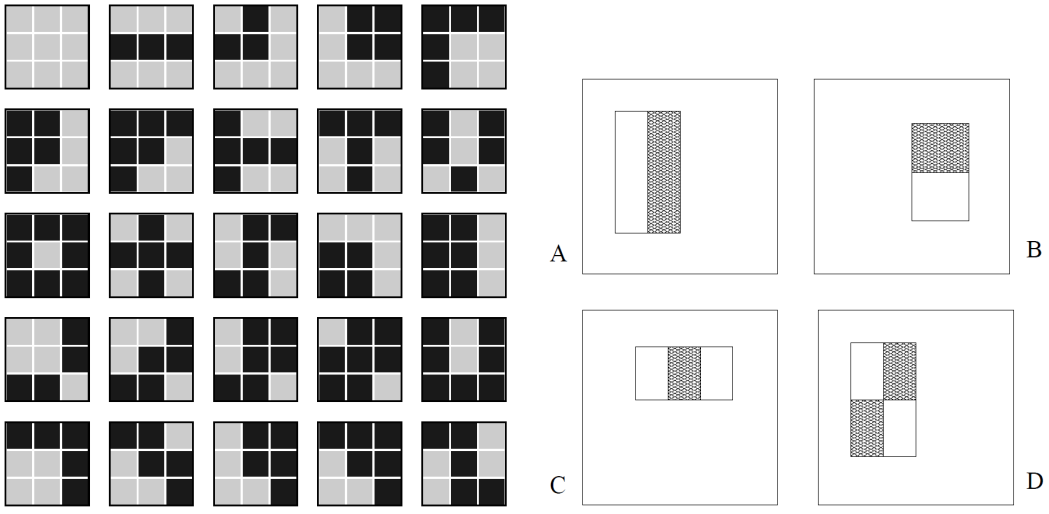
SIFT feature generation. Feature computation with the bag–of–words model requires a codebook for codeword histogram computation. For each keyframe different kind of SIFT descriptors are computed and a codeword histogram is generated representing this image using the corresponding codebook. This means that a video represented by 10 keyframes will produce 10 feature vectors, one for each frame.

3.2.4. Face Detection

The baseline system provided cognitive features based on frontal face detection described by *Viola and Jones* [VJ01]. As will be shown in the evaluation in Chapter 5, these detections include many false positives and false negatives. To analyze the usefulness of the used cognitive features, a second face detection method is added to compare the

face detection and cognitive feature results. Face detection using the *Modified Census Transform* from *Fröba and Ernst* [FE04] is added as an option. Past evaluations also showed that a lot of faces were falsely classified as negative due to slide rotation on the vertical axis. A second option to include detections of side and profile views of faces is also added. Details of the implementation will be discussed in Chapter 4.

Haar–like Face Detection. The ground breaking system by *Viola and Jones* [VJ01] has four main parts that make it so successful. The *Haar like* feature shapes as shown in Figure 3.4(b), the efficient computation of these features with the *integral image*, *AdaBoost* for feature selection and *weak classifier* training and the *classifier cascade* as shown in Figure 3.5 for classification. The training and classification is applied to 24×24 pixel sub–windows on multiple scales of the image.



(a) MCT Local Structure Features (b) Viola & Jones Rectangle Features

Figure 3.4.: (a) [FE04], (b) [VJ01].

The Haar like features describe the image areas over which the pixel intensities are summed up respectively. Afterwards the sums of the white and shaded areas are subtracted from each other. One reason for the very fast computation of these values and the real–time capabilities of the system is the computation of these areas with the integral image algorithm. The integral image is a transformation of the image, where each 'pixel' saves the value of the sum of pixel intensities up to its upper–left corner. This algorithm makes it possible to calculate the sums of rectangular areas very fast. It only references the corners of the rectangles since the integral image already provides the sum of the intensity up to this pixel. To calculate the pixel sum of one rectangle it takes only four references. Applying this method to the feature shapes of two, three or four rectangles, only six, eight or nine references are needed.

The second reason for the very robust, efficientl and fast system is the classifier cascade structure. Each sub window is fed to the cascade for classification. At each stage of the cascade the sub–image is classified into face or background. If the sub–image is classified as background, it is discarded at once, if it is classified as a

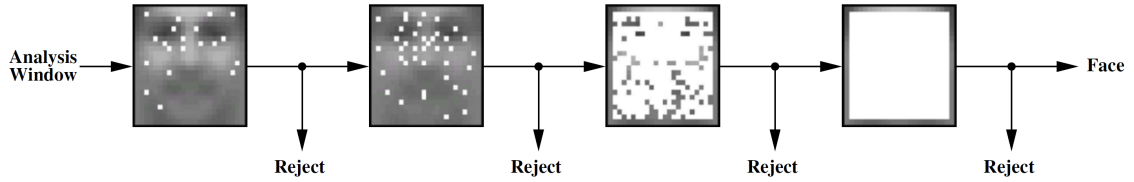


Figure 3.5.: Classifier cascade example as in [FE04].

potential face detection, it is passed to the next stage. Only if it reaches the end of the cascade, it is classified as a face detection. The benefit of this structure lies in the complexity of each cascade stage. Beginning with a very small feature selection of two features at the beginning and getting more complex with each stage, it is possible to discard up to 60% of background sub–windows while keeping 100% of potential face detections in the first stage very fast. The following stages discard even more background sub–windows while constantly becoming more complex.

MCT Face Detection. The system by *Fröba and Ernst* [FE04] is very similar in design to the work of *Viola and Jones* [VJ01]. It distinguishes itself only in a few minor points. Local structures are chosen as features. These structures are calculated on 3×3 pixel neighborhoods using a modified version of the census transform [FE04]. A subset of all possible structures ($2^9 - 1 = 511$) is shown in Figure 3.4(a). The census transform is represented by a bit string, in this case eight bits, saving information if the surrounding pixels have higher or lower intensity values than the center pixel. In the modified version all the pixel values including the center pixel are compared to the mean intensity of the 3×3 pixel neighborhood.

The training and classification of the system is also carried out with a classifier cascade of four stages. Classifiers at each stage are trained with a variation of the *boosting* algorithm. Like in the work of *Viola and Jones* [VJ01] background sub–windows are discarded at each stage, beginning with a very efficient and fast detecting first–stage classifier. The four stages are very powerful, discarding up to 99% of background images at the first stage, while keeping almost all the possible face detections.

3.2.5. SVM Module

The SVM module of the baseline system was already fully functioning but was lacking adjustment options like training subset generation, weighting of the training classes samples and weighting of the classification predictions. The first two options are important for the classification using SIFT features as will become evident in Chapters 4 and 5. Also for classification of SIFT features the fusion of multiple feature vectors per video is needed.

Basically, it is still the same classification process as described in Section 3.1.6. For overview purposes the whole new process will be described in detail:

K–Fold generation. K–fold cross validation is performed for all experiments. Videos in each genre are sorted according to the number of keyframes, so large videos with many SIFT feature vectors are evenly distributed over all k–folds. Through this,

the number of positive and negative samples is balanced along the folds. This is important for the balanced number of positive samples for each fold.

Scaling of the data. After k–fold generation the training and testing data is scaled as described in Section 3.1.6. Here it is possible to generate subsets of training data to balance the amount of positive and negative samples and speed–up the grid search and model training time. With a high number of genres and varying video durations, positive and negative samples become quickly highly unbalanced. Subsets are generated taking all positive keyframes from a training file and picking a number of randomly picked negative keyframes. Possible positive and negative sample ratios may be 1:1, 1:2 or 1:3. Important is the fact to pick the same random negative keyframes for all kind of features, especially SIFT features with many samples per video. Therefore random samples are chosen once for one feature and then the same samples are used for all the other features.

Training. The individual models are trained the same way as in the baseline system. Now it is possible to adjust weights to the positive and negative samples to cover imbalance across the sample data. As proposed by *van de Sande et al.* [vdSGS10], the positive and negative samples can be weighted as follows:

$$Positive\ weight = \frac{\#positive}{\#positive + \#negative} \tag{3.11}$$

$$Negative\ weight = \frac{\#negative}{\#positive + \#negative} \tag{3.12}$$

Classification. Prediction of the genre tags for each feature model in each genre has not changed and is as described in Section 3.1.6 of the baseline system.

Fusion. All features can be distinguished into two groups. One group describes "global" features that represent the input video through one feature vector. The other group of features are the SIFT features. One feature vector for each keyframe of the input video is extracted. The fusion of these "global" and SIFT features has to be managed. This is achieved by averaging the SVM–SIFT predictions over the number of frames for each SIFT feature for each genre and treating SIFT features as any other "global" feature. New to the fusion process is the option to weight the predictions of the SVM models using weights for the individual features. These feature weights are the same for all genres. Adjusting weights also to genres could be evaluated in future work.

3.2.6. Decision Tree/Random Forest Classification

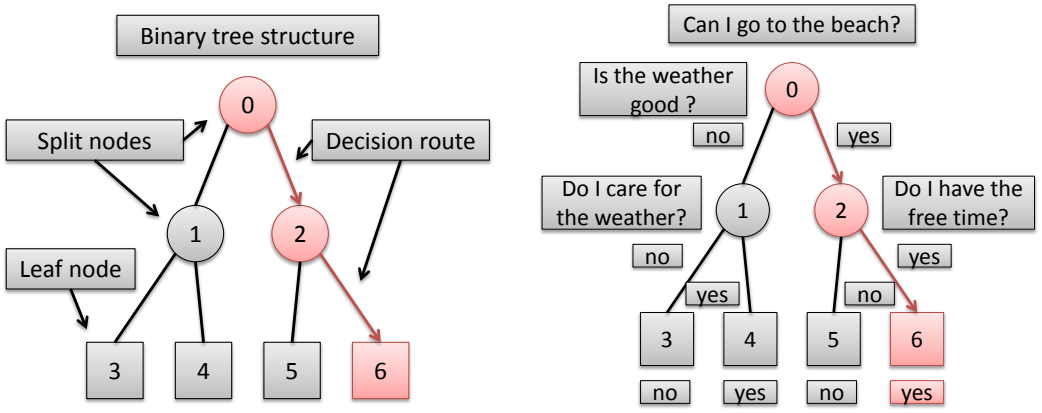
In the early versions of our system we experimented with Multilayer Perceptrons like *Montagnuolo and Messina* [MM07] and switched to SVMs achieving better classification results (presented in Table 5.4). In this thesis the evaluation of *Decision Tree* and *Random Forest* classifiers will be analyzed as part of our system. Evaluation will be performed both on old and new datasets for comparison.

Decision Tree classification dates back to the 1980s, where *Breiman et al.* [BFOS84] introduced *Classification and Regression Trees*. For many years the *C4.5* algorithm by *Quinlan* [Qui93] for improving decision tree training was very popular until *Random Decision*

Forests were introduced by *Tin Kam Ho* [Ho95] in 1995. Many researches have presented better classification results using random forests over C4.5 Decision Trees. Today *Decision Trees* and *Random Forests* are still very popular choices for supervised classifiers and quite recently contributed to one of the biggest success stories, the development of the *Microsoft Kinect*[3].

For an up–to–date extensive look and introduction to *Decision Trees* and *Random Forests* the reader is encouraged to read [CSK11] by *Criminisi et. al.* The work offers extensive related work overview, *Decision Tree* and *Random Forest* mode of operation and examples for different kind of tasks like *classification*, *regression*, *density estimation* and *manifold learning*. Furthermore, it provides an in depth evaluation of the most important parameters and their influence on classifier training on all tasks. The following introduction to *Decision Trees* and *Random Forests* is inspired by their work.

Decision Tree. A basic type of a binary tree, where each internal node (including the root node) has two outgoing edges, as shown in Figure 3.6(a). Trees are shapes of nodes and edges, containing no loops and each node has only one incoming edge. Like the name is suggesting the *Decision Tree* is used to make a decision about a specific problem. Starting at the root node the problem is propagated through the nodes until it arrives at a leaf node were the result decision is predicted. At each internal node the problem gets analyzed by specific rules and propagated to the left or the right edge depending on the result.



(a) Decision Tree (b) Decision Example

Figure 3.6.: (a) General shape of a binary Decision Tree. Internal nodes and root node are split nodes. Leaf nodes are terminal nodes with decision output. Example path of classification route is marked red. (b) Decision Tree example with results.

For better understanding a decision tree can be viewed as an hierarchical problem solver that divides one problem into several small problems, which are easier to solve at each internal node. An example is given in Figure 3.6(b). The main issue in decision tree building (training) is the establishment of the split node decision functions and the predictions in the leaf nodes through training samples.

[3]Microsoft Corp. Redmond WA. Kinect for XBox 360.

In mathematical problems and notations the input for a classification can be a numerical feature vector and the output a numerical value (class label). Training is performed optimizing the parameters of the split functions in the split nodes using the training feature vectors. Propagating the training data, the data gets separated at each internal node until a minimum number is reached or class prediction is possible with a certain amount of confidence. For example *information gain*, IG, can be used as a measure for optimal node parameters:

$$IG = H(S) - \sum_{i \in \{1,2\}} \frac{|S^i|}{|S|} H(S^i) \tag{3.13}$$

$H(S)$ can be the *Shannon entropy* for example, defined as $H(S) = -\sum_{c \in C} p(c)log(p(c))$. S denotes the samples reaching this node, S^1 and S^2 are the samples leaving the node on one of the two edges. Different training stopping criteria can be chosen, for example when the tree reaches a maximum number of levels, a minimum information gain is reached or a specific minimum of training samples is reached after a split at an internal node.

Random Forests. Random Decision Forests are an ensemble of Decision Trees. The forest is called random because each tree is trained randomly with a specific value of randomness. This approach affects the randomness and difference of the single trees and has proven that different trees in a forest improve generalization capabilities. Randomness can be achieved by splitting the training data across trees and/or randomized node parameter optimization, where a smaller subset of all parameter possibilities is available in training for a specific tree. New parameters for Random Forest training and termination are the amount of randomness and the number of trees in the forest among others. For classification each test sample is pushed through all the trees in the forest. All tree predictions are collected for one overall forest prediction. Majority voting is performed over the returned class labels to predict the final class prediction.

Our system is extended with a Decision Tree/Random Forest module, similar in function of the SVM module. Instead of binary classification the multiclass classification is utilized, reducing the training time and number of classifiers by the magnitude of genres to classify. As proven by *Criminisi et al.* [CSK11] optimal parameter selection is key for good classification results. These details are described in Chapters 4 and 5.

4. Implementation

This chapter presents implementation details of the genre classification system described in Chapter 3. It lists the used programming languages and libraries as well as their possible parameter choices and the options chosen in our framework. Parameters and their values chosen for the evaluation are listed in Chapter 5.

4.1. Framework Design

The system can be divided into four parts. A system overview is shown in Figure 3.1.

C++ Framework. The main part of the whole system is written in *C++* and can be viewed as a classification framework. Two libraries and one external binary are used with this framework. The two libraries are the *OpenCV* library and the *OKAPI* library. The external binary used is the *colorDescriptor Software*[1] for SIFT descriptor computation made available by *van de Sande* [vdSGS10].

The main modules of the C++ Framework are the keyframe extraction, visual, cognitive, structural and SIFT feature extraction, and the decision tree/random forest classification module.

OpenCV. The OpenCV library [Bra00] is the most well known computer vision library. Today it features a large number of functions like image processing, gui and media i/o, object detection (cascade classifiers), feature detection (keypoints detection and SIFT descriptors) and machine learning algorithms (Bayes, kNN, SVM, decision trees/random forest, boosting, ANN and EM).

OKAPI. OKAPI (Open Karlsruhe Library for Processing Images) [Oka12] is a C++ library from the *Institute of Anthropomatics* at the *KIT* for image processing and has been designed to be used in conjunction with OpenCV. It offers very useful utility and system functions and more important implementations of image and video i/o, camera interfaces, object detectors (MCT face detection),

[1] `http://koen.me/research/colordescriptors/`

features (DCT, Gabor, PCA) and classifiers (SVM). OKAPI uses the CMake[2] build system and is operational on Windows, Linux and MacOS.

