UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Multimodal summarization of user-generated videos from wearable cameras

by

Theodoros Psallidas

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Theodoros Giannakopoulos
Principal Researcher of Multimodal Machine Learning

Athens, May 2021

Multimodal summarization of user-generated videos from wearable cameras

Theodoros Psallidas

MSc. Thesis, MSc. Programme in Data Science

University of the Peloponnese & NCSR "Democritos", May 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Multimodal summarization of user-generated videos from wearable cameras

by

Theodoros Psallidas

A thesis submitted in partial fulfillment
of the requirements for the MSc
in Data Science

Supervisor: Theodoros Giannakopoulos
Principal Researcher of Multimodal Machine Learning

Approved by the examination committee on May, 2021.

(Signature)	(Signature)	(Signature)
.....	.....	.....
Theodoros Giannakopoulos Principal Researcher	Charilaos Akasiadis Postdoctoral Researcher	Nikolaos Platis Assistant Professor

Athens, May 2021

UNIVERSITY OF THE PELOPONNESE & NCSR "DEMOCRITOS"
MSC PROGRAMME IN DATA SCIENCE

Declaration of Authorship

(1) I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

(2) I confirm that this thesis presented for the degree of Bachelor of Science in Informatics and Telecommunications, has

(i) been composed entirely by myself

(ii) been solely the result of my own work

(iii) not been submitted for any other degree or professional qualification

(3) I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or processional qualification except as specified.

(Signature)

. . . . . .

Theodoros Psallidas

Athens, May 2021

Acknowledgments

First of all, I would like to thank Dr.Giannakopoulos Theodoros, who trusted me and guided me throughout the elaboration of my dissertation, within the postgraduate program of Data Science of the University of Peloponnese in collaboration with the National Centre for Scientific Research "Demokritos". He constantly encouraged me and through the close cooperation we had, he helped me to develop both personally and professionally. I would also like to thank Computing Intelligence Lab for providing knowledge and support.

Finally, I would like to thank all my friends and relatives, who supported me while I was preparing the dissertation.

To my family and friends.

Περίληψη

Σκοπός αυτής της εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικασίας σύνοψης ενός βίντεο σε μια πιο σύντομη και ταυτόχρονα περιεκτική μορφή. Η εκθετική αύξηση του περιεχομένου που δημιουργείται από τον χρήστη έχει αυξήσει την ανάγκη αποτελεσματικών προγραμμάτων δημιουργίας συνοπτικών βίντεο. Ωστόσο, οι περισσότερες προσεγγίσεις υποτιμούν τη δύναμη των ακουστικών χαρακτηριστικών, ενώ έχουν σχεδιαστεί για να λειτουργούν κυρίως σε εμπορικά / επαγγελματικά βίντεο. Σε αυτήν την εργασία παρουσιάζεται μια προσέγγιση που χρησιμοποιεί τόσο ηχητικά όσο και οπτικά χαρακτηριστικά, προκειμένου να δημιουργήσει συνοπτικά βίντεο από βίντεο που δημιουργούνται από χρήστες. Η συγκεκριμένη προσέγγιση παράγει δυναμικές περιλήψεις βίντεο, δηλαδή, περιλήψεις που περιλαμβάνουν τα πιο «σημαντικά» μέρη του αρχικού βίντεο τα οποία είναι διατεταγμένα έτσι ώστε να διατηρούν τη χρονική τους σειρά. Χρησιμοποιείται εποπτευόμενη γνώση και από τις δύο προαναφερθείσες μεθόδους εκπαιδεύοντας έναν δυαδικό αλγόριθμο πρόβλεψης, ο οποίος μαθαίνει να αναγνωρίζει τα σημαντικά μέρη των βίντεο. Επιπλέον, παρουσιάζεται ένα νέο σύνολο δεδομένων που δημιουργείται από τον χρήστη και περιέχει βίντεο από διάφορες κατηγορίες. Κάθε τμήμα 1 δευτερολέπτου για κάθε βίντεο του συνόλου δεδομένων έχει επισημανθεί από περισσότερους από τρεις σχολιαστές ως σημαντικό ή όχι. Αξιολογείται η προσέγγιση της παρούσας διπλωματικής χρησιμοποιώντας διάφορες στρατηγικές ταξινόμησης αξιοποιώντας πηγές ήχου, βίντεο και συνδυασμό αυτών. Τα πειραματικά αποτελέσματα δείχνουν τις δυνατότητες της προτεινόμενης προσέγγισης.

Abstract

The aim of this thesis is to construct a video summarization procedure to distill a video sequence in a more compact, and at the same time, informative form. The exponential growth of user-generated content has increased the need for efficient video summarization schemes. However, most approaches underestimate the power of aural features, while they are designed to work mainly on commercial/professional videos. In this work, we present an approach that uses both audio and visual features, in order to create video summaries from user-generated videos. Our approach produces dynamic video summaries, i.e., comprising of the most "important" parts of the original video, which are arranged so as to preserve their temporal order. We use supervised knowledge from both the aforementioned modalities and train a binary classifier, which learns to recognize the important parts of videos. Moreover, we present a novel user-generated dataset which contains videos from several categories. Every 1-sec part of each video from our dataset has been annotated by more than three annotators as being important or not. We evaluate our approach using several classification strategies based on audio, video, and fused features. Our experimental results illustarte the potential of our approach.

Contents

List of Tables

List of Figures

List of Abbreviations

i.e.	id est
UAVs	Unmanned Air Vehicles
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
GAN	Generative Adversarial Network
VAT	Video Annotator Tool
Log Res	Logistic Regression
KNN	K-Nearest Neighbor
AUC	Area Under Curve
ROC	Receiver Operating Characteristic
MFCCs	Mel Frequency Cepstral Coefficients
RGB	Red Green Blue
XGBoost	eXtreme Gradient Boosting
RFE	Recursive Feature Elimination

Chapter 1

Introduction

1.1 Problem description

Recent advances in the fields of digital imaging and electronics have allowed the integration of high definition optical sensors into affordable mobile phones and action cameras. Consequently, we have witnessed an exponential growth of user-generated video content. Moreover, a continuously increasing number of users tend to capture their daily life and other activities such as sports, trips etc. and share them via social networks such as Facebook (`http://www.facebook.com`), Instagram (`http://www.instagram.com`), Twitter (`http://www.twitter.com`) etc., or video-sharing platforms such as YouTube (`http://www.youtube.com`), Vimeo (`http://www.vimeo.com`) etc. According to the official YouTube statistics [1], every day its users watch over 1 billion hours of video and they also upload more than 500 hours of video every minute. It is generally anticipated that the aforementioned numbers will be constantly growing for the next years.

As the amount of available information grows, the amount of assistance that users need to efficiently browse huge video collections [2] and to derive useful information within large videos is rising. To fulfil the aformentioned ever-growing needs, many research efforts have shifted towards the task of *video summarization*. In brief, this task aims to create a condensed version of a given video sequence [3]; when a user watches this version, she/he should immediately capture the most im-

portant parts of the video's content. Apart from efficient browsing and retrieval of videos for entertainment purposes [4], applications of video summarization include summarization of surveillance videos [5], medical videos [6], large videos captured by unmanned aired vehicles (UAVs) [7] etc.

Approaches on video summarization may be classified into four major categories, based on the type of audiovisual visual cues produced and presented to their end user [3], i.e., their output. More specifically, output of video summarization algorithms may consist of a) keyframes [8], which are extracted video frames, presented in order and are often denoted as "static" summaries; b) (a set of) video segments [9], which are often denoted as "dynamic" summaries and consist an obvious extension of keyframes and preserve both audio and motion of videos; c) graphical cues [10] which complement other cues by some type of graphical-based syntax to further enhance the interpretation of summaries by the end user; and d) automatically generated textual annotations [11] which aim to provide efficient summaries of video content.

In this work, we propose adopting a supervised video summarization technique to produce short dynamic summaries for user-generated videos. Our approach belongs to a sub-category often stated as *video skimming* [12, 13]. Such approaches are based on uni- or multi-modal features, extracted from the video. Their output consist of parts of the original video that have been selected as significant, while preserving their temporal order. As presented by Sen and Raman [13], video skimming summarization approaches allow for better understanding of the original video by end users, based solely on its summary. Therefore, such approaches have recently drawn increased attention within the research community.

More specifically, in this work we propose the use of supervised knowledge from both audio and visual domains, to achieve summarizations of user-generated videos. In particular, we analyze a given video stream by splitting it into one-second segments of audio and visual representations. Segments are either classified as being "informative" (i.e., adequately "interesting" so that they should be used within the final video summary) or "uninteresting" (i.e., not containing information that should be included within the final video summary). We use a supervised binary classifier trained on the feature representations of either audio, video, or fused modalities.

Moreover, we present a novel dataset, comprising of user-generated videos, collected from YouTube, that have been recorded using either action cameras or smartphones. In this work, we also present in detail the way the collective annotation process has been carried out by a group of human annotators, using an annotating tool which was created for the purposes of this work.

1.2 Thesis structure

The rest of the thesis is organised as follows:

In Chapter 2 we present and discuss related, state-of-the-art research works in video summarization.

In Chapter 3 we formulate the problem at hand, and discuss the approach we followed to collect and process the dataset we have used for the experimental evaluation.

In Chapter 4 we describe in detail the theoretical background of the machine learning and the classification algorithms we used.

In Chapter 5 we describe in detail the proposed multi-modal video summarization methodology.

In Chapter 6 we present the results and discuss our experiments. Furthermore, we highlight some key-points of our methodology.

In Chapter 7 we sum up our challenges and findings, and we discuss future work, focusing on how our own contributions could build up to a large scale data set that could handle better the task of video summarization combing various modalities.

Chapter 2

Related Work

2.1 Approaches for video summarization

During the last few years, a significant number of considerable works have produced a wide range of video summarization techniques, leading to notable results.

In [14], the authors formulate a video summarization as a sequential decision-making process while they develop a deep summarization network, trained with an end-to-end reinforcement learning-based framework, which is able to predict for each video a probability that indicates whether the particular frame will be part of the video summary. The above model architecture consists of an encoder-decoder where the encoder is a convolutional neural network (CNN), responsible for frame feature extraction and the decoder is a long-short term memory network (LSTM), responsible for the frame probabilities.

A novel supervised technique was proposed in [15] for summarizing videos based on a LSTM architecture. This approach automatically selects keyframes or keyshots, deriving compact and meaningful video summaries. In addition, they report that techniques such as domain adaptation may improve the entire process of video summarizing.

A generic video summarization algorithm was proposed in [16] by fusing the features from different multimodal streams. A low-level feature fusion approach using as input visual, auditory and textual streams has been used, so as to develop a well-

formed representation of the input video in order to construct a video summarization based on the informative parts from all streams.

In [17] it is pointed out that the main goal of a video summarization methodology is to make a more compact version of the initial raw video, without losing much semantic information and making it quite comprehensive for the viewer. They present an innovative solution namely SASUM, which in contrast to the techniques so far that only take the diversity of the summary, extracts the most descriptive parts of the video summarizing the video. Specifically, SASUM consists of a frame selector as well as the video descriptors, to compose the final video that will minimize the distance with the generated description from the description that has already been created from human.

A memory and computational efficient technique, based on hierarchical graph-based algorithm which is able to make spatio-temporal segmentation on long video sequences was presented in [18]. The spatio-temporal algorithm repeatedly makes segments into space-time regions clustered by their frequencies, constructing a tree consisting of such spatio-temporal segments. Moreover, the algorithm is boosted by introducing dense optical flow to describe the temporal connections on the aforementioned graph.

In [19], it is emphasized that the huge number of videos that are produced on a daily basis need a summary technique to present a condensed format of the video without the unnecessary information. More specifically, their approach, namely SalSum makes use of a generative adversarial network (GAN), which has been pre-trained using the human eye fixations. The model combines colors, as well as visual elements in an unsupervised model. The protrusions, along with the color information deriving from the visual flow of video through SalSum, compose a video summary.

The work proposed in [20] focuses on the computational model development based on the visual attention in order to summarize videos, mostly from television archives. Their computational model is using several techniques so as to ensemble a static video summary, such as face detection, motion estimation and saliency map computation. The final video summary from the above computational model

consists of a collection of key frames or saliency images extracted from the raw video.

A novel video summarization approach, namely VISCOM was proposed in [21] and was based on the color occurrence matrices from the video, used to describe each video frame. Then, a synopsis of the most informative frames of the original video was composed. VISCOM was tested on a large amount of videos from a variety of categories, in order to make the aforementioned video summarization model robust.

In [22] authors focused on the importance of the video summary on tasks such as video search, retrieval etc. Inspired by the approaches based on recurrent neural networks, they tested a fully convolutional sequence neural network on semantic segmentation as the solution of the sequence labeling problem for the video summarization task.

A deep video feature extraction process was proposed in [23], aiming to find the most informative parts of the video which are required so as to analyze video content. They included various levels of content to train their deep feature extraction technique. Their deep neural network also combined the description of the video, in order to extract the video features and then constructed the video summary by applying clustering based techniques as mentioned by the authors in [24]. The evaluation followed on their work is based on their own video summaries constructed by humans.

The main goal of [24] was to remove redundant frames of an input video by clustering informative frames, which appeared to be the most effective way to construct a static video summary, built from all cluster centers. The frame representation that has been used within the clustering process was based on the Bag-of-Visual Words model.

KVS is a novel video summarization approach, proposed in [25], specified on the video category provided, mainly from the title or the description of the video. A temporal segmentation is initially applied on a given video; its result is used as input on the KVS supervised algorithm, in order to build a higher quality video summary compared to the unsupervised blind video category approaches.