C++ shot boundary. The shot boundary module is separate, written in C++ and also built with OKAPI. It produces *.xml* shot detection files for each input video (see Appendix A).

MATLAB audio module. For fast and easy audio analysis *MATLAB*[3] [MAT10] and the *VOICEBOX*[4] Toolkit were chosen. They provide easy and powerful tools for audio analysis like audio signal processing, window functions, features like MFCCs and GMM modeling.

Python SVM classification. *Python*[5] [Pyt11] was chosen for SVM classification because of the provided scripts and tools by the *libSVM*[6] library [CL11]. The libSVM software comes with SVM training, prediction and scaling binaries, and also subset creation and grid search python scripts.

4.2. Configuration

The main framework of the genre system uses a *.xml* configuration file, where all important parameters can be set before running the system. An example of a possible configuration can be found in Appendix B. It is roughly divided into five parts. The following detailed look also explains the functionality of the framework. A framework folder structure is shown in Appendix C:

4.2.1. Main module

Source. The main C++ framework part works only on the extracted keyframes of each video. The possibility to set the source to *video* enables the extraction of the keyframes first. All other operations are performed on *keyframes*.

Keyframes. For keyframe extraction the number of keyframes per shot, the maximum number of keyframes per video and the extraction type can be chosen. The keyframes can be extracted in three different types. The default way is to specify a number of keyframes to be extracted for each shot. For long videos the number of frames may be too high for processing, therefore two options were added to limit the keyframe extraction to a specified number of frames overall. Either the maximum number of keyframes can be uniformly extracted over the whole video or a combination of the first and second option can be used. In this case the keyframes are extracted for each shot and only overall, if the maximum number of allowed keyframes is exceeded by $number\ of\ shots \times number\ of\ keyframes\ per\ shot$.

Extraction. Each feature category (visual, cognitive, structural, SIFT) can be enabled or disabled. While visual and cognitive feature extraction does not need any further parameters, SIFT and face detection modules have separate options.

[2] `http://www.cmake.org/`
[3] `http://www.mathworks.de/products/matlab/`
[4] `http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html`
[5] `http://www.python.org/`
[6] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Face Detection. The parameters for MCT or Haar cascade face detection are the same. Each type can be enabled or disabled. For each type frontal and/or side view face detection can be chosen. For Haar cascade face detection profile view cascades are available, for MCT face detection $\pm 45^o$ side view models are already pre–trained. Paths for the trained classifier models have to be specified. For evaluation the option to save images with face detection rectangles of all enabled face detectors can be activated (see Appendix. D).

SIFT. The SIFT options can be divided into SIFT codebook generation and SIFT feature extraction. The SIFT feature extraction options directly influence the codebook generation.

The codebook generation can be enabled or disabled. The desired number of clusters (codewords) and the number of input samples can be set. The k–means algorithm can be modified with different parameters like the choice of iterations and cluster center initialization types.

The SIFT feature extraction can be enabled or disabled. The keypoint detector strategy can be selected between Harris–Laplace or dense sampling, though Harris–Laplace is not supported yet. Further options for dense sampling are *pixel space* and *scale*. The three descriptors SIFT, OpponentSIFT and RGB–SIFT are available for choosing. And finally the feature extraction with spatial pyramid support can be selected. The possibilities are 1×1, 1×1–2×2 and 1×1–2×2–1×3. For example the third option means that features are extracted for all three spatial pyramid types at once. These parameters come from the colorDescriptors binary. The whole functionality of this software is described here[7].

Decision Trees/Random Forests. The classification with decision trees and random forests is heavily linked to the SVM classification module, the generated output and folder structure. The input for the tree classification are the scaled input files of the SVM module. The tree and forest parameters from OpenCV are hardcoded into the system and will be available for choosing in the configuration in the future. Following options are available for the decision tree/random forest classification in the framework configuration file. The module can be activated or deactivated. It is possible to use decision trees and random forests at the same time or separately. The user can decide to train models for the selected classifiers or use already trained models for class prediction. For the difference in training time of 'global' features compared to SIFT features, which take a lot more time, the option to select either set of features or the combination was added. Individual feature prediction has to be done manually. The module was designed to support k–fold cross evaluation. K–fold data is created with the already existing SVM classification script. Therefore the number of k–folds used for the SVM data creation has to be specified. The dataset for processing has to be chosen, since the number of genres and the genres themselves are not automatically recognized. Classification is performed in multiclass fashion with class label outputs. The trees can also be used as binary classifiers with probability output but this option is not implemented, yet.

[7] `http://koen.me/research/colordescriptors/readme`

The OpenCV implementation of decision trees and random forest includes many parameters. They influence the training stage and classification quality of the classifiers and are specifically chosen for the evaluation in Chapter 5:

- Decision Tree:
  - Max depth: The maximum possible depth of the tree. Can be smaller if other termination criteria are met or the tree is pruned.
  - Min sample count: The nodes will not be split if the number of samples reach this minimum at a node.
  - Regression accuracy: Termination criteria for regression trees. Not important for classification and set to *null*.
  - Use surrogate splits: If *true* then surrogate splits will be build. These splits are important to compute variable importance correctly.
  - Max number of categories: Cluster possible values of a categorical variables into $K \leq max\_categories$ clusters to find a suboptimal split.
  - Number of cross–validation folds: If ≥ 1 then the tree is pruned with k–fold cross–validation.
  - 1SE rule (smaller tree): If *true* then pruning will be harsher, which makes the tree more compact, more resistant to training data noise but also less accurate.
  - Truncate tree branches: If *true* the pruned branches are physically removed from the tree.
  - Priors: The array of a priori class probabilities, sorted by the class label value (not used).
- Random Forest:
  - Max depth: The maximum possible depth of the tree. Can be smaller if other termination criteria are met or the tree is pruned.
  - Min sample count: The nodes will not be split if the number of samples reach this minimum at a node.
  - Regression accuracy: Termination criteria for regression trees. Not important for classification and set to *null*.
  - Use surrogate splits: If *true* then surrogate splits will be build. These splits are important to compute variable importance correctly.
  - Max number of categories: Cluster possible values of a categorical variable into $K \leq max_categories$ clusters to find a suboptimal split.
  - Priors: The array of a priori class probabilities, sorted by the class label value (not used).
  - Variable importance: If *true* then variable importance will be calculated.

- Nactive Vars: The size of the randomly selected subset of features at each tree node and that are used to find the best split(s). If you set it to 0 then the size will be set to the square root of the total number of features.
- Max number of trees: The maximum number of trees in the forest.
- Forest accuracy: Sufficient forest accuracy for termination criteria.
- Termination criteria:
  * Terminate learning by reaching the maximum number of trees in the forest.
  * Terminate learning when the sufficient forest accuracy reached.
  * Use of both criteria depending on which is met first.

4.2.2. Audio Feature Extraction

As already mentioned the audio feature extraction is implemented in MATLAB utilizing the VOICEBOX Toolkit. The feature extraction is performed on *.wav* files extracted from the video. Their specifics are described in Section 3.1.3. The audio feature extraction itself provides some parameters and options. First the desired features and feature representations can be selected. It is possible to compute one feature in one representation, all features with all representations at the same time or any user–defined combination. The window function (Hamming or Rectangle), the size of the window (number of samples) and the shift size (in samples) are also available options.

4.2.3. SVM Classification

The SVM classification module consists of several scripts and the SVM training, testing and scaling binaries. The different scripts are described next:

py_create_SVM_input.py The extracted features of each video have to be first accumulated together and transformed into a specific format the libSVM software supports. With this process several statistics are collected that are necessary for classification. Part of these statistics are the number of videos for each genre and the groundtruth for each video. For SIFT classification the groundtruth and number of frames for each video are saved as well. Since SVM classification is performed as binary classification, files and groundtruths for each available genre are created, too.

py_subset.py The script can be used to generate subsets of training data like previously explained. The script provided by the libSVM library was modified for the subset options described earlier, namely, selecting all positive samples and a random but repeatable relative number of negative samples (see Section 3.2.5).

py_grid.py The script provided by libSVM performs the grid parameter search for the SVM model training. Parameters like class weights can be passed. The grid search is performed with different parameter combinations and on a predefined number of k–folds. These values, the k–fold number and parameter ranges of the RBF kernel parameters can be selected. Default values are 5–fold and 110 parameter combinations.

py_tvgenre.py This is the global SVM classification script utilizing all the other scripts and performing all of the SVM classification duties like, k–fold creation, scaling, training, predicting and results fusion. The separate steps can be performed together or one by one to avoid re–doing unnecessary steps again to save computational time, which is very important especially for model training. It produces overall classification accuracy, single feature accuracies, a genre confusion matrix and much more detailed information, if needed. Most important are the number of desired k–folds and the dataset to use. All features can be enabled individually for specific model training or class prediction and fusion. Manual weights can be assigned to them. As described in Section 3.2.5, class weights for grid search and model training can be used. For prediction the SVM output can be *probability* or *binary* values. They can be fused in majority voting or max rule fashion. It is also possible to fuse results from trained models, from subsets of training data and models trained on all training data. The subset option is mainly used to reduce the long SVM grid search and model training time. Therefore, 'global' features can be predicted with normal SVM models, while SIFT features get predicted with SVM models trained on subsets.

5. Evaluation

To evaluate the new genre classification system in both the TV and web domain and to be able to compare the results to the old baseline system, two old TV and one new YouTube dataset will be used. Old results can be compared on the TV datasets and the more interesting evaluation of web video domain material and categories will be carried out on the most recent taxonomy of YouTube categories. The organization of this chapter is as follows. First the datasets will be described in Section 5.1 in detail, followed by a precise specification of the system's parameter choices for this evaluation in Section 5.2. Then,

Table 5.1.: Number of genre videos and durations in the datasets.

	RAI		Quaero 2010		YouTube		
Genre	#	hh:mm	#	hh:mm	Category	#	hh:mm
Cartoon	27	07:13	3	01:0	Activism	50	12:39
Commercial	58	03:04	126	05:20	Animals	50	07:56
Documentary	-	-	12	04:55	Autos	50	13:17
Football	22	17:41	-	-	Comedy	50	04:31
Magazine	-	-	27	12:07	Education	50	15:09
Movie	-	-	30	17:51	Entertainment	50	04:06
Music Show	7	00:36	-	-	Film	50	07:01
News	49	17:19	18	07:03	Games	50	07:16
Show Games	-	-	23	09:20	Howto	50	12:01
Talk Show	38	21:06	6	09:12	News	50	09:32
Traffic For.	-	-	20	00.35	People	50	03:39
Weather For.	60	01:52	37	01:38	Science	50	05:41
-	-	-	-	-	Sports	50	05:11
-	-	-	-	-	Travel	50	05:54
Total duration	-	69:27	-	69:07	-	-	113:59
Total #	261	-	302	-	-	700	-

the results for each dataset will be presented in Section 5.3.

5.1. Datasets

The three datasets contain different genres, different number of genres and different number of videos per genre. The two TV datasets, RAI (Italian TV) as used by *Montagnuolo and Messina* [MM07] [MM09] and Quaero (French TV) have overlapping genres and almost the same number of videos and same overall duration. The web video domain dataset from YouTube has completely different categories from the TV domain and overall twice as much videos and double the amount of overall duration. This information is presented in Table 5.1.

5.1.1. Italian TV broadcast

The RAI dataset consists of 261 videos from 7 genres collected from three Italian TV channels, *RAI1–RAI3*. These genres are *cartoon*, *commercial*, *football*, *music*, *news*, *talk show* and *weather forecast*. The number of individual videos per genre as well as their duration can be found in Table 5.1. The average cartoon runtime is about 16 minutes. Commercials and weather forecast videos are the shortest in average with 2 minutes, and football and talk show, the categories with the longest videos, are 46 and 33 minutes in average, respectively. An example frame of each genre is shown in Figure 5.1. The variation within the genres is very little compared to the YouTube dataset as will be shown in Section 5.1.3. This is the same dataset as used by *Montagnuolo and Messina* [MM07] [MM09] on their state–of–the–art system and is one of the largest collections of TV broadcasting. It was the very first dataset containing entire programs, instead of having short clips from them. It also enabled us to compare results with their system (see Table. 5.4).

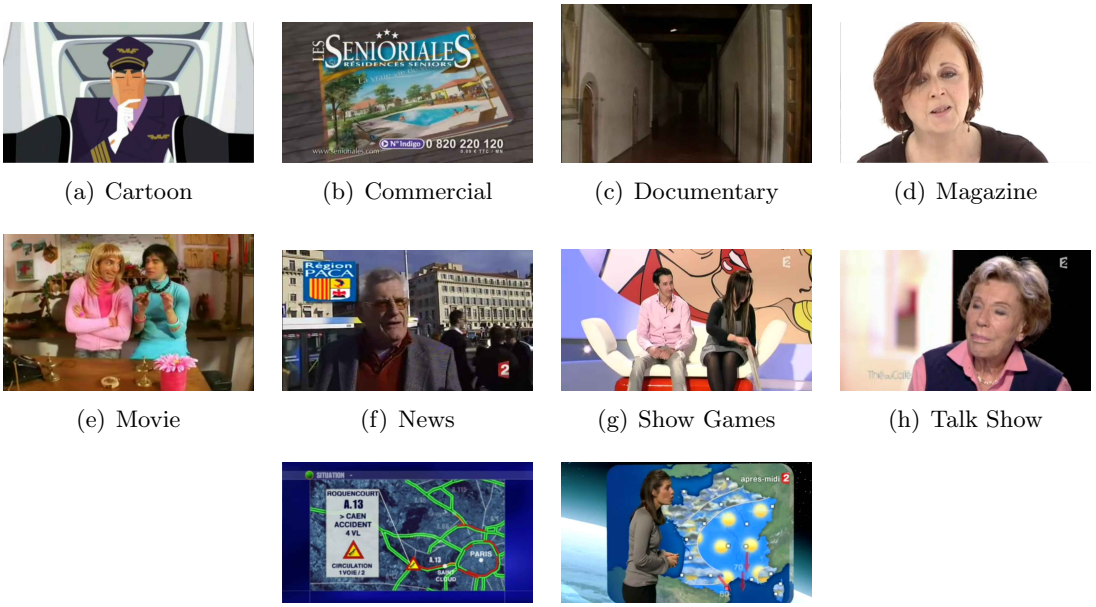
(a) Cartoon (b) Commercial (c) Football (d) Music

(e) News (f) Talkshow (g) Weather

Figure 5.1.: Sample frames from the RAI dataset

5.1.2. French TV broadcast

The Quaero dataset is a week of broadcasting from one French TV channel. The 10 genres are: *cartoons*, *commercials*, *documentary*, *magazine*, *movie*, *news*, *show games*, *talk show*, *traffic forecast* and *weather forecast*. The genre sample frames can be found in Figure 5.2. Compared to the RAI dataset, the Quaero data offers a more complete view on one TV channel and its program from one week of broadcasting. This dataset was also used in earlier evaluations and results will be compared to the new system as well in Section 5.3.2.