Ma et al. [26] proposed an approach for keyframe extraction and video skimming

that was based on a user attention model. To build a motion model, they extracted video, audio, and linguistic features and built an attention model based on the motion vector field. They created three types of maps based on intensity, spatial and temporal coherence which were then fused to form a saliency map. They also incorporated a static model to select salient background regions and extracted faces as well as camera attention features. Finally, the created audio, speech and music models. The aforementioned attention components were linearly fused to create an "attention" curve. Local maxima of this curve within shots were used for keyframe extraction, while skim segments were selected using several criteria.

Early static approaches were typically based on the extraction of features and clustering; cluster centroids were selected as keyframes. For example, in the work of Gong and Liu [27], feature vectors were extracted using RGB histograms. Then they applied Singular Value Decomposition to obtain the essential underlying structure of this description and drive the problem to a refined feature space. Keyframes were then selected upon clustering of frames within this space.

Mahaseni et al. [28] trained a deep adversarial LSTM network consisting of a "summarizer" and a "discriminator", so as to minimize distance between ground truth videos and their summarizations, based on deep features extracted by a CNN. More specifically, the former consists of a selector and an encoder that selects interesting frames from the input video and encode them to a deep feature. The latter is a decoder that classifies a given frame as "original" or "summary." The deep neural network proposed here, tries to fool the discriminator by providing the video summary as the original input video, assuming that both representations are the same.

All methods and techniques presented in this section are quite significant for creating video summaries, with some of them being the current state-of-the-art. However, most of them do not consider both visual and aural information. Adding that none of the aforementioned works is applied on user-generated videos, our work, which concentrates at the combination of information from the different modalities extracted from a user-generated video stream, can address this need.

2.2 Related Data Sets

As it has already been mentioned, in this work we aim to automatically generate summaries from user-generated videos, mostly from action and extreme sports. Consequently, we attempt to present recent, publicly available data sets, for related video summarization tasks.

The "MED Summaries" [25] is a new dataset for evaluation of dynamic video summaries, containing annotations of 160 videos in total, with 10 event categories in the test set. Indicative categories are "birthday party," "changing a vehicle tire," "flash mob gathering," "getting a vehicle unstuck," "grooming an animal," etc.

The "TVSum" (Title-based Video Summarization) dataset [29] aims to solve the challenging task of prior knowledge in the main topic of the video. The entire dataset consists of 50 videos of various genres (e.g., "news," "how-to," "documentary," "vlog," "egocentric") and 1,000 annotations of shot-level importance scores obtained via crowd-sourcing (20 per video), while video duration ranges between 2 and 10 minutes. The video and annotation data permit an automatic evaluation of various video summarization techniques, without having to conduct an (expensive) user study.

The "SumMe" [30] is a video summarization dataset consisting of 25 videos, covering holidays, events and sports, downloaded from the popular platform of YouTube, each annotated with at least 15 human-created summaries (390 in total), while the length of the videos ranges from 1 to 6 minutes.

The "UT Ego" (Univ. of Texas at Austin Egocentric) Dataset [31] contains 10[1] videos captured from head-mounted cameras on a variety of activities such as "eating," "shopping," "attending a lecture," "driving," and "cooking." Each video is about 3-5 hours long, captured at 15 fps and at 320× 480 resolution uncontrolled setting. Therefore, videos contain shots with fast motion.

Finally, the VSUMM dataset [32] has been initially used to produce a static video summary, by a novel evaluation method, able to remove the subjectivity of

[1] 4 out of 10 are available due to privacy reasons

the summary quality by allowing objective comparisons of methodology between different approaches. This dataset, also known as "YouTube Dataset" consists of 50 videos from the *Open Video*[2] Project. The duration of the videos varies from 1 to 4 minutes while the duration of the videos in total is 75 minutes approximately. The videos originate from a variety of genres such as documentary, educational, ephemeral, historical and lecture. It consists of 250 user summaries created manually by 50 individuals, each one annotating 5 videos, i.e., each video has 5 video summaries created by 5 different users.

However, in all of the aforementioned cases, the datasets are either not sufficiently large or they are from a wider domain, i.e. they are not explicitly user-generated. Therefore, in this work we also aim to compile a well-defined user-generated dataset to evaluate the training of the proposed methodology.

[2] http://www.open-video.org/

Chapter 3

Dataset

This chapter describes in detail the criteria and the video collection process followed in this work.

3.1 Data Collection

The entire dataset for training and evaluating the proposed video summarization technique, consists of 409 user generated videos that can be found on the YouTube platform. A single camera setup, such as action camera (i.e. GoPro) or smartphone's camera, and the non-existence of video edits and music scores over the original audio source of the videos are the two main criteria that have been applied when gathering the data. This is due to the fact that we wanted our proposed methodology to be applied on unedited "raw" videos so that the process of summarization has a more imperative usefulness. Most videos came from outdoor activities such as action and extreme sports. In particular, the following 14 video categories have been considered: Car Review, Motorcycle Racing, Kayaking , Climbing, Fishing, Spearfishing, Snack Review, Sky Diving, Roller Coasters, Theme Park Review, Downhill, Mountain Bike, Survival in Wild and Primitive Building.

3.2 Annotation procedure

At the end of the video collection process, the annotation process on the selected videos took place. The purpose of the video annotation process was to create the

video summaries as ground truth for training and testing the proposed video summarization technique. More specifically, 22 humans were asked to watch and annotate some videos in order to construct the ground truth video summaries. This process was executed through a web application, specifically designed for this particular annotation pipeline. The application was capable of randomly serving all the videos, one by one, to the end user, while the user was able to watch the whole video, go back and forth in time, and note the time intervals she/he found interesting (informative). The user was able to freely label the timestamps of each informative time interval while the number of interesting intervals was arbitrary without restricting the user, making the whole process completely subjective. The users only had to provide the endpoints (starting and ending timestamps) for each informative time interval. This web application tool, which includes a quick user registration process, is called Video Annotator Tool (VAT) and can be found at the following repository[1].

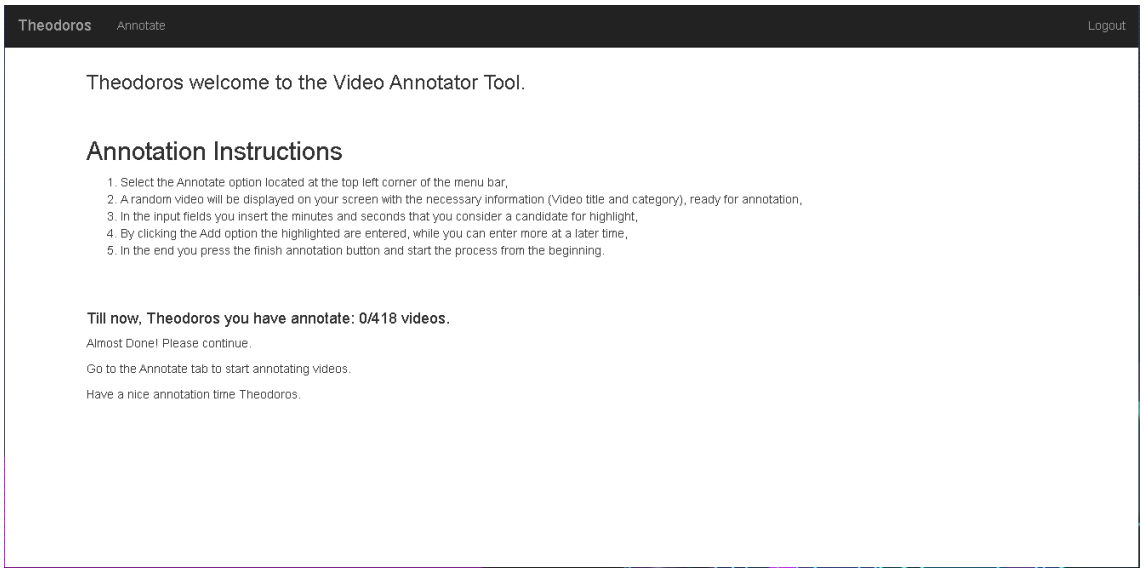

Figure 3.1: Video Annotator Tool Home page

A dataset of 1430 videos was voluntarily annotated by 22 annotators. The group of annotators consisted of students and internship partners from the Computational Intelligence Laboratory. In most of the cases, the videos were annotated by 3 to 4 annotators, while the maximum number of annotators for a given video was 8. The complete distribution of the number of human annotations per video are presented

[1]https://github.com/theopsall/video_annotator

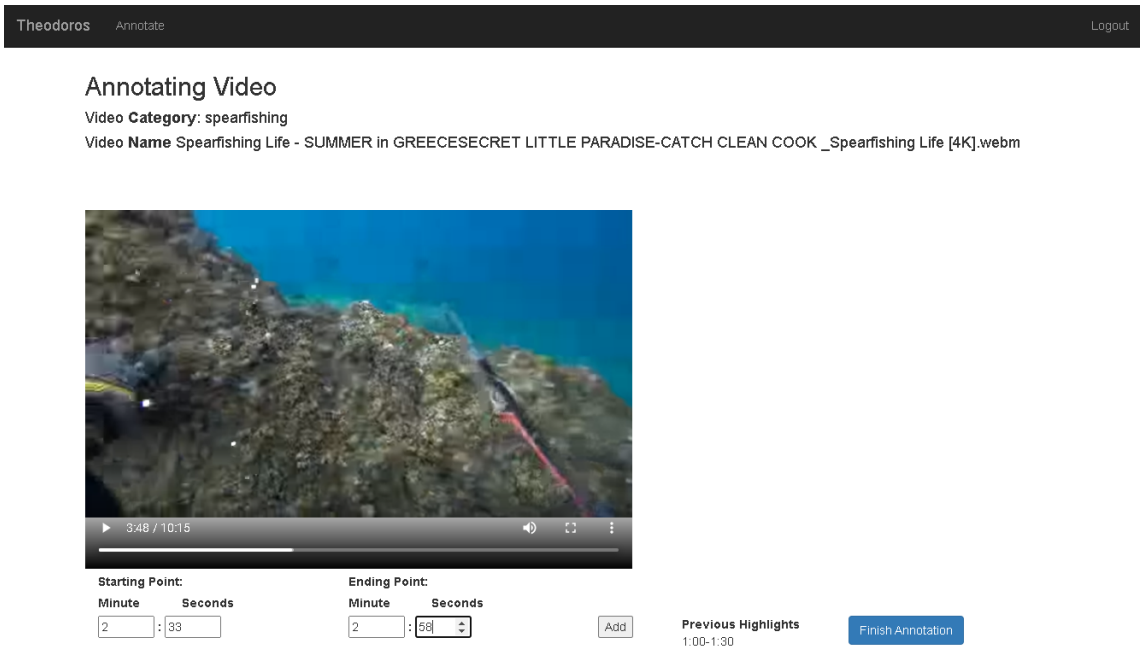

Figure 3.2: The annotating page of the web application with a randomly selected video

in Fig. 3.3.

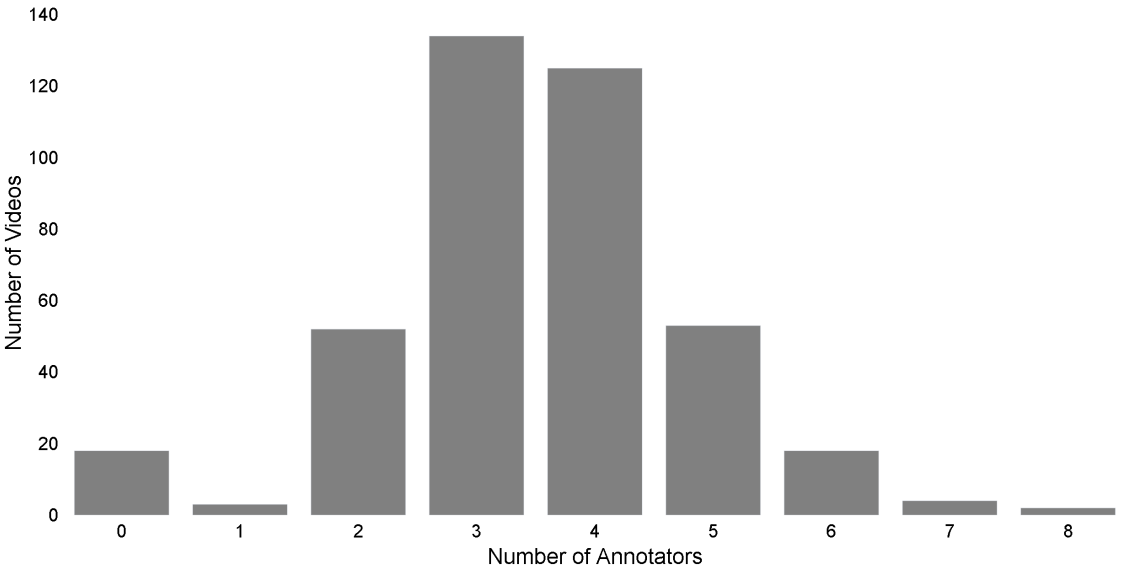

Figure 3.3: Distribution of the number of users annotated each video. Only 73 videos have been annotated by only 2 or less humans.

3.3 Annotation data aggregation

Once the annotation process was completed, the individual video summaries had to be combined, resulting to an acceptable, final ground truth summary that aggregates the opinions of all the users who watched and annotated the specific video. This aggregation process, is rather important for constructing a robust ground truth, since it will be used to evaluate and train the proposed supervised pipeline.

The resulting video summaries that were annotated, as mentioned in the previous section, did not necessarily have the same number of annotators per video, making the construction of a robust dataset more difficult. For that reason, the original dataset has been reduced by deleting the videos that have been annotated by less than 3 annotators. During that process, 61 videos were excluded from the dataset as not having sufficient annotation data. Also, for 12 videos the aggregated annotation resulted in no informative segments at all, therefore these videos were also removed from the dataset. This procedure led to the final dataset comprising of 336 videos that had been annotated by at least 3 different annotators. For these videos that consist the final dataset, the respective ground truth was generated by a simple majority-based aggregation rule. In particular, each segment of 1-sec was considered to be included in the summary (i.e., characterized as "informative") if at least by 60% of the annotators agreed on that decision.