(a) Cartoon (b) Commercial (c) Documentary (d) Magazine

(e) Movie (f) News (g) Show Games (h) Talk Show

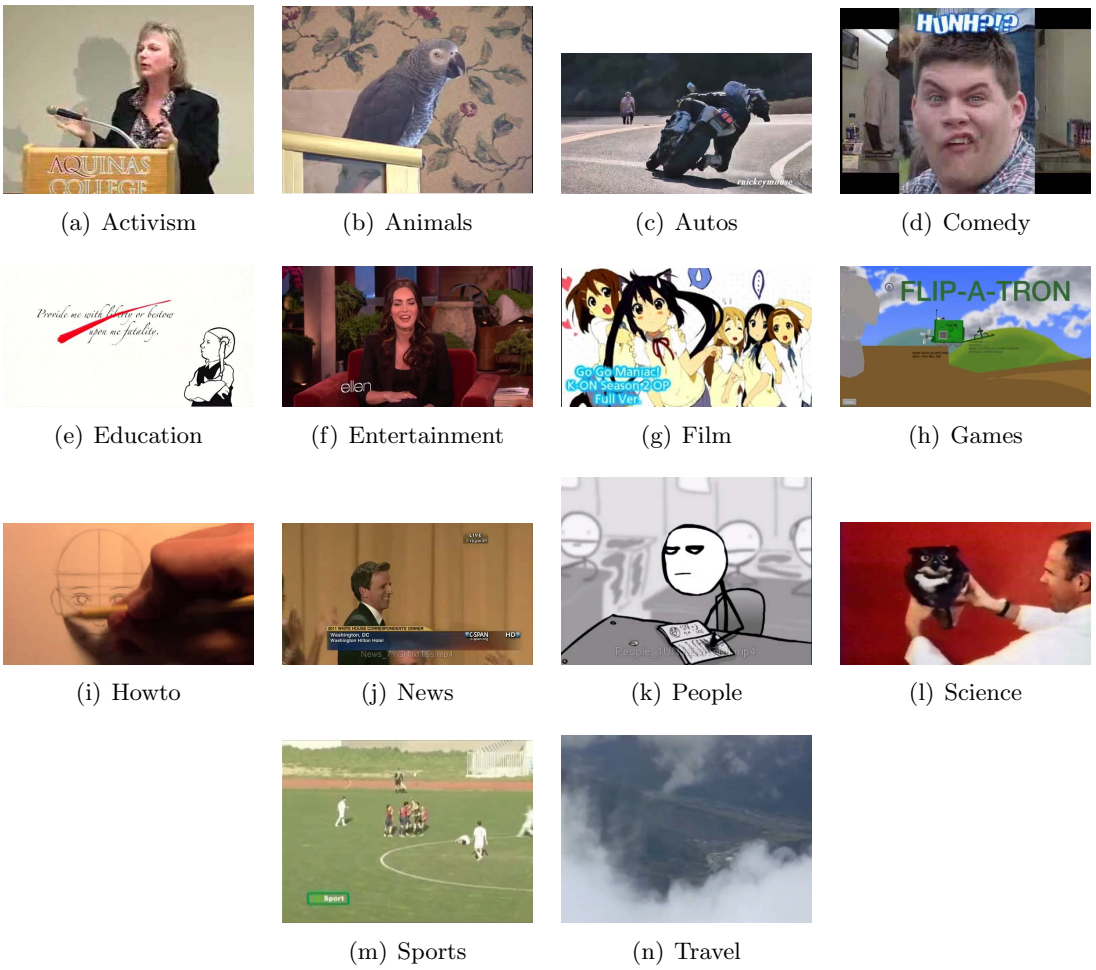
(i) Traffic Forecast (j) Weather Forecast

Figure 5.2.: Sample frames from the Quaero 2010 evaluation dataset

5.1.3. YouTube

To evaluate the genre classification system on a collection of web videos, we benefit from YouTube. Videos were collected from all the 14 top–level categories YouTube offers on its categories page[1] early 2012. These categories are *activism*, *animals*, *autos*, *comedy*, *education*, *entertainment*, *film*, *games*, *howto*, *news*, *people*, *science*, *sports* and *travel*. The difference between TV genres and web video categories is distinctive. The only categories overlapping are news and sports. The other categories have more resemblance to *topics* and not so much with an underlying genre. The difference between topics and genres is that a topic or category in this case, like *autos*, can contain all kind of videos not bound to any genre styles. Videos can be about people talking about cars, showing races or auto shops and car engines. This shows the difficulty of web video categorization and the difference to TV genre classification. Sample pictures of all categories are given in Figure 5.3. A larger selection for each category is presented in Appendix E. Even for humans some samples prove to be difficult to assign the correct category tag to.

[1] `http://www.youtube.com/videos?feature=mh`



(a) Activism (b) Animals (c) Autos (d) Comedy

(e) Education (f) Entertainment (g) Film (h) Games

(i) Howto (j) News (k) People (l) Science

(m) Sports (n) Travel

Figure 5.3.: Sample frames from the YouTube evaluation dataset. More samples for diversity comparison in Appendix E

The dataset was collected downloading 50 clips of each category using the YouTube API[2]. Searching was performed with an empty search tag and the search category set for each individual category. This process was repeated several times to eliminate very small and very long clips, with durations over two hours and under 3 seconds. A total of 700 videos with a duration of 114 hours was collected (see. Table 5.1).

YouTube changed its categories over the years. This leads to unreliable metadata information coming with the videos, since they were categorized in a category that no longer exists. An example for this is music videos which have the category tag *music* assigned to them but are found using the *entertainment* category tag for search. Therefore, the groundtruth is chosen as the category search term in the YouTube API for the respective search results. This approach ensures a realistic view on the web video classification problem with already chosen category labels from YouTube.

[2] `https://developers.google.com/youtube/?hl=de-DE`

Table 5.2.: Random Forest parameter combinations for the evaluation of each dataset.

Random Forest	Default	Example	Choice
Max depth	5	10	25
Min sample count	10	10	1
Reg. accuracy	0	0	0
Surrogate split	true	true	true
Max categories	10	15	14
Priors	0	0	0
Var importance	true	true	true
Nactive vars	0	4	0
Max trees	50	100	250
Forest accuracy	0.1	0.01	0.01
Term criteria	Max trees | Forest accuracy	Max trees	Max trees | Forest accuracy

5.2. Setup

The implementation details and configuration options were already presented in Chapter 4. But with the three large datasets, with the total amount of 250 hours, the various SIFT features and the various classification methods for training and evaluation, the number of keyframes, parameter options and configurations have to be minimized to make the computation time affordable.

5.2.1. Dataset Constraints

After extracting one and three keyframes per shot from all videos of the YouTube dataset and one keyframe per shot for the two TV datasets respectively, over $450,000$ keyframes were extracted, generating over 500 GBs of extracted features data, 99% of the amount coming from the extracted SIFT features. This number was computational and memory wise too high for processing. The global features extracted were kept for all video files, but the evaluation with three keyframes per shot was dropped. Furthermore, the maximum number of keyframes that SIFT features are extracted from per video was limited to 100 per video for the two TV datasets and 50 per video for the YouTube dataset.

5.2.2. Parameter Choices

This section will present the exact evaluation specifications, which differ for the various datasets. The YouTube dataset is considered the most important one to analyze and, therefore, lies in the focus of the evaluation. Not all SIFT descriptors are evaluated for the two TV datasets. The features used will be mentioned in the appropriate sections of the datasets.

Face detection. Cognitive features are extracted in four different types. Both, MCT and Haar cascade face detection is used, one time using only frontal face detectors and one time in combination with profile and side view detectors.

Table 5.3.: Decision Tree parameter combinations for the evaluation of each dataset.

Decision Tree	Default	Example	Choice
Max depth	INT_MAX	25	INT_MAX
Min sample count	10	5	1
Reg. accuracy	0	0	0
Surrogate split	true	true	true
Max categories	10	15	14
K–fold validation	10	15	0
1SE rule	true	false	false
Truncate tree	true	false	false
Priors	0	0	0

SIFT codebooks. Codebooks for the three SIFT descriptors are created using the following values: 250,000 samples for clustering of 1,000 codewords. Clustering is performed with k–means in three iterations and three different cluster center initializations and the *KMEANS_PP_CEN-TERS*[3] OpenCV parameter option.

SIFT descriptors. For each available SIFT descriptor, SIFT, OpponentSIFT and RGB–SIFT, features are extracted using the default settings for dense sampling ($scale = 1.2$ and $pixel\ space = 6$ with the 1×1–2×2 spatial pyramid option. This produces 1000–dimensional and 4000–dimensional feature vectors for the spatial pyramid.

SVM parameters. 3–fold cross–validation is used for all three datasets. Training and testing files are scaled. For the tree classifiers the unscaled input files are use. For the classification of the SIFT features with SVMs, subsets for SVM grid search and training a created. The subsets are created with a pos/neg ratio of 1:3. For SVM prediction the predicted classes from the 'global' features (see Section 3.2.5) and the SIFT features are fused. Grid search and SVM model training is further performed without class weighting. Grid search parameters are optimized for faster search, which means 3–fold instead of 5–fold validation and 42 instead of 110 parameter combinations are computed. This step was necessary to further reduce the SIFT SVM training time which has to be conducted for every genre. Only probability output is evaluated at this point.

Decision Tree/Random Forest. As for the SVM classifier the optimal parameter combination can not be defined individually. A grid search with several combinations would be best but was not available at the time of the evaluation. Therefore, three combinations of the available OpenCV parameters were tested. The *variable importance* is calculated always for decision tree classifiers. The first setting is the *default* OpenCV setting for these two classifiers. The second setting comes from the examples in the OpenCV package[4] and one example found on the world wide web[5] and the final setting was chosen personally. These values are presented in Table 5.2 and Table 5.3.

[3] Uses k–means++ center initialization by Arthur and Vassilvitskii
[4] samples/letter_recog.cpp
[5] `http://public.cranfield.ac.uk/c5354/teaching/ml/examples/c++/speech_ex/decisiontree.cpp`

5.3. Evaluation Results

This section presents the evaluation results on the three datasets. For each dataset the same information will be analyzed. First, an overview table highlighting overall accuracies from different parameter combinations as well as best baseline and extended system results, if available. Furthermore, confusion matrices for each classifier are presented to investigate classification performance on each genre or category and finally, single feature classification accuracies are presented for each genre computed with every classifier. Tree classifier results are presented only for the 'choice' parameter set since it always achieved the best results. Including the results of the other two parameter sets would be out of the scope of this thesis. Additional information like face detection samples and tree classifier structure example and tree classifier results can be found in the Appendix.

5.3.1. Italian RAI results

Table 5.4.: Average classification rates obtained on the RAI dataset. Comparison of classifiers, keyframe and SIFT descriptor influence. Best overall results are presented bolt. All available features are listed in Table 5.6

Classifier	System	Class. Rate
SVM	**Baseline on all frames (Aural + Cognitive + Structural + Visual)**	**99.6%**
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	97.7%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	97.3%
Random Forest Choice parameter set	Baseline on all frames (Aural + Cognitive + Structural + Visual)	97.3%
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	94.3%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	94.6%
Decision Tree Choice parameter set	Baseline on all frames (Aural + Cognitive + Structural + Visual)	96.2%
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	95.4%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	95.8%
SVM	**Extended on keyframes (all available features)**	**98.1%**

Overall the Italian RAI dataset achieves the best classification rates. An overview is given in Table 5.4. The first row shows the highest classification accuracy achieved with the

baseline system. A rate of 99.6% is obtained using all modalities, extracting features on all frames and performing classification with SVM classifiers. In comparison the extended system achieves an overall best accuracy of 98.1%, using all available features and SVM classifiers. The list of all available features is shown in Table 5.6. The drop in accuracy may have two reasons. First, the table shows that changing the extended system to keyframe extraction (2nd row) and using the same set of features and SVM classifiers, the performance drops to 97.7%. Second, including SIFT features the performance drops even more to 97.3%. Comparing these results clearly shows, that classification results vary with the fusion of different features. Many combinations were evaluated for this thesis. For the RAI dataset using all features proved most successfull for the extended system. This differs for the two other datasets as will be shown in the following subsections.

Single performances of each feature, which are dependent on frame input as shown in Table 5.7, show, that the visual features vary only slightly in performance on the RAI and Quaero dataset when using different numbers of input frames. By contrast, the cognitive feature performance drops around 10% on both datasets. The cognitive statstics are derived from the usage of the frontal OpenCV face detector. From this it follows, that performing face detection on keyframes alone is not a reasonable approach.

Table 5.5.: Confusion matrix obtained on the RAI dataset using the extended system and SVMs (%). Genre confusions over 15% are boxed for visualization purposes.

	Ca	Co	Fo	Mu	Ne	Ta	We
Ca	**100**	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0
Fo	0.0	0.0	**100**	0.0	0.0	0.0	0.0
Mu	0.0	14.3	0.0	**28.5**	14.3	0.0	42.9
Ne	0.0	0.0	0.0	0.0	**100**	0.0	0.0
Ta	0.0	0.0	0.0	0.0	0.0	**100**	0.0
We	0.0	0.0	0.0	0.0	0.0	0.0	**100**

Table 5.5 shows the confusion matrix for the extended system with SVM classifiers, the best new system as presented in Table 5.4. Confusion matrices of the tree classifiers are shown in the Appendix. Interesting is the fact, that decision trees perform better than random forest classifiers using the extended system. The genres and data in the RAI dataset seem easy enough to use decision tree classification, which performs model computation in a fraction of the time compared to SVM or Random Forest training. The small performance drop is acceptable if computation speed is a more important factor. In any case the SIFT features do not enhance the performance anymore and each genre shows best results using SVM classifiers, followed by decision trees and finally random forests. Only using SVM classifiers all genres except *music* achieve a 100% classification rate.

Individual feature performances on the RAI dataset are presented in the Table 5.6 for the best extended system. Tables for the other classifiers are presented in the Appendix. Looking at the aural feature and the *music* genre in Table 5.6 one notices, that an accuracy of 71.4% is reached. Comparing this to the confusion matrix rate for music (28.5%), the numbers confirm the possible performance drop while using different features and fusing

Table 5.6.: Single feature accuracy on the RAI dataset using the extended system and SVM (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Fo	Mu	Ne	Ta	We	Avg
Aural	92.6	96.6	**100**	71.4	91.8	**100**	98.3	95.8
MFCC_2	81.5	91.4	**100**	42.9	85.7	68.4	90.0	85.1
MFCC_3	22.2	91.4	**100**	0.0	83.7	52.6	78.3	72.4
SP_2	0.0	24.1	4.6	0.0	42.9	26.3	**48.3**	28.7
SP_3	3.7	**67.2**	4.6	0.0	8.2	5.3	43.3	28.1
ZCR_2	40.7	48.3	9.1	0.0	26.5	0.0	**88.3**	41.0
ZCR_3	0.0	51.7	0.0	0.0	8.2	26.3	**66.7**	32.2
Auto Color	63.0	98.3	90.9	42.9	95.9	79.0	**100**	89.6
Color Mom	85.2	94.8	**100**	57.1	95.9	92.1	98.3	93.9
HSV Hist	74.1	98.3	90.9	42.9	95.9	84.2	**100**	91.6
CoOccurrence	74.1	**100**	90.9	0.0	95.9	79.0	**100**	90.1
Edge Hist	96.3	94.3	90.9	0.0	93.9	81.6	**100**	91.2
Wavelet	96.3	**100**	**100**	28.6	95.9	81.6	96.7	93.5
Struct	37.0	**100**	72.7	0.0	87.8	71.1	96.7	81.3
Haar Front	14.8	84.5	54.6	0.0	61.2	86.8	**96.7**	71.2
Haar + Profile	44.4	65.5	86.4	0.0	63.3	**92.1**	91.7	72.8
MCT Front	14.8	82.8	**86.4**	14.3	71.4	76.3	83.3	71.3
MCT + Side	33.3	79.3	86.4	0.0	83.7	86.8	**90.0**	77.4
SIFT 1x1	-	-	-	-	-	-	-	-
SIFT 2x2	-	-	-	-	-	-	-	-
rgbSIFT 1x1	96.3	**100**	**100**	57.1	**100**	89.5	96.7	96.1
rgbSIFT 2x2	**100**	**100**	**100**	42.9	**100**	92.1	98.3	**96.9**
oppSIFT 1x1	**100**	**100**	**100**	57.1	**100**	81.6	98.3	95.8
oppSIFT 2x2	-	-	-	-	-	-	-	-

them in different combinations. This phenomenon can be sighted over all experiments as will be shown in the results on the two other datasets. This may indicate that individual feature combinations for each genre or category might improve the system performance. All single feature tables highlight the fact, that most features perform best on the *commercial* and *weather forecast* genre. The aural feature performs best for *music*, while SIFT features completely fail for the *music* genre using tree classifiers and perform a lot worse using SVM classifiers compared to all other genres. Overall all features except aural perform very bad for the *music* genre, which may be result of the very small number of videos in this genre. Cognitive features show most promissing results on *weather forecast* and visual features perform very good over all genres, in case of using tree classifiers even better than the SIFT descriptors. For the cartoon genre the SIFT features provide the most promissing results for all classifiers. Overall the SIFT descriptors have the highest accuracy rates over all genres.

Table 5.7.: Comparison of feature extraction on all frames and only on keyframes using SVM classifiers.

	RAI all frames	RAI keyframes	Quaero all frames	Quaero keyframes
AutoColor	83%	89%	85%	89%
ColorMom	95%	93%	91%	91%
CoOccurence	89%	90%	83%	78%
Edge Hist	93%	91%	88%	91%
HSV Hist	92%	91%	86%	88%
Wavelet	96%	93%	90%	91%
Cognitive	82%	71%	77%	66%

5.3.2. French TV Results

The overview results for the Quaero dataset show over 90% classification rates for the most important experiments (see Table 5.9). Comparing the basline system (1st row) and the best extended system (last row) two things become clear. The classification accuracy has been slightly improved, and the classification rate was reached using only the opponentSIFT descriptor (1x1 image region). This again proves that fusion of many, even promising features, can degrade the overall accuracy rates. Comparing keyframe and SIFT descriptor influence for this data the experiments show, that both keyframe usage and SIFT feature inclusion improve the results. Again, SVM classifiers perform best with 94.7%, followed by Random Forests and Decision Trees.

Table 5.8.: Confusion matrix obtained on the Quaero 2010 evaluation dataset using the extended system and SVMs (%). Genre confusions over 15% are boxed for visualization purposes.

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We
Ca	**100**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**99.2**	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0
Do	0.0	0.0	**91.7**	0.0	8.3	0.0	0.0	0.0	0.0	0.0
Ma	0.0	11.1	0.0	**81.5**	7.4	0.0	0.0	0.0	0.0	0.0
Mo	0.0	3.5	13.8	0.0	**82.7**	0.0	0.0	0.0	0.0	0.0
Ne	0.0	0.0	0.0	0.0	0.0	**94.7**	0.0	0.0	5.3	0.0
Sh	0.0	0.0	0.0	0.0	0.0	0.0	**100**	0.0	0.0	0.0
Ta	0.0	0.0	0.0	0.0	16.7	0.0	16.7	**66.6**	0.0	0.0
Tr	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100**	0.0
We	0.0	0.0	0.0	0.0	0.0	2.7	0.0	0.0	0.0	**97.3**

Looking at the confusion matrix for the best overall system in the Table 5.8, some interesting points leap to the eye. Comparing the difference in keyframe extraction using the SVM classifiers, the three lowest genre accuracies for *documentaries*, *show games* and *talk*

shows improve very much classifying with the opponentSIFT feature. Table 5.10 with the individual feature rates confirms, that all SIFT features perform very well for these genres. Unfortunately the fusion of more features degrades the overall performance.

Table 5.9.: Average correct classification rates obtained on the Quaero dataset. Comparison of classifiers, keyframe influence on features and SIFT descriptor inclusion. Best overall new result at the bottom compared to the best overall baseline system result at the top.

Classifier	System	Class. Rate
SVM	**Baseline on all frames (Aural + Cognitive + Structural + Visual)**	**94.5%**
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	94.0%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	94.6%
Random Forest Choice parameter set	Baseline on all frames (Aural + Cognitive + Structural + Visual)	91.3%
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	91.0%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	91.3%
Decision Tree Choice parameter set	Baseline on all frames (Aural + Cognitive + Structural + Visual)	87.3%
	Extended on keyframes (Aural + Cognitive + Structural + Visual)	90.7%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	91.7%
SVM	**Extended on keyframes (opponentSIFT 1x1)**	**94.7%**

The individual feature accuracy table for the extended system with SVM classification provides even more insight. The *cartoon* genre is either classified very good or missed completely by the different features. Only aural, visual and SIFT features prove useful for this genre. The most successful genres are again *commercial*, *weather forecast* and *traffic forecast*. All features achieve over 90% accuracy in the commercial accuracy. Comparing the three audio representations, the aural descriptor performs best. From the 2nd and 3rd representation only the MFCC feature looks promissing for genre classification compared to ZCR and signal energy. Also the 2nd feature representation shows better accuracy results than the 3rd representation. These observations are backed up by the results on the RAI dataset in Table 5.6. Again, SIFT descriptors show the highest average accuracy rates followed by the aural and visual features, compared to the other features.

Table 5.10.: Single feature accuracy on the Quaero dataset using the extended system and SVM (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We	Avg
Aural	66.7	98.4	58.3	74.1	96.6	89.5	91.3	16.7	**100**	94.6	91.0
MFCC_2	0.0	**97.6**	16.7	29.6	75.9	52.6	65.2	0.0	76.5	91.9	76.0
MFCC_3	0.0	**97.6**	25.0	26.0	55.2	63.2	30.4	0.0	41.2	83.8	69.1
SP_2	0.0	**96.1**	0.0	18.5	20.7	5.7	0.0	0.0	0.0	37.8	49.4
SP_3	0.0	**97.6**	0.0	0.0	13.8	0.0	0.0	0.0	0.0	2.7	43.0
ZCR_2	0.0	**93.7**	0.0	18.5	13.8	0.0	0.0	0.0	0.0	62.2	50.4
ZCR_3	0.0	**96.1**	0.0	11.1	0.0	0.0	0.0	0.0	0.0	18.9	44.0
Auto Color	**100**	97.6	66.7	77.8	86.2	84.2	87.0	0.0	94.2	94.6	89.3
Color Mom	33.3	97.6	83.3	77.8	96.6	94.7	82.6	0.0	**100**	97.3	91.4
HSV Hist	**100**	96.9	50.0	74.1	79.3	89.5	87.0	16.7	**100**	94.6	88.4
CoOccurrence	0.0	**96.9**	8.3	40.7	72.4	84.2	60.9	0.0	82.4	94.6	78.3
Edge Hist	0.0	98.4	91.7	63.0	93.1	94.7	95.6	0.0	**100**	97.3	91.0
Wavelet	0.0	97.6	83.3	74.1	89.7	94.7	87.0	33.3	**100**	97.3	91.0
Struct	0.0	**98.4**	58.3	22.2	17.2	73.7	21.7	0.0	76.5	86.5	69.1
Haar Front	0.0	**92.9**	0.0	22.2	79.3	57.9	47.9	16.7	29.4	62.2	66.0
Haar + Profile	0.0	**92.1**	0.0	33.3	86.2	47.4	69.6	16.7	53.0	64.9	70.0
MCT Front	0.0	**95.3**	0.0	44.4	72.4	63.2	65.2	0.0	35.3	83.8	72.7
MCT + Side	0.0	**93.7**	0.0	25.9	65.5	42.1	69.6	0.0	29.4	73.0	67.0
SIFT 1x1	**100**	99.2	91.7	66.7	82.8	94.7	95.7	66.7	**100**	97.3	93.0
SIFT 2x2	-	-	-	-	-	-	-	-	-	-	-
rgbSIFT 1x1	**100**	99.2	91.7	66.7	86.2	94.7	**100**	66.7	**100**	97.3	93.7
rgbSIFT 2x2	-	-	-	-	-	-	-	-	-	-	-
oppSIFT 1x1	**100**	99.2	91.7	81.5	82.8	94.7	**100**	66.7	**100**	97.3	**94.7**
oppSIFT 2x2	**100**	99.2	91.7	77.8	86.2	94.7	**100**	50.0	**100**	97.3	94.4

5.3.3. YouTube Results

The results on the YouTube dataset further validate the findings of the TV domain experiments of the genre classification system. Table 5.11 shows that once again the SVM classifiers perform best with an overall classification rate of 44.0%. Interesting is the fact, that again another combination of features made this classification accuracy possible. Fusing all visual and all SIFT features achieved a better classification rate than including aural, structural and cognitive information. SIFT features alone manage to classify the dataset with an 42.6% accuracy, while all other features together excluding SIFT achieve 28.2%. Fusion of all feature modalities is slightly worse with 42.8% than the best overall performance mentioned before. The impression is, that SIFT features prove more successful the more difficult the data gets compared to all the other features. The same applies to the different classifiers. The classifier performances disperse even more on this challenging web video dataset. Best Random Forest classification rate is 39.9% incorporating all feature modalities and Decision Trees performance peaks at 26.9% with the same set of features.

Table 5.11.: Average correct classification rates obtained on the YouTube dataset. Comparison of classifiers and SIFT descriptor inclusion. Best overall new result at the bottom.

Classifier	System	Class. Rate
SVM	Extended on keyframes (Aural + Cognitive + Structural + Visual)	28.2%
	Extended on keyframes (SIFT)	42.6%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	42.8%
Random Forest Choice parameter set	Extended on keyframes (Aural + Cognitive + Structural + Visual)	35.9%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	39.9%
Decision Tree Choice parameter set	Extended on keyframes (Aural + Cognitive + Structural + Visual)	24.0%
	Extended on keyframes (Aural + Cognitive + Structural + Visual + SIFT)	26.9%
SVM	**Extended on keyframes (Visual + SIFT)**	**44.0%**

A lower accuracy performance due to high diversity and more difficult categories was expected for this domain, but the confusion matrix in Table 5.12 for the overall best experiment shows that the low overall classification rate may also be the result of the high number of categories and very low classification rates for some of these categories. The

top three categories are *autos*, *howto* and *entertainment* with 80.0%, 72.0% and 56.0%, respectively. *Sports* and *travel* are also over 50%. But there are also many categories with very low classification rates. The three lowest performances being news, people and film, and education and science with 22.0%, 24.0% and 32.0% resepectively, dropping the average accuracy.

Comparing these results to the single feature results in Table 5.13, one can see, that several category performances are higher using one feature for classification than the fusion of visual and SIFT features for the overall best performance. Example categories are *animals*, *games*, *science*, *sports* and most importantly *news* and *people*. The *news* category classification performance is most disappointing. First of all, the accuracy drops from 40.0% to 22.0% by using the best overall system with visual and SIFT featues instead of only MCT based frontal face detection. And second, it is one of the categories overlapping with the TV domain where classification rates above 90% were reached for both datasets. This example shows that the big diversity in one catgory can lead to a very huge performance drop. More examples of the news category and all other YouTube categories can be found in the Appendix. The small number of frames for each category easily shows, that categories have high diversity and could belong to other categories as well than the one assigned to them.

The performances of the single features a very low compared to the the experiments in the TV domain. Audio, structural and some cognitive features drop below 10% average classification accuracy. Only color moments, edge histogram and wavelet texture reach around 20%, SIFT features are around the 40% classification rate. SIFT features perform above average for *autos* (∼76%), *entertainment* (∼52%), *howto* (∼64%), *sports* (∼50%) and *travel* (∼44%). All SIFT features perform best for *autos* and almost all visual features perform best for *howto* videos. Audio features are very low except for single outstanding performances in different categories. For example, signal energy using the second feature representation achieves 0.0% classification rates for most of the categories except for *entertainment* with 48.0%. Similar findings apply to the strcutural feature with a 36.0% classification rate in the *science* category and MCT based frontal face detection in the news category with 40.0%.

Table 5.12.: Confusion matrix obtained on the YouTube evaluation dataset and SVMs (%). Genre confusions over 15% are boxed for visualization purposes.

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr
Activism	**48.0**	6.0	2.0	0.0	2.0	14.0	8.0	2.0	0.0	6.0	2.0	2.0	0.0	8.0
Animals	18.0	**40.0**	0.0	0.0	2.0	0.0	2.0	4.0	2.0	2.0	8.0	0.0	4.0	18.0
Autos	2.0	0.0	**80.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	2.0	6.0	8.0
Comedy	8.0	2.0	4.0	**42.0**	0.0	12.0	8.0	8.0	2.0	0.0	0.0	2.0	0.0	12.0
Education	22.0	2.0	6.0	2.0	**32.0**	0.0	6.0	4.0	4.0	4.0	2.0	8.0	0.0	8.0
Entertainment	6.0	4.0	0.0	8.0	2.0	**56.0**	2.0	2.0	0.0	6.0	2.0	4.0	4.0	4.0
Film	18.0	4.0	2.0	2.0	0.0	2.0	**24.0**	6.0	0.0	10.0	4.0	6.0	0.0	22.0
Games	10.0	0.0	8.0	4.0	2.0	0.0	14.0	**44.0**	0.0	2.0	2.0	2.0	6.0	6.0
Howto	4.0	6.0	2.0	0.0	0.0	4.0	0.0	0.0	**72.0**	0.0	2.0	4.0	2.0	4.0
News	18.0	4.0	4.0	2.0	8.0	8.0	12.0	2.0	0.0	**22.0**	4.0	8.0	2.0	6.0
People	12.0	8.0	2.0	6.0	4.0	6.0	4.0	4.0	4.0	0.0	**24.0**	4.0	2.0	20.0
Science	8.0	8.0	6.0	2.0	6.0	10.0	4.0	6.0	2.0	0.0	4.0	**32.0**	8.0	4.0
Sports	4.0	4.0	8.0	0.0	0.0	2.0	2.0	4.0	0.0	4.0	4.0	4.0	**52.0**	12.0
Travel	4.0	6.0	6.0	4.0	0.0	10.0	8.0	4.0	0.0	0.0	4.0	0.0	4.0	**50.0**

Table 5.13.: Single feature accuracy on the YouTube dataset using the extended system and SVM (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr	Avg
Aural	6.0	4.0	10.0	14.0	2.0	14.0	4.0	0.0	0.0	0.0	0.0	8.0	2.0	**28.0**	6.6
MFCC_2	10.0	0.0	4.0	6.0	8.0	4.0	**20.0**	0.0	8.0	6.0	0.0	2.0	8.0	2.0	5.6
MFCC_3	0.0	0.0	18.0	10.0	2.0	**20.0**	6.0	6.0	4.0	6.0	6.0	6.0	16.0	2.0	7.2
SP_2	4.0	0.0	0.0	8.0	12.0	**48.0**	0.0	0.0	0.0	0.0	22.0	6.0	0.0	0.0	7.1
SP_3	2.0	2.0	6.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	28.0	0.0	**36.0**	2.0	6.0
ZCR_2	0.0	0.0	0.0	0.0	12.0	14.0	4.0	2.0	12.0	0.0	16.0	8.0	**24.0**	12.0	7.4
ZCR_3	0.0	0.0	0.0	28.0	2.0	6.0	0.0	4.0	0.0	8.0	**32.0**	10.0	8.0	2.0	7.2
Auto Color	2.0	4.0	12.0	14.0	8.0	44.0	6.0	0.0	**64.0**	22.0	18.0	8.0	28.0	4.0	16.7
Color Mom	10.0	20.0	24.0	26.0	24.0	44.0	6.0	20.0	**60.0**	16.0	12.0	8.0	22.0	22.0	22.4
HSV Hist	8.0	8.0	30.0	10.0	28.0	46.0	6.0	14.0	**58.0**	8.0	12.0	10.0	20.0	10.0	19.1
CoOccurrence	0.0	0.0	**68.0**	0.0	0.0	24.0	2.0	50.0	14.0	2.0	4.0	14.0	4.0	0.0	13.0
Edge Hist	0.0	0.0	48.0	36.0	26.0	42.0	0.0	18.0	**54.0**	18.0	6.0	16.0	20.0	0.0	20.7
Wavelet	10.0	4.0	42.0	34.0	28.0	20.0	10.0	20.0	**58.0**	20.0	16.0	0.0	8.0	32.0	21.9
Struct	6.0	10.0	6.0	2.0	8.0	22.0	0.0	4.0	20.0	0.0	0.0	**36.0**	2.0	2.0	8.4
Haar Front	4.0	**30.0**	2.0	8.0	4.0	22.0	2.0	0.0	0.0	24.0	28.0	8.0	4.0	0.0	9.7
Haar + Profile	4.0	4.0	12.0	18.0	24.0	**30.0**	0.0	2.0	8.0	16.0	14.0	0.0	24.0	8.0	11.7
MCT Front	10.0	0.0	0.0	0.0	2.0	16.0	6.0	28.0	8.0	**40.0**	4.0	0.0	0.0	16.0	9.3
MCT + Side	0.0	0.0	12.0	0.0	4.0	**36.0**	0.0	6.0	16.0	16.0	2.0	0.0	22.0	28.0	10.2
SIFT 1x1	40.0	38.0	**78.0**	40.0	28.0	54.0	24.0	40.0	62.0	20.0	16.0	26.0	52.0	48.0	**40.4**
SIFT 2x2	38.0	32.0	**74.0**	36.0	28.0	50.0	22.0	34.0	68.0	22.0	22.0	30.0	42.0	34.0	37.9
rgbSIFT 1x1	32.0	32.0	**78.0**	38.0	32.0	52.0	24.0	42.0	60.0	22.0	22.0	30.0	48.0	50.0	40.1
rgbSIFT 2x2	46.0	36.0	**76.0**	38.0	28.0	48.0	22.0	34.0	64.0	26.0	20.0	30.0	38.0	48.0	39.5
oppSIFT 1x1	34.0	32.0	**76.0**	42.0	26.0	52.0	24.0	44.0	60.0	16.0	20.0	30.0	54.0	44.0	39.5
oppSIFT 2x2	46.0	42.0	**76.0**	36.0	28.0	50.0	24.0	36.0	64.0	18.0	20.0	34.0	48.0	44.0	**40.4**

6. Conclusion & Future Work

The focus of this study was to build a genre classification system, that automatically tags videos with predefined labels. The system is evaluated on two different domains, the TV broadcast domain and the web video domain. TV programs are classified into genres and web videos are tagged with category labels. The system utilizes an extensive set of features from different modalities and fuses information from aural, visual, structural and cognitive content representations. These features are classified with different machine learning algorithms, the support vector machines and decision tree classifiers.