Figure 5.1 illustrates the aforementioned process: in this examples, 5 annotators have provided their opinions about the non-interesting and informative areas of each video. All annotations are first translated into arrays of binary annotations, corresponding to each 1-sec segment. Then, for each segment we extract the aggregated possibility that this segment can be characterized as "informative," based on the 5 annotations. We then threshold and accept as final ground truth, the segments for which the respective threshold is greater than or equal to 0.6. Note that the aggregated agreement is computed as the average agreement between each individual annotation and the aggregated (final) ground truth. In this particular example, this is the average of $[0.9, 0.8, 0.9, 0.8, 0.8]$.

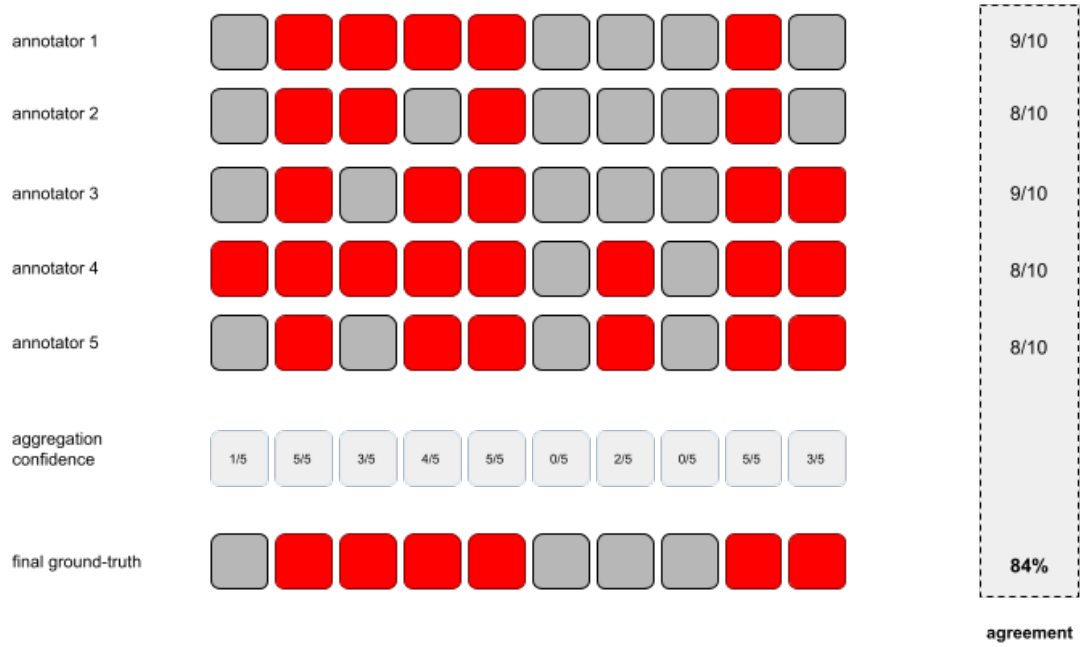

Figure 3.4: Example of aggregating annotation decisions from 5 different human annotators

The final dataset (both raw data and respective annotations)[2], along with the respective tools for the entire aggregation process are publicly available[3].

Table 3.1: Video dataset statistics

Dataset	Total Videos	Total Duration	Av. Dur.	Min Dur.	Max Dur.
Raw Dataset	409	~56.3h	~8.25m	15 sec	15 m
Final Dataset	336	~44.2h	~8m	15 sec	~15 m

We calculated the average agreement as described above, but as a macro averaged F1 metric to have a direct comparison with the automatically generated summarizations. The result was found to be equal to 72.8%.

[2] https://drive.google.com/drive/folders/1-nBp2zJKXsUe2xa9DtxonNdZ6frwWkMp?usp=sharing

[3] https://github.com/theopsall/Video-Summarization

Chapter 4

Theoretical Background

In this chapter, the selected algorithms for the experimental results referred in Chapter 6 are analyzed and described in depth. Before proceeding with the in detailed analysis of each algorithm, a small brief explanation of some machine learning terms follows.

4.1 Multimodal Feature Extraction

Two types of information have been utilized for summarizing the videos: the auditory and the visual modality. To achieve feature representation in both modalities, we extracted hand-crafted features that are frequently used in audio and visual classification, as well as clustering tasks such as music information retrieval, auditory scene analysis, video classification and image retrieval. Our goal was to include as many informative audio and visual features as possible. Figure 4.1 shows the conceptual diagram of the process followed to extract features for both the audio and visual modalities. More details on feature extraction are presented in the following paragraphs.

4.1.1 Aural Features

Low-level hand-crafted features have been shown that they may capture both perceived information, as well as the harmonic information of sound signals thus, they have been widely used in several application domains. Regarding the low-level de-

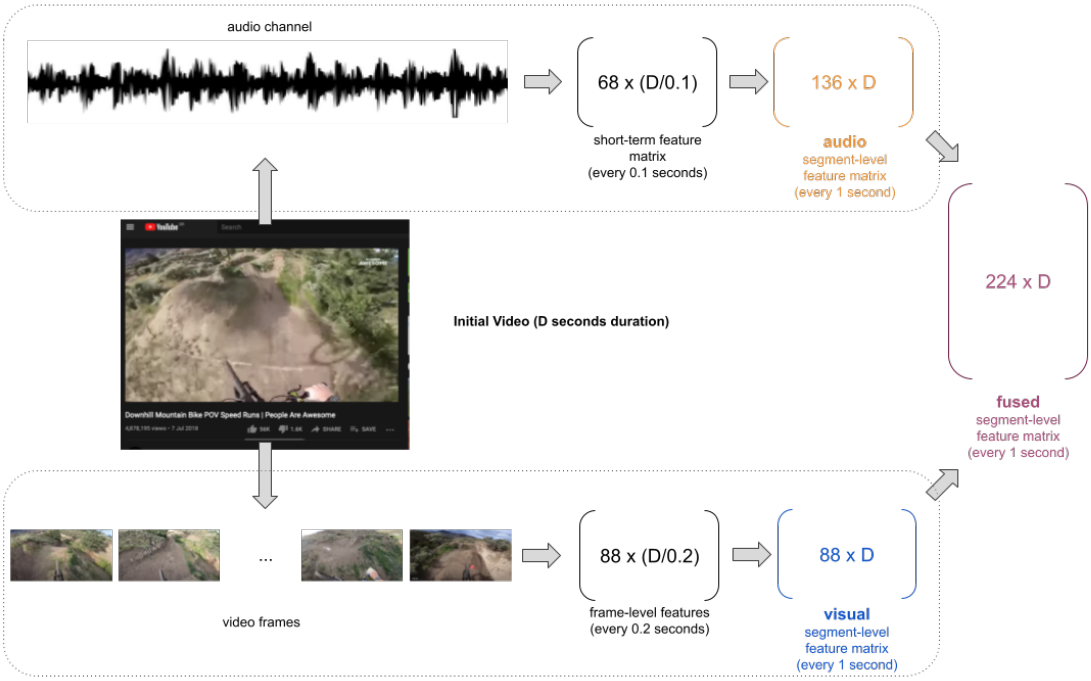

Figure 4.1: Conceptual diagram of the feature extraction process for both audio and visual modalities

scription of the whole event, it has been shown that feature vectors constructed using statistics, such as mean and standard deviation of features, can be efficiently used in event recognition-related tasks [33].