The performance on two TV domain datasets was 98.1% and 94.7%. The classification accuracy on one YouTube dataset from the web domain reached 44.0%. Some interesting aspects can be seen across all experiments. The tree classifiers always performed best using the ‘choice‘ parameter set. In very few instances the ‘example‘ parameter set achieved around the same classification performance. Still the SVM classification outperformed the tree classifiers on all datasets. The reason for this may be the grid search for optimal SVM training model parameters optimizing the classification rate. Another reason may be that SVM classification was performed binary, one vs. all, and not multi–class like with the tree classifiers. As for the features some interesting reoccurrences lead to the following conclusions. As for audio feature representation the single aural feature representation of all audio features works best in the TV domain. Performance on the YouTube dataset shows the same low accuracy as the other aural feature representations with peaks in different categories. Cognitive feature performance rate dropped using keyframes for face detection compared to using all video frames. Their performance could not keep up with visual or SIFT features, but proved useful for some genres like news or weather forecast and could be improved even further in the future. The visual features achieve around the best classification rates on all datasets with a slight drop compared to SIFT features on the YouTube dataset. The SIFT features perform best on all datasets and clearly distinguish themselves from the other features the more difficult the data gets. For all datasets they achieve highest accuracy rates for a majority of genres/categories. But, high computational time and SVM model training because of the high dimensionality are their

weakness compared to the "global"[1] low–dimensional visual features.

Finally, patterns emerge for individual features for specific genres and categories and at the same time it can be noticed that overall best performances come from the fusion of different number of features. For some cases single features perform better than any combination of two or more features. In other cases performance drops fusing more and more features. This two points strongly indicate that the classification of many genres could benefit from individual classification approaches utilizing handpicked or empirical chosen features instead of a general approach.

6.1. Future Work

The related work chapter already showed clearly that a lot of research areas can be investigated and included in the study of multimedia genre/category tagging. Several potential future work options are:

SIFT training model time. The limitations of the performed evaluation were already explained. Investing more time and resources, the evaluation can be done with different keyframe options and without using subset limitations for training the classifiers.

Face detection. The results showed that improving the face detection rate enhances the accuracy of the cognitive feature. Simultaneously switching the feature extraction from all frames to keyframes degraded the cognitive feature results. A solution regarding both aspects could be the utilization of a tracking based face detector on the whole video. The possibility to achieve more robust face detection over all frames and thus compute a more reliable cognitive feature vector could lead to promising results.

Temporal features. Usage of keyframes brought the system further away of using temporal features like HoG and HoF, which are used for action detection. For single shots the temporal information could be vital to distinguish between slow and and fast paced visual material, and measure different motions in the videos.

Mid–level semantics. The idea of using mid–level features was not investigated for this system yet. Several possibilities are available which are not mutually exclusive like action features, LSA and audio segmentation with categories like silence, noise, speech and music, maybe even more detailed like motor sounds, live audience and animal sounds. This information can be used on a shot–level basis.

ASR. The most important future work would be including an automatic speech recognition system to perform classification on the spoken words in each video, which would be carried out with typical documentation classification methods like presented in the related work chapter. Especially for the web video domain information about the topic of a video is more closely related to the category than the usual genre characteristics.

Classification. Inspiration could be found by the two face detection methods presented in this thesis, the Haar and MCT cascade classification. Building individual classifier cascades for each genre eliminating videos not belonging to this genre can be a better approach instead of the typical classification.

[1] One feature vector for the whole video.

Individual frameworks. The evaluation revealed two interesting points. For one, the fusion of many unreliable features is not helping the overall classification results. And second, some features are better suited for specific genres or categories. Instead of using a generic system with the same features for all genres or categories, individual combinations for each genre could prove to be more successful.

Bibliography

[BC08] D. Brezeale and D. Cook, "Automatic Video Classification: A Survey of the Literature," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,*, vol. 38, no. 3, pp. 416–430, May 2008.

[BCFG00] F. Brugnara, M. Cettolo, M. Federico, and D. Giuliani, "A System for the Segmentation and Transcription of Italian Radio News," in *Proceedings of the RIAO conference*, 2000, pp. 364–371.

[BFOS84] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Monterey, CA: Wadsworth and Brooks, 1984.

[BHK+09] D. Borth, J. Hees, M. Koch, A. Ulges, C. Schulze, T. Breuel, and R. Paredes, "TubeFiler: an automatic web video categorizer," in *Proceedings of the 17th ACM international conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 1111–1112.

[Bra00] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. [Online]. Available: http://opencv.willowgarage.com/wiki/

[Bur98] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, Jun. 1998.

[CH00] H. Chen and T. K. Ho, *Evaluation of Decision Forests on Text Categorization.* MIT Press, 2000, vol. 3967, no. 2, pp. 191–199.

[CHE+06] M. Campbell, A. Haubold, S. Ebadollahi, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and L. Xie, "IBM research TRECVID-2006 video retrieval system," in *TREC Video Retrieval Evaluation Proceedings*, 2006.

[CL11] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[CSK11] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning." Microsoft Research, Tech. Rep. MSR-TR-2011-114, Oct 2011.

[CZS+09] J. Cao, Y. Zhang, Y. Song, Z. Chen, X. Zhang, and J. Li, "MCG-WEBV: A Benchmark Dataset for Web Video Analysis," Technical Report, Institute of Computing Technology, Mai 2009.

[DDF+90] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[EFG+07] H. K. Ekenel, M. Fischer, H. Gao, K. Kilgour, J. S. Marcos, and R. Stiefelhagen, "Universität Karlsruhe (TH) at TRECVID 2007," in *Proc TRECVID Evaluation Workshop*, 2007, pp. 4–7.

[EGS08] H. K. Ekenel, H. Gao, and R. Stiefelhagen, "Universität Karlsruhe (TH) at TRECVID 2008," in *Proceedings of NIST TRECVID Workshop*, 2008.

[FE04] B. Fröba and A. Ernst, "Face detection with the modified census transform," in *Proceedings of the Sixth IEEE international conference on Automatic face and gesture recognition*, ser. FGR' 04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 91–96.

[FLE95] S. Fischer, R. Lienhart, and W. Effelsberg, "Automatic recognition of film genres," in *Proceedings of the third ACM international conference on Multimedia*, ser. MULTIMEDIA '95. New York, NY, USA: ACM, 1995, pp. 295–304.

[HKM+97] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," in *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, ser. CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, pp. 762–.

[Ho95] T. K. Ho, "Random decision forests," in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ser. ICDAR '95. Washington, DC, USA: IEEE Computer Society, 1995, pp. 278–.

[Hof99] T. Hofmann, "Probabilistic Latent Semantic Analysis," in *Proceedings of Uncertainty in Artificial Intelligence, UAI'99*, 1999, pp. 289–296.

[LFL98] T. Landauer, P. Foltz, and D. Laham, "An Introduction to Latent Semantic Analysis," *Discourse processes*, vol. 25, pp. 259–284, 1998.

[Lie01] R. Lienhart, "Reliable Transition Detection in Videos: A Survey and Practitioner's Guide," *Internation Journal of Image and Graphics*, vol. 1, no. 3, pp. 469–486, 2001.

[LK02] E. Leopold and J. Kindermann, "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?" *Machine Learning*, vol. 46, pp. 423–444, March 2002.

[LLC+07] A. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa, "Kodak's consumer video benchmark data set: concept definition and annotation," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 245–254.

[LLZ01] L. Lu, S. Z. Li, and H.-J. Zhang, "Content-based audio segmentation using support vector machines," in *IEEE International Conference on Multimedia and Expo, 2001. ICME 2001*, aug. 2001, pp. 749 –752.

[Log00] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," in *International Symposium on Music Information Retrieval*, 2000.

[Low04] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, ser. CVPR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 2169–2178.

[LYCR05] S. Liu, H. Yi, L.-T. Chia, and D. Rajan, "Adaptive hierarchical multi-class SVM classifier for texture-based image classification," in *IEEE International Conference on Multimedia and Expo, 2005. ICME 2005,*, july 2005, p. 4 pp.

[MAT10] MATLAB, *version 7.10.0 (R2010a)*. Natick, Massachusetts: The MathWorks Inc., 2010. [Online]. Available: http://www.mathworks.de/products/matlab/

[MM07] M. Montagnuolo and A. Messina, "TV Genre Classification Using Multimodal Information and Multilayer Perceptrons," in *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence on AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, ser. AI*IA '07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 730–741.

[MM08a] A. Messina and M. Montagnuolo, "Fuzzy mining of multimedia genre applied to television archives," in *IEEE International Conference on Multimedia and Expo, 2008*, 23 2008-april 26 2008, pp. 117 –120.

[MM08b] ——, "Multimedia genre characterisation with fuzzy embedding classifiers," in *Proceedings of the 2008 Ambi-Sys workshop on Ambient media delivery and interactive television*, ser. AMDIT '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 7:1–7:8.

[MM09] M. Montagnuolo and A. Messina, "Parallel neural networks for multimodal video genre classification," *Multimedia Tools Appl.*, vol. 41, pp. 125–159, January 2009.

[Oka12] (2012) Okapi Library. [Online]. Available: http://cvhci.anthropomatik.kit.edu/okapi/trac/

[PR09] K. Punera and S. Rajan, "Improved Multi Label Classification in Hierarchical Taxonomies," in *Proceedings of the 2009 IEEE International Conference on Data Mining Workshops*, ser. ICDMW '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 388–393.

[Pyt11] (2011) Python Software. [Online]. Available: http://www.python.org/

[Qui93] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[Sau96] J. Saunders, "Real-time discrimination of broadcast speech/music," in *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. on Conference Proceedings., 1996 IEEE International Conference - Volume 02*, ser. ICASSP '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 993–996.

[SB91] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vision*, vol. 7, no. 1, pp. 11–32, Nov. 1991.

[SC12] S. Sadanand and J. J. Corso, "Action Bank: A High-Level Representation of Activity in Video," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[SO95] M. Stricker and M. Orengo, "Similarity of Color Images," in *Proceedings SPIE Storage and Retrieval for Image and Video Databases*, vol. 2420, Washington DC, USA, 1995, pp. 381–392.

[SOD10] A. F. Smeaton, P. Over, and A. R. Doherty, "Video shot boundary detection: Seven years of TRECVid activity," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411–418, Apr. 2010.

[SR09] A. P. Santos and F. Rodrigues, *Multi-label Hierarchical Text Classification using the ACM Taxonomy*, 2009, pp. 553–564.

[SvdSdR+08] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, J. C. van Gemert, J. R. R. Uijlings, J. He, X. Li, I. Everts, V. Nedović, M. van Liempt, R. van Balen, F. Yan, M. A. Tahir, K. Mikolajczyk, J. Kittler, M. de Rijke, J.-M. Geusebroek, T. Gevers, M. Worring, A. W. M. Smeulders, and D. C. Koelma, "The MediaMill TRECVID 2008 semantic video search engine," in *Proceedings of the 6th TRECVID Workshop*, Gaithersburg, USA, November 2008.

[SvdSdR+10] C. G. M. Snoek, K. E. A. van de Sande, O. de Rooij, B. Huurnink, E. Gavves, D. Odijk, M. de Rijke, T. Gevers, M. Worring, D. C. Koelma, and A. W. M. Smeulders, "The MediaMill TRECVID 2010 Semantic Video Search Engine," in *Proceedings of the TRECVID Workshop*, 2010.

[SvdSL+11] C. G. M. Snoek, K. E. A. van de Sande, X. Li, M. Mazloom, Y.-G. Jiang, D. C. Koelma, and A. W. M. Smeulders, "The MediaMill TRECVID 2011 semantic video search engine," in *Proceedings of the 9th TRECVID Workshop*, Gaithersburg, USA, December 2011.

[SWG+05] C. G. M. Snoek, M. Worring, J.-M. Geusebroek, D. Koelma, and F. J. Seinstra, "On the surplus value of semantic video analysis beyond the key frame," in *ICME*, 2005, pp. 386–389.

[SZ03] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *Proceedings of the International Conference on Computer Vision*, vol. 2, Oct. 2003, pp. 1470–1477.

[SZYW10] Y. Song, M. Zhao, J. Yagnik, and X. Wu, "Taxonomic classification for web-based videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, june 2010, pp. 871 –878.

[SZZ+09] Y. Song, Y.-d. Zhang, X. Zhang, J. Cao, and J.-T. Li, "Google challenge: incremental-learning for web video categorization on robust semantic feature space," in *Proceedings of the 17th ACM international conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 1113–1114.

[TC02] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293 – 302, jul 2002.

[USKB07] A. Ulges, C. Schulze, D. Keysers, and T. M. Breuel, "Content-based Video Tagging for Online Video Portals," 2007.

[Vap95] V. N. Vapnik, *The nature of statistical learning theory.* New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[vdSGS10] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[vGVSG10] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek, "Visual Word Ambiguity," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1271–1283, Jul. 2010.

[VJ01] P. Viola and M. Jones, "Robust Real-time Object Detection," *International Journal of Computer Vision*, vol. 57, no. 2, p. 137–154, 2001.

[WZN09] X. Wu, W.-L. Zhao, and C.-W. Ngo, "Towards google challenge: combining contextual and social information for web video categorization," in *Proceedings of the 17th ACM international conference on Multimedia*, ser. MM '09. New York, NY, USA: ACM, 2009, pp. 1109–1110.

[WZS+10] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li, "YouTubeCat: Learning to categorize wild web videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, june 2010, pp. 879 –886.

[YH08] J. Yang and A. G. Hauptmann, "(Un)Reliability of video concept detection," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, ser. CIVR '08. New York, NY, USA: ACM, 2008, pp. 85–94.

[YJHN07] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 197–206.

[YLM+06] X. Yuan, W. Lai, T. Mei, X.-S. Hua, X.-Q. Wu, and S. Li, "Automatic Video Genre Categorization using Hierarchical SVM," in *IEEE International Conference on Image Processing, 2006*, oct. 2006, pp. 2905 –2908.

[YLYH07] L. Yang, J. Liu, X. Yang, and X.-S. Hua, "Multi-modality web video categorization," in *Proceedings of the international workshop on Workshop on multimedia information retrieval*, ser. MIR '07. New York, NY, USA: ACM, 2007, pp. 265–274.

[YP97] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, ser. ICML '97. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 412–420.

[ZKS93] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, Jan. 1993.