Therefore, for each audio clip, extracted from the respective video file using `ffmpeg`, we calculate segment-level audio features, using the `pyAudioAnalysis` library[1] [34]. According to this procedure, audio feature extraction is firstly carried out at a short-term basis. At a second level, segment-level feature statistics are computed and compose the final segment representation. In particular, the audio signal is divided into segment-level windows (either overlapping or non-overlapping) and for each segment a short-term processing is taking place, according to which, 68 short-term features are computed (34 features and 34 deltas) for each short-term window. Short-term windows usually vary from 10 to 200 mseconds, while segment windows can be from 0.5 second to several seconds, depending on what is considered a homogeneous segment in the individual application domain. The short-term features extracted by the particular library fall into three categories: time-domain,

[1] `https://github.com/tyiannak/pyaudioanalysis`

frequency domain, and cepstral domain. The adopted short-term features are implemented in the pyAudioAnalysis library [34] and are shown in Table 4.1.

Table 4.1: Adopted short-term audio features

Index	Name	Description
1	Zero Crossing Rate	Rate of sign-changes of the frame
2	Energy	Sum of squares of the signal values, normalized by frame length
3	Entropy of Energy	Entropy of sub-frames' normalized energies. A measure of abrupt changes
4	Spectral Centroid	Spectrum's center of gravity
5	Spectral Spread	Spectrum's second central moment of the spectrum
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames
7	Spectral Flux	Squared difference between the normalized magnitudes of the spectra of the two successive frames
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients: a cepstral representation with mel-scaled frequency bands
22-33	Chroma Vector	A 12-element representation of the spectral energy in 12 equal-tempered pitch classes of western-type music
34	Chroma Deviation	Standard deviation of the 12 chroma coefficients.

According to the aforementioned procedure, for every audio segment a sequence of $68-D$ feature vectors is extracted for each short-term window. These vectors are used to compute segment-level statistics, as the final segment representation: for each segment (that contains several short-term windows corresponding to several $68-D$ short-term feature vectors), two segment-level statistics are extracted, namely the mean and standard deviation. Therefore, in total, $2 \times 68 = 136$ audio statistics are used to represent each audio segment. In this work we propose using a short-term window size and step of 100 msec, while a non-overlapping segment window of 1 sec has been adopted.

4.1.2 Visual Features

Apart from extracting auditory features from the sound signal of each video, we have adopted a wide range of visual features to describe the content of the visual information, as this modality is expected to be of major importance in the summarization procedure. For extracting these visual features, the `multimodal_movie_analysis` library[2] has been used to extract features representing visual characteristics of a video. In particular, every 0.2 sec, the following 88 visual features are extracted from the corresponding frame:

- Color - related features (45 features):
  - 8-bin histogram of the green values
  - 8-bin histogram of the blue values
  - 8-bin histogram of the grayscale values
  - 5-bin histogram of the max-by-mean-ratio for each RGB triplet
  - 8-bin histogram of the saturation values
- Average absolute difference between two successive frames in grey scale (1 feature)
- Facial features (2 features): The Viola-Jones [35] OpenCV implementation is used to detect frontal faces and the following features are extracted per frame:
  - number of faces detected
  - average ratio of the faces' bounding boxes areas divided by the total area of the frame
- Optical-flow related features (3 features): The optical flow is estimated using the Lucas-Kanade method [36] and the following 3 features are extracted:
  - average magnitude of the flow vectors
  - standard deviation of the angles of the flow vectors

[2] `https://github.com/tyiannak/multimodal_movie_analysis`

  - a hand-crafted feature that measures the possibility of a camera tilt movement – this is achieved by measuring a ratio of the magnitude of the flow vectors by the deviation of the angles of the flow vectors.

- Current shot duration (1 feature): a basic shot detection method is implemented in this library. The length of the shot (in seconds) in which each frame belongs to, is used as a feature.

- Object-related features (36 features): We use the Single Shot Multibox Detector [37] method for detecting 12 categories of objects. For each frame, as soon as the object(s) of each category are detected, three statistics are extracted: number of objects detected, average detection confidence and average ratio of the objects' area to the area of the frame. Thus in total, 3x12=36 object-related features are extracted. The 12 object categories we detect are the following: person, vehicle, outdoor, animal, accessory, sports, kitchen, food, furniture, electronic, appliance, and indoor.

The aforementioned features provide a wide range of low (simple color aggregates), mid (optical flows), and high (existence of objects and faces) representation levels. The rationale behind the selection of this wide range of types of features lies in the fact that our goal is to cover every type of information that may possibly be correlated to the visual "informativeness" of the video. In other words, part of this work is to discover which types of visual information is mostly associated with what makes (or does not make) an informative video part, i.e., which visual cues make a visual segment interesting.

The dataset presented in this work has been annotated at a resolution of 1-sec segments, thus a way to represent these intervals in the feature space is needed, since the visual features are produced in a 0.2 sec step. For that reason, the mean value across 5 subsequent feature vectors was calculated for each feature. In this way, the representation of every 1-sec segment represent a straightforward alignment to the respective audio features.

4.1.3 Fusion Features

The process by which we combine information from multiple sources is called fusion. The fusion can contribute positively to the data comprehension and feature extraction in the model by increasing its performance. The two most well-known ways to combine information are early and late fusion. In early fusion, different sources of information are combined before being given as input to a model, thus creating a superset of data. This superset will be treated differently from the model so it will discover more patterns. The late fusion concerns the combination of information after it has been given to the models, a combination at a higher level in the pipeline of the methodology to create a more robust model or to combine information from different models.

In this work, early fusion was performed as already mentioned above. After data aggregation in the aural features, we converted the features from both different modalities in order to have time as a common axis, so that they can be combined. The concatenation of the two modalities resulted in a feature matrix with 224 features.

4.2 Machine Learning

Machine learning is the most popular field of computer science and major topic in the information era. Machine learning is an subarea of the artificial intelligence, which enables a computer system to "Learn" through some algorithms and methods.

Machine learning relies purely on the experience accumulated by the training algorithm while is trying to improve the results it produces. After the Big Data explosion, machine learning algorithms were fueled by huge amounts of data, resulting into increased data accuracy. An example of machine learning is email filtering, which detects if one email is spam or not. The filter is a machine learning model, that has been trained through a series of emails from which, some have been labeled as spam and some as not spam, in order to gain the knowledge and experience to separate them, informing the user not to open them.

Therefore, machine learning invents complex mathematical models that lead to

better decision making. There is no strict definition for machine learning, though the most appropriate definition is that machine learning is the creation and continuous improvement of mathematical models within a dataset, in order to extract valuable results and learn patterns [38].

Machine learning is divided into four major categories, depending on the nature of the data. The categories are supervised learning, unsupervised learning, semi-supervised and reinforcement learning. Based on the structure of the data, the problem of our work belongs to the supervised learning category.

4.2.1 Supervised Learning

Supervised learning is one of the most common subcategories of machine learning. More specifically, in supervised learning belongs every task where the samples come with labels, and the system is called to learn a function that maps the input data to the existing labels. Labels are also known as groundtruth. Supervised learning algorithms can be divided into 2 sub-categories classification algorithms and regression algorithms. In the former, the labels are strictly distinct values, such as in the video summarization problem. The classification in turn, is also divided into two subsections: binary classification, where the labels can have only two values and multiple classification, where the data classes are more than two. On the other hand, regression algorithms involve datasets that have continuous and non-discrete values.

4.2.2 Classification

The output result of this problem, indicates whether a particular second of a video stream is informative or non-informative. We will analyze the classification algorithms and more specifically, the binary classification algorithms as we used in this work.

4.2.3 Naive Bayes

Naive Bayes classifier belongs to the supervised learning category exclusively. Naive Bayes classifier took its name from the famous British mathematician Tomas Bayes. The classifier is purely a probabilistic algorithm based on the Bayes theorem, widely used in statistical analysis.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

The term naive on the classifier is derived from the assumption that the input data and the attributes are independent of each other. Despite the independence assumption of its nature, It seems that Naive Bayes that often competes [39] well with more advanced classifiers.

4.2.4 K-Nearest Neighbor

K Nearest Neighbor or KNN in short, can be used for both classification and regression tasks. KNN is a non-parametric algorithm as it does not make any assumption for the distribution of the data. KNN is also known as a lazy algorithm [40] since the algorithm does not need to be trained in order to build a model, due to the fact that all the data is used in the testing phase. This approach makes the training process fast but memory inefficient. The K in the algorithm's name implying the number of the nearest neighbors, is the key decision factor for the output.

KNN first finds the k closest neighbors using a distance metric such Euclidean, Manhattan or Minkowski distance and then classifies the corresponding data points by majority voting of their k-neighbors.

4.2.5 Logistic Regression

Logistic regression or logit model, inpired by the field of statistics, is named after the sigmoid function used at the core of the algorithm. Logistic regression is one of the most popular classification algorithms of machine learning used when the dependent

variable is categorical, based on the concept of probabilities [41]. Basically, this algorithm is most often used in binary classification tasks. Logistic regression is also called sigmoid function. It is essentially, a function that models the data in such a way that the result is clear, e.g 0 or 1, fail or pass, win or loss.

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

The decision boundary of the Logistic Classifier can be changed by setting the threshold. Decision boundary is a major importance to the final outcome of the classifier.

4.2.6 Decision Tree

The decision tree classifier is one of the most widely known classifiers in the category of supervised machine learning algorithms [42]. Decision trees are used only on supervised learning problems, without any restriction on the nature of the labels, whether they are discrete or continuous. For each attribute in the data set, the decision tree creates a node, where it places the most important attribute depending on the entropy it has, to determine the result. The evaluation of the performance of the decision tree starts from the root of the tree and gradually descends to the leaves, which contain the prediction result of the algorithm. Each node in a decision tree is a control based on each attribute. The path that each value can follow is unique within the tree until it ends up in the leaves where the labels are stored. The aim is to create the smallest tree as possible. For this reason, the feature in which the split will take place at each node is of major importance. To achieve the best post-data breakdown, the attribute with the highest entropy is always set as a control feature at the node. The greater the entropy in the distribution of data the greater the confidence in the result. Decision trees are sensitive to the unequal distribution of data between classes.

4.2.7 Ensembles

The last two algorithms that took place in the experiments belong to the ensemble category. More specifically, ensembles algorithms are meta-algorithms that combine the result from individual classifiers. They tend to give robust results as they combine the knowledge of many algorithms, not necessarily the same ones. These meta-algorithms are divided into 3 subcategories: voting, bagging, and boosting. The voting subcategory includes the meta-algorithms that consist of many simple classifiers, usually of different nature each, wherewith simple statistical methods, such as the mean value are used to combine the results. Bagging meta-algorithms consist of many models usually of the same type, which have been trained on different subsets of the training dataset. Finally, in the boosting ensembles category belongs the meta-classifiers, where many models of the same type are placed sequentially and each one tries to correct wrong predictions from the previous one. In this work 2 meta-algorithms were used, Random Forest and XGBOOST from the bagging and boosting category, respectively.

4.2.8 Random Forest

The random forest algorithm is used for both classification and regression tasks and belongs to the category of bagging ensembles as it consists of many decision trees, hence the name forest. The term "random" in the classifiers' name results from the random subset selection of attributes from the original dataset, in order to build its decision tree [43]. Random Forest classifier first trains various decision trees, where each tree is grown without any pruning, on random subsets of the training dataset. Afterwards, the classifier gets the prediction from each decision tree and aggregates the results by majority voting, in order to extract the final prediction. The majority voting strategy in the final step, is able to tackle the problem of overfitting. Random Forest is also known as a pretty good indicator of the feature importance.

4.2.9 XGBoost

Extreme Gradient Boosting or XGBoost in short, is a decision-tree-based classifier that uses the gradient descent in order to minimize the prediction errors at the core of the algorithm and belongs to the boosting ensembles family. XGBoost is an implementation of gradient boosted decision trees optimised for speed and performance. The boosting classifier seems to give state-of-the-art results [44] in most of the machine learning tasks, especially when it has to handle sparse data.

4.2.10 Hyper Parameter Tuning

The most crucial part of every machine learning application, is the correct choice of hyperparameters in order to increase the performance and construct a robust model. Each machine learning algorithm has a variety of hyperparameters. The hyperparameters of a model contribute to the training process like weights. In contrast to the weights though, hyperparameters are not learned from the model during the training process, thus they are given externally from the user. This external intervention in the hyperparameters is of major importance. Hyperparameters differ from algorithm to algorithm, but also differ in the same algorithm applied on different problems with different input data. Essentially, during the process of hyperparameter tuning, we will find the best combination of hyperparameters for each classifier. A common tactic is to optimize the hyperparameters, only on the best classifier among others in the task, as the optimization process is time-consuming and requires a huge amount of power and resources from the building system. The optimal choice of hyperparameters can contribute beneficially in the performance of the model. Furthermore, it can reduce the human effort needed to find the best model, while reducing the human error and the model's bias. Finally, it allows the reuse of heuristic methods [45].