[ZSC+09] X. Zhang, Y.-C. Song, J. Cao, Y.-D. Zhang, and J.-T. Li, "Large scale incremental web video categorization," in *Proceedings of the 1st workshop on Web-scale multimedia corpus*, ser. WSMC '09. New York, NY, USA: ACM, 2009, pp. 33–40.

[ZZMP08] S. Zanetti, L. Zelnik-Manor, and P. Perona, "A walk through the webs video clips," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08.*, june 2008, pp. 1 –8.

Appendix

A. Shot Detection File

```
<shotBoundaryResults>
  <shotBoundaryResult sysId="Music_Domenica_in_20060409_1825.avi.sbd"
  totalRunTime="0.0" totalDecodeTime="0.0" totalSegmentationTime="0.0"
  processorTypeSpeed="Quad-Core AMD Opteron(tm) Processor 2354">
    <seg src="Music_Domenica_in_20060409_1825.avi">
      <trans type="CUT" preFNum="3524" postFNum="3525"/>
      <trans type="CUT" preFNum="7008" postFNum="7009"/>
      <trans type="CUT" preFNum="7064" postFNum="7065"/>
      <trans type="CUT" preFNum="7217" postFNum="7218"/>
      <trans type="CUT" preFNum="7275" postFNum="7276"/>
      <trans type="CUT" preFNum="7388" postFNum="7389"/>
      <trans type="CUT" preFNum="7426" postFNum="7427"/>
      <trans type="CUT" preFNum="7528" postFNum="7529"/>
      <trans type="CUT" preFNum="7750" postFNum="7751"/>
      <trans type="CUT" preFNum="8352" postFNum="8353"/>
      <trans type="CUT" preFNum="8405" postFNum="8406"/>
      <trans type="CUT" preFNum="8630" postFNum="8631"/>
      <trans type="CUT" preFNum="8743" postFNum="8744"/>
      <trans type="CUT" preFNum="8775" postFNum="8776"/>
    </seg>
  </shotBoundaryResult>
</shotBoundaryResults>
```

B. Config File

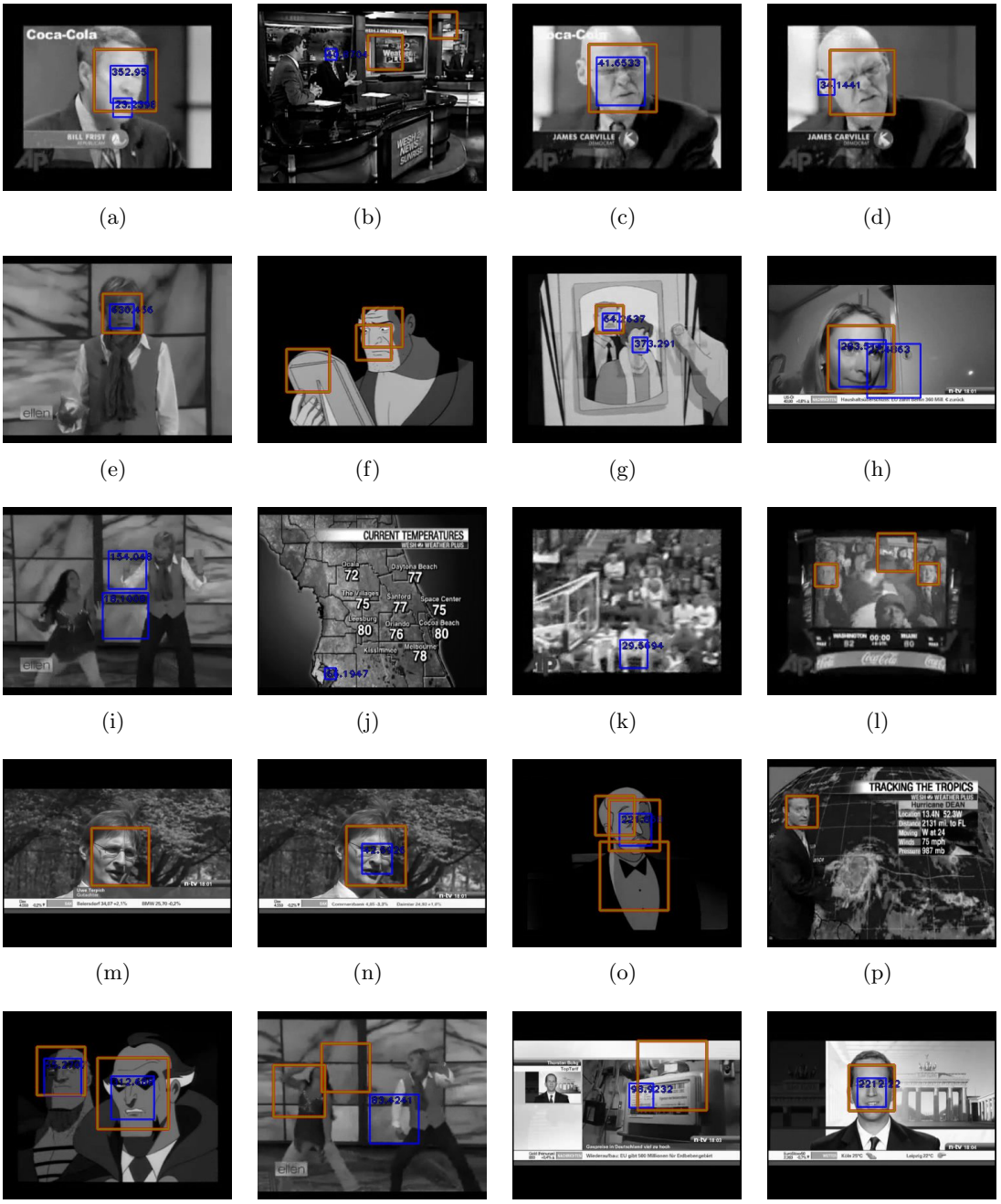
```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <!--video OR key-->
    <source>key</source>
    <keyframes>
        <number>1</number>
        <max>100</max>
        <!--0 = shot ; 1 = video ; 2 = video if not max else shot-->
        <type>0</type>
    </keyframes>
    <do>
        <visual>0</visual>
        <structural>0</structural>
        <cognitive>0</cognitive>
        <sift>0</sift>
        <codebook>0</codebook>
        <treeClassifier>0</treeClassifier>
    </do>
    <facedetection>
        <mct>0</mct>
        <profile>0</profile>
        <detector>face_frontal_new.xml</detector>
        <detector>face_45deg_new.xml</detector>
        <mct_confidence>0</mct_confidence>
        <haar>0</haar>
        <profile>0</profile>
        <cascade>haarcascade_frontalface_default.xml</cascade>
        <cascade>haarcascade_profileface.xml</cascade>
        <lbp>0</lbp>
        <cascade>lbpcascade_frontalface.xml</cascade>
        <!-- save images with face rectangles-->
        <prove>0</prove>
    </facedetection>
    <sift>
        <!--harrislaplace OR densesampling-->
        <dectector>densesampling</dectector>
        <!--1.2 OR 1.2+2.0 etc-->
        <scale>1.2</scale>
        <pixel>6</pixel>
        <!--sift OR opponentsift OR rgbsift-->
        <descriptor>opponentsift</descriptor>
        <!--pyramid-1x1-2x2 OR pyramid-1x1-2x2-1x3 OR pyramid-1x1-->
        <spatialpyramid>pyramid-1x1-2x2-1x3</spatialpyramid>
    </sift>
    <codebook>
```

```
        <size>1000</size>
        <samples>250000</samples>
        <iterations>3</iterations>
        <!--KMEANS_RANDOM_CENTERS OR KMEANS_PP_CENTERS-->
        <centerinit>KMEANS_PP_CENTERS</centerinit>
        <centertrys>3</centertrys>
    </codebook>
    <treeClassifier>
        <path>./classification/</path>
        <kfold>3</kfold>
        <!--0 = italy, 1 = quaero, 2 = youtube-->
        <genres>2</genres>
        <do_norm>1</do_norm>
        <do_sift>1</do_sift>
        <randomtree>1</randomtree>
        <decisiontree>1</decisiontree>
        <trainmodel>0</trainmodel>
        <usemodel>1</usemodel>
        <!--parameter set: 0 = default : 1 = choice : 2 = example-->
        <choice>0</choice>
    </treeClassifier>
</config>
```

C. Folder Structure

- / ..... tvGenre executeable binary
  - audio ..... Matlab audio analysis
    - results ..... Extracted audio features
    - Voicebox ..... Voicebox Toolkit
  - classification ..... SVM and Tree classification folder
    - Data ..... SVM input data
      - For_Genre1
      - For_Genre2
      - ...
    - Kfold ..... SVM kfold data
      - For_Genre1
      - For_Genre2
      - ...
    - Results ..... SVM models, predicitions and results
      - For_Genre1
      - For_Genre2
      - ...
    - py_tvgenre.py
    - py_create_SVM_input.py
    - py_grid.py
    - py_subset.py
  - codebook ..... SIFT binary and codebooks
    - images ..... Codebook creation sample images
  - config ..... Default config file location
  - features ..... Extracted features
    - Features_for_Video_1
    - ...
  - haarcascades ..... Haar cascades
  - key ..... Extracted keyframes
    - Keyframes_for_Video_1
    - ...
  - mct ..... MCT cascade files
  - shot ..... Shot detection files
  - video ..... Video files

D. Face Detection



(a) (b) (c) (d)

(e) (f) (g) (h)

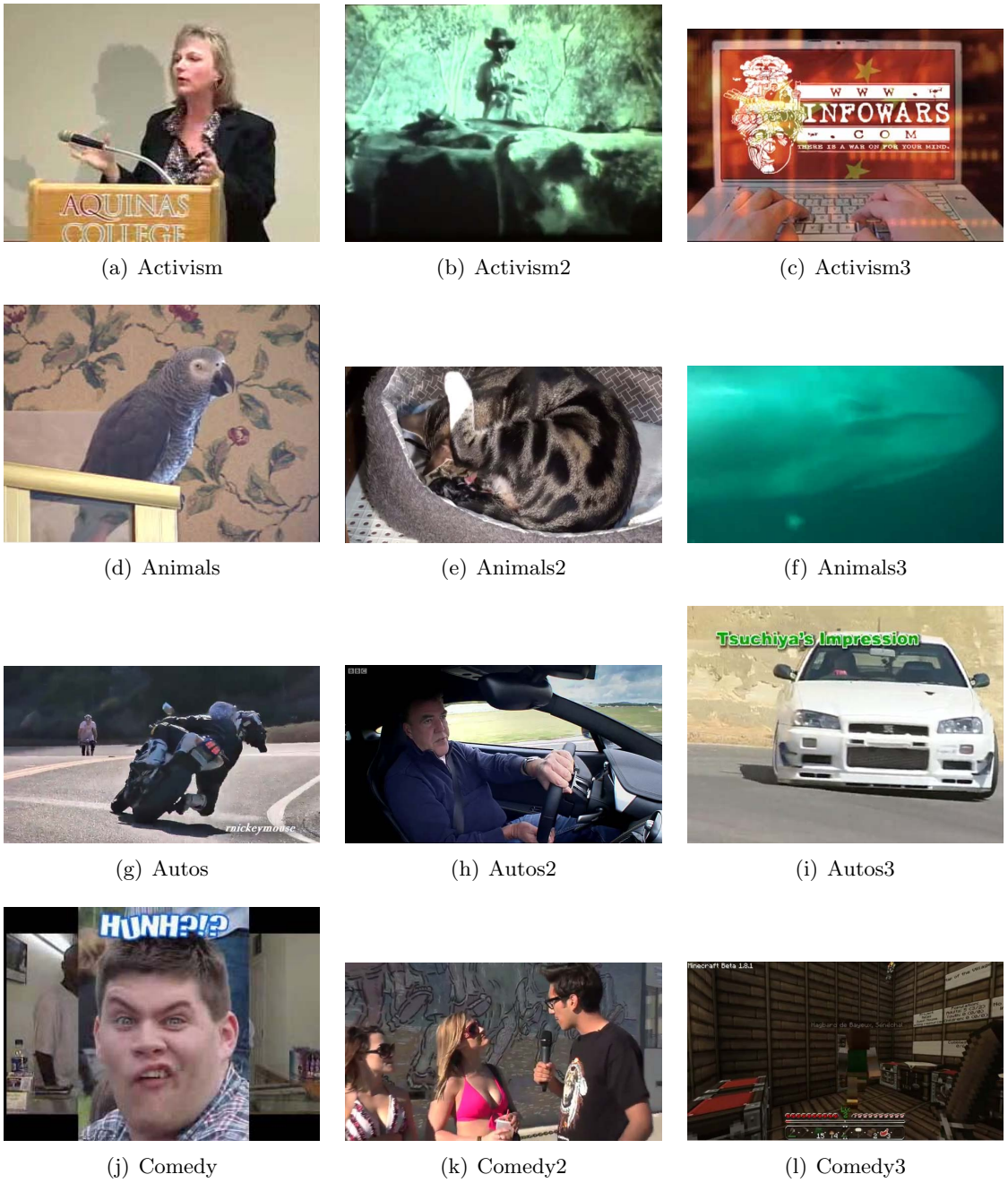
(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

Figure D.1.: Sample frames of the frontal face detection displaying some false positive and false negative detections. The blue rectangles are the MCT face detection, the red/yellow frames are from the Haar cascades.

E. YouTube samples



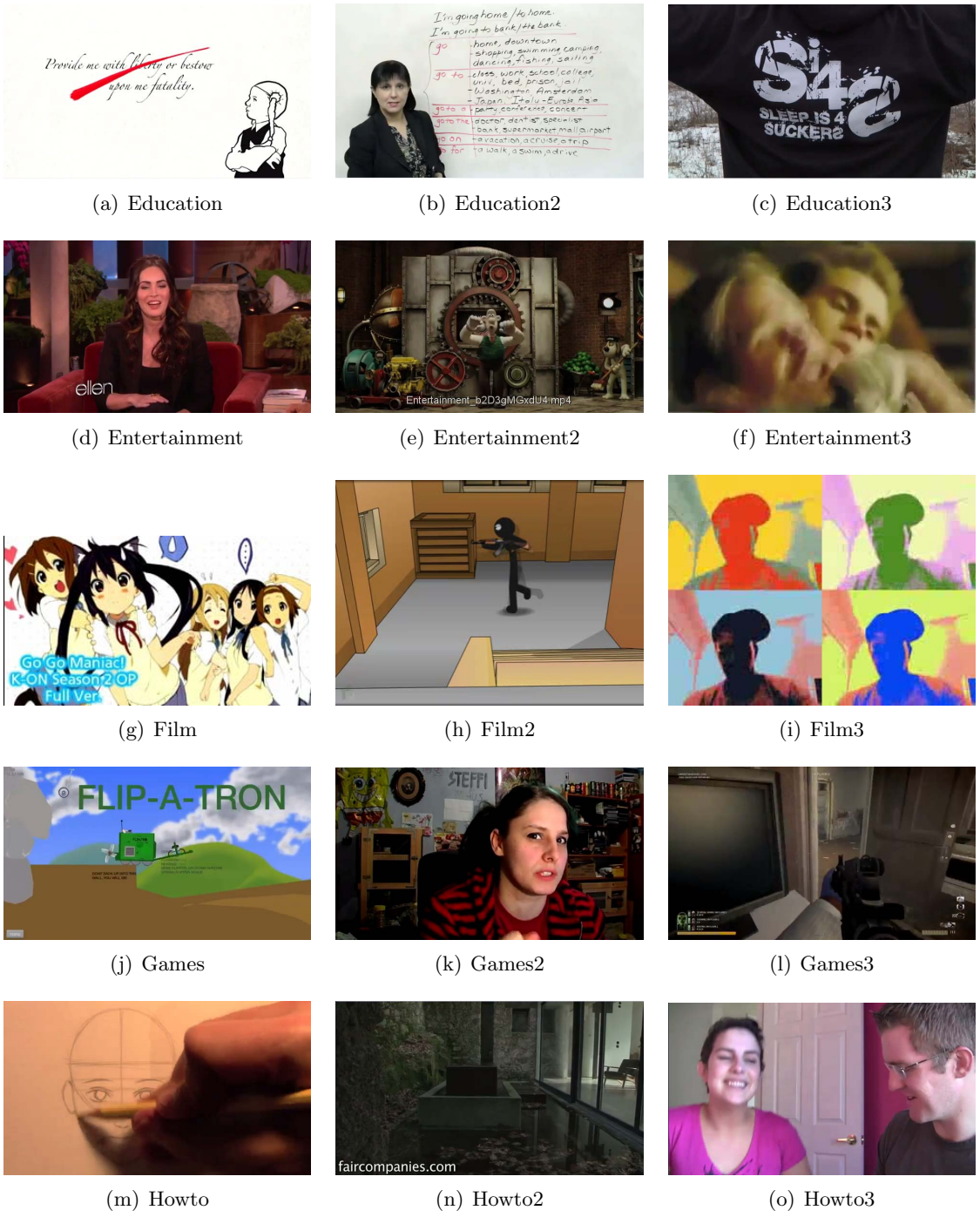
(a) Activism (b) Activism2 (c) Activism3

(d) Animals (e) Animals2 (f) Animals3

(g) Autos (h) Autos2 (i) Autos3

(j) Comedy (k) Comedy2 (l) Comedy3

Figure E.2.: Sample frames from the YouTube evaluation dataset showing the diversity in the single genres



(a) Education (b) Education2 (c) Education3

(d) Entertainment (e) Entertainment2 (f) Entertainment3

(g) Film (h) Film2 (i) Film3

(j) Games (k) Games2 (l) Games3

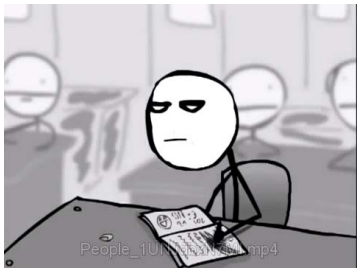
(m) Howto (n) Howto2 (o) Howto3

Figure E.3.: Sample frames from the YouTube evaluation dataset showing the diversity in the single genres

(a) News

(b) News2

(c) News3



(d) People

(e) People2

(f) People3

(g) Science

(h) Science2

(i) Science3

(j) Sports

(k) Sports2

(l) Sports3

(m) Travel

(n) Travel2

(o) Travel3

Figure E.4.: Sample frames from the YouTube evaluation dataset showing the diversity in the single genres

F. Other Result Tables

Table F.1.: Confusion matrix obtained on the RAI dataset using the baseline system and SVMs (%).

	Ca	Co	Fo	Mu	Ne	Ta	We
Ca	**100**	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0
Fo	0.0	0.0	**100**	0.0	0.0	0.0	0.0
Mu	0.0	14.2	0.0	**85.7**	0.0	0.0	0.0
Ne	0.0	0.0	0.0	0.0	**100**	0.0	0.0
Ta	0.0	0.0	0.0	0.0	0.0	**100**	0.0
We	0.0	0.0	0.0	0.0	0.0	0.0	**100**

Table F.2.: Confusion matrix obtained on the Quaero 2010 evaluation dataset using the baseline system and SVMs (%).

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We
Ca	**100**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Do	0.0	0.0	**75.0**	8.3	16.6	0.0	0.0	0.0	0.0	0.0
Ma	0.0	7.4	0.0	**85.2**	3.7	3.7	0.0	0.0	0.0	0.0
Mo	0.0	3.3	0.0	0.0	**96.7**	0.0	0.0	0.0	0.0	0.0
Ne	0.0	0.0	0.0	0.0	0	**100**	0.0	0.0	0.0	0.0
Sh	0.0	0.0	0.0	8.7	4.3	0.0	**87.0**	0.0	0.0	0.0
Ta	0.0	0.0	0.0	33.3	0	0.0	50.0	**16.6**	0.0	0.0
Tr	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	**100**	0.0
We	0.0	0.0	0.0	0.0	0	2.7	0.0	0.0	0.0	**97.3**

Table F.3.: Confusion matrix obtained on the RAI dataset using the extended system and Random Forests and choice parameter set (%).

	Ca	Co	Fo	Mu	Ne	Ta	We
Ca	**92.6**	3.7	0.0	0.0	3.7	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0
Fo	0.0	0.0	**95.5**	0.0	0.0	4.5	0.0
Mu	14.3	0.0	14.3	**14.3**	0.0	28.6	28.6
Ne	0.0	0.0	0.0	0.0	**95.9**	4.1	0.0
Ta	0.0	0.0	0.0	0.0	2.6	**94.7**	2.6
We	0.0	0.0	0.0	0.0	0.0	1.7	**98.3**

Table F.4.: Single feature accuracy on the RAI dataset using the extended system and Random Forest and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Fo	Mu	Ne	Ta	We	Avg
Aural	77.8	96.6	**100**	71.4	89.8	76.3	91.7	88.9
MFCC_2	37.0	89.7	**95.5**	0.0	79.6	31.6	95.0	73.2
MFCC_3	3.7	**94.8**	90.9	0.0	79.6	21.1	93.3	68.6
SP_2	18.5	**60.3**	54.5	0.0	40.8	26.3	58.3	44.8
SP_3	33.3	**63.8**	54.5	0.0	40.8	28.9	56.7	47.1
ZCR_2	18.5	69.0	9.1	0.0	65.3	18.4	**86.7**	52.9
ZCR_3	7.4	69.0	40.9	0.0	51.0	31.6	**78.3**	51.7
Auto Color	77.8	**100**	90.9	28.6	91.8	92.1	98.3	92.0
Color Mom	85.2	98.3	**100**	0.0	89.8	89.5	98.3	91.6
HSV Hist	70.4	**100**	95.5	57.1	**100**	94.7	98.3	**94.3**
CoOccurrence	70.4	**100**	31.8	28.6	81.6	68.4	91.7	79.3
Edge Hist	63.0	94.8	77.3	0.0	93.9	81.6	**96.7**	85.8
Wavelet	70.4	**100**	95.5	57.1	**100**	94.7	98.3	**94.3**
Struct	51.9	96.6	54.5	0.0	85.7	84.2	**98.3**	82.4
Haar Front	33.3	82.8	81.8	14.3	75.5	94.7	**100**	80.1
Haar + Profile	55.6	81.0	86.4	14.3	77.6	92.1	**100**	82.4
MCT Front	44.4	89.7	90.9	0.0	79.6	97.4	**100**	84.3
MCT + Side	48.1	89.7	90.9	0.0	77.6	94.7	**96.7**	83.1
SIFT 1x1	88.9	**100**	81.8	0.0	**100**	73.7	3.3	68.6
SIFT 2x2	63.0	**100**	81.8	0.0	**100**	65.8	5.0	65.1
rgbSIFT 1x1	88.9	**100**	86.4	0.0	**100**	81.6	16.7	73.2
rgbSIFT 2x2	81.5	**100**	81.8	0.0	**100**	71.1	6.7	68.2
oppSIFT 1x1	92.6	**100**	81.8	0.0	**100**	84.2	5.0	70.9
oppSIFT 2x2	74.1	**100**	81.8	0.0	**100**	81.6	1.7	67.8

Table F.5.: Confusion matrix obtained on the RAI dataset using the extended system and Decision Trees and choice parameter set (%).

	Ca	Co	Fo	Mu	Ne	Ta	We
Ca	**92.6**	0.0	0.0	0.0	7.4	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0
Fo	0.0	0.0	**100**	0.0	0.0	0.0	0.0
Mu	0.0	0.0	14.3	**28.6**	28.6	0.0	28.6
Ne	0.0	0.0	0.0	0.0	**98.0**	2.0	0.0
Ta	0.0	0.0	0.0	0.0	2.6	**97.4**	0.0
We	0.0	0.0	0.0	0.0	0.0	1.7	**98.3**

Table F.6.: Single feature accuracy on the RAI dataset using the extended system and Decision Tree and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Fo	Mu	Ne	Ta	We	Avg
Aural	70.4	**91.4**	77.3	57.1	75.5	65.8	76.7	77.0
MFCC_2	25.9	**44.8**	54.5	0.0	30.6	34.2	43.3	37.9
MFCC_3	18.5	56.9	**63.6**	14.3	40.8	10.5	45.0	39.8
SP_2	40.7	27.6	**54.5**	0.0	20.4	26.3	31.7	29.9
SP_3	18.5	32.8	40.9	14.3	36.7	31.6	**43.3**	34.5
ZCR_2	48.1	31.0	9.1	0.0	40.8	18.4	**73.3**	39.8
ZCR_3	14.8	41.4	36.4	0.0	46.9	42.1	**53.3**	41.0
Auto Color	66.7	**100**	81.8	28.6	85.7	89.5	98.3	**88.5**
Color Mom	63.0	98.3	**100**	14.3	93.9	76.3	90.0	86.6
HSV Hist	66.7	**100**	81.8	28.6	95.9	76.3	96.7	88.1
CoOccurrence	66.7	**100**	68.2	28.6	69.4	60.5	88.3	77.8
Edge Hist	66.7	**94.8**	68.2	0.0	85.7	73.7	78.3	78.5
Wavelet	74.1	**98.3**	77.3	14.3	83.7	68.4	95.0	83.9
Struct	55.6	**96.6**	54.5	0.0	63.3	73.7	88.3	74.7
Haar Front	40.7	63.8	68.2	14.3	61.2	76.3	**95.0**	69.0
Haar + Profile	37.0	65.5	81.8	28.6	61.2	76.3	**90.0**	69.3
MCT Front	40.7	70.7	77.3	14.3	53.1	86.8	**88.3**	69.7
MCT + Side	51.9	69.0	81.8	14.3	75.5	81.6	**96.7**	76.2
SIFT 1x1	81.5	**100**	86.4	0.0	**100**	81.6	15.0	72.0
SIFT 2x2	74.1	**100**	86.4	0.0	**100**	78.9	5.0	68.6
rgbSIFT 1x1	92.6	**100**	86.4	0.0	**100**	76.30	25.0	74.7
rgbSIFT 2x2	74.1	**100**	86.4	0.0	98.0	84.2	13.3	70.9
oppSIFT 1x1	77.8	**100**	90.9	14.3	**100**	78.9	16.7	72.4
oppSIFT 2x2	74.1	**100**	81.8	0.0	**100**	84.2	16.7	71.6

Table F.7.: Confusion matrix obtained on the Quaero 2010 evaluation dataset using the extended system and Random Forests and choice parameter set (%).

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We
Ca	**66.7**	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**99.2**	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.8
Do	0.0	8.3	**75.0**	0.0	16.7	0.0	0.0	0.0	0.0	0.0
Ma	0.0	7.4	0.0	**77.8**	7.4	0.0	3.7	0.0	0.0	3.7
Mo	0.0	6.7	3.3	0.0	**86.7**	0.0	0.0	0.0	0.0	0.0
Ne	0.0	0.0	0.0	0.0	0.0	**100**	0.0	0.0	5.6	0.0
Sh	0.0	4.3	0.0	4.3	4.3	0.0	**87.0**	0.0	0.0	0.0
Ta	0.0	0.0	0.0	33.3	16.7	0.0	50.0	**0.0**	0.0	0.0
Tr	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**94.4**	0.0
We	0.0	0.0	0.0	0.0	0.0	2.7	0.0	0.0	0.0	**97.3**

Table F.8.: Confusion matrix obtained on the Quaero 2010 evaluation dataset using the extended system and Decision Trees and choice parameter set (%).

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We
Ca	**66.7**	33.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Co	0.0	**100**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8
Do	0.0	0.0	**83.3**	0.0	16.7	0.0	0.0	0.0	0.0	0.0
Ma	0.0	14.8	0.0	**70.4**	3.7	0.0	7.4	0.0	0.0	3.7
Mo	0.0	3.3	3.3	0.0	**90.0**	0.0	0.0	0.0	0.0	0.0
Ne	0.0	0.0	0.0	0.0	0.0	**100**	0.0	0.0	5.6	0.0
Sh	0.0	4.3	0.0	0.0	4.3	0.0	**87.0**	4.3	0.0	0.0
Ta	0.0	0.0	0.0	33.3	16.7	0.0	33.3	**16.7**	0.0	0.0
Tr	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**94.4**	0.0
We	0.0	0.0	0.0	2.7	0.0	2.7	0.0	0.0	0.0	**94.6**

Table F.9.: Single feature accuracy on the Quaero dataset using the extended system and Random Forest and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We	Avg
Aural	33.3	**100**	41.7	74.1	93.3	94.4	**100**	33.3	94.4	91.9	**91.0**
MFCC_2	0.0	**100**	8.3	7.4	16.7	5.6	17.4	0.0	0.0	91.9	58.0
MFCC_3	0.0	**100**	0.0	7.4	3.3	0.0	0.0	0.0	0.0	81.1	53.3
SP_2	0.0	**99.2**	0.0	3.7	3.3	0.0	0.0	0.0	0.0	21.6	45.0
SP_3	0.0	**96.0**	0.0	14.8	36.7	11.1	0.0	0.0	11.1	8.1	47.7
ZCR_2	0.0	**99.2**	0.0	0.0	10.0	5.6	0.0	16.7	0.0	16.2	45.3
ZCR_3	0.0	**93.7**	8.3	14.8	0.0	5.6	0.0	0.0	5.6	24.3	44.7
Auto Color	66.7	**98.4**	41.7	66.7	83.3	83.3	91.3	16.7	88.9	97.3	87.7
Color Mom	0.0	96.0	58.3	70.4	70.0	**100**	87.0	0.0	88.9	94.6	85.7
HSV Hist	**100**	98.4	41.7	70.4	83.3	88.9	95.7	33.3	94.4	97.3	89.7
CoOccurrence	66.7	**95.2**	50.0	48.1	63.3	88.9	30.4	0.0	66.7	91.9	76.3
Edge Hist	66.7	97.6	58.3	59.3	73.3	**100**	65.2	0.0	88.9	94.6	84.7
Wavelet	66.7	97.6	58.3	51.9	86.7	**100**	91.3	0.0	94.4	94.6	87.7
Struct	0.0	**98.4**	50.0	40.7	53.3	77.8	56.5	0.0	88.9	91.9	78.0
Haar Front	0.0	**95.2**	0.0	33.3	83.3	44.4	60.9	33.3	38.9	70.3	70.3
Haar + Profile	0.0	**92.9**	0.0	55.6	80.0	50.0	65.2	33.3	61.1	81.1	74.3
MCT Front	0.0	**93.7**	8.3	40.7	76.7	38.9	69.6	16.7	38.9	83.8	71.7
MCT + Side	0.0	**95.2**	0.0	55.6	83.3	72.2	69.6	0.0	50.0	89.2	77.0
SIFT 1x1	**100**	**100**	91.7	3.7	60.0	0.0	91.3	0.0	77.8	29.7	68.3
SIFT 2x2	**100**	**100**	91.7	29.6	56.7	11.1	91.3	0.0	83.3	27.0	71.0
rgbSIFT 1x1	**100**	**100**	66.7	7.4	30.0	0.0	91.3	0.0	94.4	48.6	68.0
rgbSIFT 2x2	66.7	**100**	91.7	29.6	63.3	16.7	87.0	0.0	83.3	35.1	72.3
oppSIFT 1x1	33.3	**100**	58.3	0.0	33.3	0.0	78.3	0.0	88.9	35.1	63.7
oppSIFT 2x2	0.0	**100**	66.7	0.0	26.7	100	56.5	0.0	88.9	29.7	66.7

Table F.10.: Single feature accuracy on the Quaero dataset using the extended system and Decision Trees and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ca	Co	Do	Ma	Mo	Ne	Sh	Ta	Tr	We	Avg
Aural	66.7	**95.2**	16.7	37.0	80.0	61.1	78.3	33.3	50.0	70.3	74.7
MFCC_2	0.0	**65.1**	8.3	11.1	40.0	11.1	17.4	0.0	22.2	45.9	41.7
MFCC_3	0.0	**66.7**	8.3	18.5	26.7	11.1	4.3	0.0	5.6	37.8	38.7
SP_2	0.0	**50.0**	8.3	18.5	16.7	11.1	8.7	0.0	11.1	21.6	29.3
SP_3	0.0	**48.4**	0.0	11.1	36.7	5.6	0.0	0.0	22.2	21.6	29.3
ZCR_2	0.0	**47.6**	25.0	18.5	16.7	22.2	4.3	16.7	5.6	35.1	31.0
ZCR_3	0.0	**61.1**	8.3	7.4	10.0	27.8	13.0	0.0	0.0	16.2	32.3
Auto Color	66.7	**92.1**	16.7	37.0	60.0	61.1	82.6	0.0	72.2	83.8	74.0
Color Mom	33.3	94.4	50.0	66.7	56.7	88.9	65.2	16.7	77.8	**94.6**	80.7
HSV Hist	66.7	**91.3**	25.0	29.6	60.0	44.4	56.5	16.7	83.3	81.1	71.0
CoOccurrence	33.3	**92.1**	33.3	44.4	50.0	66.7	39.1	0.0	66.7	89.2	71.3
Edge Hist	0.0	87.3	66.7	66.7	56.7	83.3	47.8	16.7	83.3	**91.9**	76.3
Wavelet	33.3	**92.1**	50.0	48.1	56.7	94.4	73.9	33.3	66.7	75.7	76.3
Struct	66.7	**96.0**	41.7	33.3	36.7	66.7	43.5	0.0	66.7	89.2	71.7
Haar Front	33.3	**65.9**	8.3	33.3	56.7	38.9	39.1	50.0	27.8	56.8	52.0
Haar + Profile	33.3	**78.6**	8.3	37.0	70.0	50.0	60.9	33.3	61.1	59.5	63.3
MCT Front	0.0	**77.8**	33.3	48.1	56.7	27.8	56.5	33.3	44.4	70.3	62.0
MCT + Side	33.3	**77.8**	25.0	40.7	73.3	33.3	73.9	33.3	55.6	59.5	64.0
SIFT 1x1	**100**	**100**	91.7	37.0	60.0	44.4	91.3	16.7	66.7	51.4	76.3
SIFT 2x2	**100**	**100**	91.7	40.7	80.0	**100**	95.7	16.7	72.2	54.1	**83.0**
rgbSIFT 1x1	**100**	**100**	91.7	55.6	63.3	61.1	82.6	0.0	88.9	48.6	79.3
rgbSIFT 2x2	66.7	**100**	91.7	48.1	63.3	100	**100**	16.7	94.4	48.6	82.7
oppSIFT 1x1	**100**	**100**	66.7	18.5	30.0	**100**	95.7	0.0	77.8	59.5	75.7
oppSIFT 2x2	33.3	**100**	91.7	44.4	60.0	**100**	87.0	0.0	77.8	37.8	78.0

Table F.11.: Confusion matrix obtained on the YouTube evaluation dataset and Random Forests (%).

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr
Ac	**24.0**	12.0	12.0	4.0	6.0	8.0	8.0	6.0	10.0	2.0	0.0	4.0	0.0	4.0
An	8.0	**42.0**	2.0	2.0	4.0	2.0	2.0	6.0	14.0	4.0	2.0	2.0	0.0	10.0
Au	8.0	8.0	**70.0**	0.0	2.0	0.0	2.0	2.0	2.0	2.0	0.0	0.0	0.0	4.0
Co	2.0	6.0	2.0	**38.0**	2.0	18.0	4.0	8.0	10.0	0.0	2.0	2.0	0.0	6.0
Ed	8.0	10.0	4.0	4.0	**36.0**	2.0	2.0	4.0	10.0	6.0	2.0	4.0	0.0	8.0
En	2.0	0.0	2.0	2.0	0.0	**74.0**	6.0	2.0	2.0	4.0	0.0	4.0	2.0	0.0
Fi	8.0	6.0	2.0	14.0	4.0	12.0	**26.0**	14.0	4.0	2.0	2.0	4.0	0.0	2.0
Ga	4.0	0.0	2.0	4.0	2.0	6.0	10.0	**50.0**	2.0	4.0	2.0	0.0	8.0	6.0
Ho	4.0	8.0	2.0	0.0	4.0	4.0	0.0	0.0	**70.0**	2.0	2.0	2.0	0.0	2.0
Ne	4.0	2.0	6.0	2.0	8.0	8.0	6.0	6.0	8.0	**30.0**	4.0	4.0	0.0	12.0
Pe	6.0	8.0	2.0	12.0	4.0	14.0	8.0	4.0	12.0	8.0	**14.0**	0.0	4.0	4.0
Sc	8.0	8.0	0.0	8.0	2.0	18.0	2.0	6.0	2.0	4.0	0.0	**32.0**	4.0	6.0
Sp	4.0	16.0	10.0	6.0	0.0	8.0	2.0	16.0	0.0	2.0	2.0	2.0	**30.0**	2.0
Tr	8.0	8.0	12.0	6.0	2.0	8.0	6.0	12.0	8.0	0.0	2.0	4.0	2.0	**22.0**

Table F.12.: Confusion matrix obtained on the YouTube evaluation dataset and Decision Trees (%).

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr
Ac	**12.0**	18.0	4.0	10.0	12.0	6.0	12.0	8.0	2.0	2.0	0.0	2.0	4.0	8.0
An	2.0	**30.0**	10.0	4.0	8.0	0.0	8.0	6.0	2.0	4.0	0.0	4.0	10.0	12.0
Au	4.0	4.0	**46.0**	2.0	6.0	0.0	8.0	6.0	2.0	6.0	0.0	2.0	2.0	12.0
Co	4.0	0.0	2.0	**40.0**	2.0	8.0	8.0	10.0	4.0	2.0	6.0	2.0	6.0	6.0
Ed	10.0	4.0	6.0	2.0	**36.0**	2.0	2.0	4.0	2.0	2.0	2.0	10.0	2.0	16.0
En	0.0	0.0	0.0	6.0	2.0	**48.0**	10.0	12.0	2.0	4.0	2.0	6.0	0.0	8.0
Fi	1.0	6.0	4.0	2.0	4.0	12.0	**24.0**	20.0	2.0	6.0	2.0	0.0	4.0	10.0
Ga	0.0	6.0	4.0	4.0	4.0	10.0	24.0	**28.0**	0.0	4.0	2.0	6.0	4.0	4.0
Ho	6.0	4.0	0.0	4.0	2.0	6.0	2.0	0.0	**60.0**	4.0	4.0	0.0	2.0	6.0
Ne	4.0	4.0	8.0	2.0	6.0	6.0	6.0	16.0	6.0	**18.0**	4.0	6.0	2.0	12.0
Pe	12.0	12.0	4.0	14.0	4.0	10.0	6.0	6.0	0.0	0.0	**12.0**	2.0	2.0	16.0
Sc	0.0	12.0	4.0	10.0	12.0	8.0	4.0	6.0	0.0	4.0	0.0	**18.0**	8.0	14.0
Sp	8.0	8.0	14.0	10.0	6.0	4.0	8.0	10.0	0.0	0.0	6.0	2.0	**16.0**	8.0
Tr	10.0	6.0	10.0	4.0	2.0	2.0	4.0	12.0	4.0	4.0	0.0	6.0	4.0	**32.0**

Table F.13.: Single feature accuracy on the YouTube dataset using the extended system and Random Forest and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr	Avg
Aural	4.0	8.0	6.0	2.0	4.0	6.0	2.0	10.0	**12.0**	2.0	4.0	6.0	6.0	4.0	5.4
MFCC_2	4.0	4.0	8.0	4.0	2.0	0.0	2.0	6.0	4.0	4.0	8.0	8.0	16.0	**18.0**	6.3
MFCC_3	2.0	4.0	2.0	8.0	14.0	10.0	8.0	8.0	12.0	2.0	6.0	8.0	4.0	**16.0**	7.4
SP_2	6.0	6.0	8.0	4.0	**18.0**	8.0	4.0	2.0	12.0	6.0	2.0	**18.0**	**18.0**	8.0	8.6
SP_3	4.0	**14.0**	6.0	**14.0**	8.0	4.0	0.0	4.0	8.0	12.0	8.0	8.0	8.0	2.0	7.1
ZCR_2	4.0	6.0	8.0	8.0	8.0	2.0	2.0	6.0	4.0	**12.0**	6.0	2.0	6.0	4.0	5.6
ZCR_3	8.0	6.0	4.0	4.0	0.0	**16.0**	14.0	4.0	6.0	6.0	6.0	8.0	12.0	8.0	7.3
Auto Color	10.0	24.0	32.0	38.0	30.0	52.0	20.0	32.0	**62.0**	30.0	6.0	20.0	34.0	16.0	29.0
Color Mom	24.0	38.0	40.0	28.0	38.0	54.0	4.0	34.0	**72.0**	28.0	14.0	26.0	18.0	18.0	31.1
HSV Hist	14.0	32.0	36.0	36.0	34.0	52.0	24.0	32.0	**62.0**	32.0	4.0	26.0	32.0	20.0	31.1
CoOccurrence	10.0	10.0	20.0	24.0	24.0	26.0	16.0	10.0	**48.0**	28.0	10.0	24.0	18.0	14.0	20.1
Edge Hist	22.0	22.0	52.0	28.0	30.0	46.0	8.0	32.0	**62.0**	26.0	10.0	26.0	24.0	18.0	29.0
Wavelet	42.0	18.0	56.0	28.0	34.0	46.0	26.0	36.0	**66.0**	18.0	24.0	34.0	28.0	20.0	34.0
Struct	6.0	14.0	**40.0**	14.0	14.0	24.0	10.0	28.0	34.0	22.0	26.0	18.0	14.0	2.0	19.0
Haar Front	10.0	20.0	28.0	28.0	10.0	**42.0**	20.0	14.0	28.0	22.0	14.0	8.0	16.0	4.0	18.9
Haar + Profile	2.0	22.0	36.0	28.0	10.0	**46.0**	16.0	26.0	24.0	20.0	18.0	12.0	16.0	2.0	19.9
MCT Front	4.0	18.0	**34.0**	32.0	16.0	30.0	14.0	28.0	28.0	22.0	16.0	10.0	26.0	4.0	20.1
MCT + Side	12.0	18.0	34.0	26.0	16.0	30.0	14.0	22.0	**38.0**	16.0	14.0	14.0	22.0	2.0	19.9
SIFT 1x1	22.0	32.0	**80.0**	40.0	24.0	70.0	28.0	52.0	62.0	20.0	8.0	16.0	16.0	6.0	34.0
SIFT 2x2	20.0	28.0	**84.0**	36.0	34.0	58.0	32.0	46.0	52.0	20.0	8.0	16.0	8.0	18.0	32.9
rgbSIFT 1x1	28.0	36.0	**84.0**	42.0	32.0	66.0	26.0	48.0	66.0	20.0	6.0	12.0	20.0	18.0	**36.0**
rgbSIFT 2x2	16.0	28.0	**76.0**	40.0	30.0	62.0	28.0	46.0	58.0	18.0	8.0	20.0	6.0	16.0	32.3
oppSIFT 1x1	22.0	42.0	**82.0**	40.0	32.0	70.0	30.0	46.0	56.0	18.0	6.0	18.0	26.0	14.0	35.9
oppSIFT 2x2	16.0	24.0	**82.0**	40.0	36.0	60.0	26.0	52.0	58.0	18.0	6.0	16.0	10.0	12.0	32.6

Table F.14.: Single feature accuracy on the YouTube dataset using the extended system and Decision Tree and choice parameter set (%). Bold values display the best genre for each feature. Values with a box display the best feature for each genre.

	Ac	An	Au	Co	Ed	En	Fi	Ga	Ho	Ne	Pe	Sc	Sp	Tr	Avg
Aural	10.0	4.0	10.0	**12.0**	4.0	6.0	0.0	6.0	4.0	8.0	6.0	4.0	4.0	2.0	5.7
MFCC_2	4.0	8.0	12.0	4.0	6.0	4.0	8.0	2.0	4.0	8.0	4.0	**14.0**	4.0	8.0	6.4
MFCC_3	4.0	4.0	8.0	4.0	6.0	6.0	2.0	**14.0**	2.0	12.0	4.0	8.0	6.0	6.0	6.1
SP_2	10.0	4.0	8.0	2.0	**12.0**	8.0	6.0	8.0	6.0	0.0	6.0	6.0	4.0	8.0	6.3
SP_3	4.0	**12.0**	2.0	4.0	8.0	2.0	4.0	6.0	10.0	10.0	6.0	4.0	10.0	0.0	5.9
ZCR_2	4.0	6.0	4.0	10.0	4.0	4.0	0.0	12.0	2.0	8.0	8.0	**14.0**	6.0	4.0	6.1
ZCR_3	8.0	6.0	**12.0**	10.0	4.0	**12.0**	4.0	8.0	4.0	2.0	8.0	8.0	4.0	2.0	6.6
Auto Color	8.0	6.0	18.0	28.0	18.0	22.0	28.0	18.0	**44.0**	16.0	10.0	14.0	14.0	12.0	18.3
Color Mom	6.0	14.0	12.0	24.0	26.0	26.0	12.0	10.0	**40.0**	16.0	10.0	10.0	10.0	10.0	16.1
HSV Hist	8.0	16.0	26.0	20.0	16.0	**32.0**	18.0	12.0	**32.0**	16.0	12.0	14.0	24.0	14.0	18.6
CoOccurrence	6.0	12.0	22.0	16.0	12.0	26.0	12.0	18.0	**34.0**	10.0	10.0	24.0	14.0	8.0	16.0
Edge Hist	16.0	14.0	24.0	16.0	8.0	20.0	8.0	8.0	**28.0**	12.0	10.0	14.0	22.0	18.0	15.6
Wavelet	20.0	10.0	24.0	26.0	20.0	20.0	18.0	14.0	**34.0**	18.0	8.0	24.0	8.0	16.0	18.6
Struct	14.0	12.0	18.0	10.0	22.0	6.0	6.0	16.0	20.0	12.0	**24.0**	12.0	14.0	4.0	13.6
Haar Front	2.0	18.0	22.0	16.0	12.0	**20.0**	12.0	16.0	14.0	12.0	8.0	14.0	12.0	2.0	12.9
Haar + Profile	6.0	18.0	18.0	6.0	12.0	**20.0**	10.0	**20.0**	14.0	18.0	18.0	12.0	18.0	8.0	14.1
MCT Front	8.0	14.0	16.0	18.0	12.0	**22.0**	4.0	16.0	20.0	10.0	10.0	10.0	12.0	0.0	12.3
MCT + Side	8.0	22.0	18.0	14.0	18.0	24.0	10.0	8.0	**28.0**	16.0	6.0	16.0	14.0	4.0	14.7
SIFT 1x1	12.0	14.0	**58.0**	30.0	24.0	44.0	24.0	34.0	48.0	20.0	10.0	8.0	10.0	18.0	25.3
SIFT 2x2	8.0	**52.0**	40.0	26.0	18.0	50.0	4.0	14.0	46.0	16.0	4.0	4.0	6.0	36.0	23.1
rgbSIFT 1x1	12.0	24.0	**60.0**	26.0	26.0	42.0	12.0	32.0	52.0	22.0	6.0	24.0	18.0	12.0	**26.3**
rgbSIFT 2x2	6.0	4.0	**62.0**	24.0	18.0	24.0	12.0	12.0	50.0	26.0	2.0	14.0	10.0	44.0	22.0
oppSIFT 1x1	24.0	18.0	**56.0**	36.0	12.0	54.0	20.0	20.0	48.0	14.0	4.0	12.0	16.0	16.0	25.0
oppSIFT 2x2	14.0	22.0	22.0	36.0	14.0	26.0	26.0	24.0	**40.0**	18.0	0.0	0.0	2.0	22.0	19.0

G. Sample Decision Tree

Decision Tree structure of the *AutoColorCorrelogram* feature for Decision Tree classifier for the RAI dataset. The split–node numbers are the features from the feature vector responsible for the split. The number in the brackets are the number of samples arriving at the particular node. The numbers in the boxed leafs are the class label outputs when a sample is predicted.