Hyperparameter optimization can be divided into various categories, depending on the optimization algorithm process, with the most prevalent being random search and exhaustive search. On the one hand, random search tests for a finite number of iterations, defined by the user, various random combinations from a range of values

that have been set for each hyperparameter. On the other hand, exhaustive search or grid-search tries all possible combinations of these parameters. Values of each hyperparameter during the grid-search have also been set by the user, but in this search there is not a repetition limit. Grid-search is obviously slower in execution compared to random search, which can become an obstacle in some applications. The values for the hyperparameters can also be given as distributions. As a result, in the process of finding the best hyperparameter, a value through a distribution is being chosen.

In this work, we attempt to improve the classifiers in order to export a more robust model with a higher performance, affecting the final result of the video summary, using grid-search.

Chapter 5

Multimodal Video Summarization

5.1 Problem formulation

In this work, a multimodal supervised video summarization technique is proposed, which belongs to the general video summarization category widely known as *video skimming*. This includes methods that focus on generating a temporally abridged version of a longer video, by identifying significant parts of the video. In this work, we propose analyzing the video stream in one-second segments of audio and visual representations in the context of a supervised technique, according to which the segments are either classified as "informative" (i.e. the are interesting enough so they can be used to compose the final video summary) or "uninformative" (i.e. they don't contain information that could be used in a summary. This is a achieved through a supervised binary classifier trained upon the feature representations of either audio, video, or fusion modalities.

5.1.1 Segment-level classification

According to the process presented above, the content of each video has been described by an audio and a visual feature vector that represents each 1-sec segment of the video. In addition, as described in Section 4, each 1-sec segment of the video has been characterized either as "informative" or as "uninformative". Informative segments are the ones that construct the summary according to the aggregation

process of the annotation data described in Section 3, while "uninformative" are the remaining 1-sec segments that the annotators have agreed that should not be a part of the video summary. Based on this separation, a binary classification task can be formulated.

In order to properly classify each segment of the video with regards to this binary classification task of significant vs insignificant video segments, a variety of classifiers have been trained on three different feature categories:

- audio features: the 136-D audio feature vectors
- visual features: the 88-D visual feature vectors
- audio-visual features: the merged 224-D feature representation (as an *early* fusion approach)

For the training procedure, a training set was created, which consisted of 80% of the number of videos from the initial dataset. The rest of the data was used for validation purposes. The following classifier types were evaluated for all three feature modality setups: Naive Bayes, k-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest and XGBoost. For the first four, the implementation of [46] with the appropriate parameter tuning was used. For the k-Nearest Neighbors classifier the k parameter, which represents the number of neighbors, was optimized using grid search. As for the Logistic Regression, the inverse regularization parameter (i.e. C), was tuned. Decision Tree classifier was optimized with respect to the split quality measure criterion (i.e. Gini impurity, entropy etc) and the maximum tree depth. The Random Forest classifier was based on the balanced classifier provided by [47], while for the XGBoost the classifier of [48] was used. Both of these classifiers were optimized in regard to the split quality measure criterion and the number of tree estimators. The optimization for all the estimators, in this work, was achieved by using grid-search.

Table 5.1: Hyperparameters of estimators

Classifier	HyperParameter	Best Value
Logistic Regression	C	0.1
KNN	k	5
Decision Tree	criterion	entropy
Decision Tree	max depth	6
Random Forest	criterion	gini

5.1.2 Post-processing

An attempt was made to further improve the final result, by offering an aid on the best model. Due to the fact that we want the short video to be continuous without many irregularities in the flow of the frames, we applied a median filter.

Once the segment level classifiers are trained, they can be used to generate the summary of a video. This is achieved in three steps:

1. calculate the audio, visual or fused features for each segment of the video
2. classify each segment of the video by applying the respective audio, visual or fusion classifier
3. post-process the sequential classifier predictions in order to avoid obvious errors

With regards to the post-processing step, a pipeline of two different filters was created to address this need. First, a median filter of length N_1 is applied to the input array using local windows to smooth the sequential classifier predictions. Subsequently, hard filtering was used to determine the final predictions by applying a simple rule, according to which, a sequence of successive positive predictions (informative segments) is kept as is, if at least N_2 segments belong to that sequence. In other words, that rule forces a minimum duration of an informative segment of N_2 seconds. As explained in the experimental section, we have set N_1 equal to 3 and N_2 equal to 5. For example:

- if $p = [10100111001101100000]$ are the predictions of the segment classifier for a particular video

- then $p_m = [11100111001111100000]$ is the output of the median filtering
- and $p_f = [00000000001111100000]$ is the final post-processed prediction.

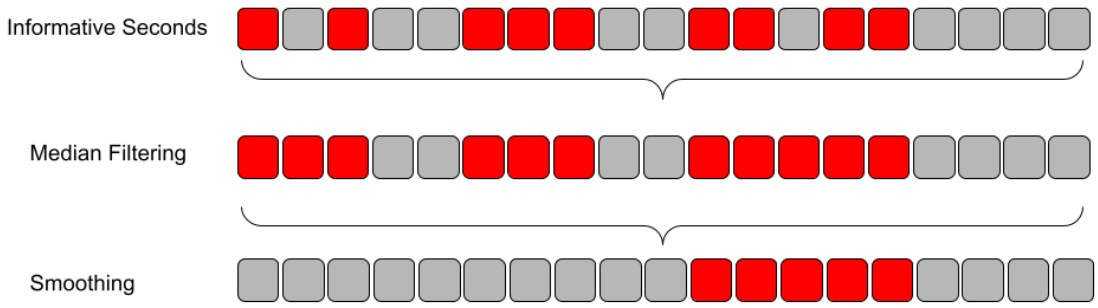

Figure 5.1: Example of the post processing pipeline in the final prediction

Chapter 6

Experimental Evaluation

6.1 Evaluation Metrics

Before proceeding to the definition of the adopted evaluation metrics for the proposed classification task, let us focus on the way the data was split into training and testing subsets. Train/test split strategies are significantly contributing to the statistical correctness of the results of any supervised methodology. In this work, we chose to split the data not at segment level, but at the video level of the dataset. In that way, different 1-sec segments (i.e. individual examples of the classification task) of the same video cannot belong to both training and test sets at the same time, since that would introduce significant bias in the results, as the classifiers would be "video-dependent". Under that constraint, 20% of the overall data were used for testing as presented in table 6.1.

Table 6.1: Training - test samples

Subset	Total Videos	Total samples
Training Dataset	268	127972
Test Dataset	68	31113

As soon as the data were split based on the aforementioned procedure, we measured the following classification metrics:

- Precision for the positive class ("informative"): this measures the percentage

of 1-sec video segments classified (detected) as "informative" that are, indeed, informative according to the ground truth.

- Recall for the positive class: the percentage of 1-sec video segments that have been annotated as "informative" and are correctly detected as such.
- F1 score (macro averaged), i.e., the macro average of the individual class-specific F1 scores. F1 score is the harmonic mean of recall and precision, per class; therefore, the F1 macro average provides an overall normalized metric for the general classification performance.
- Overall accuracy: the overall percentage of the correctly classified (negative or positive) 1-sec segments
- AUC: area under the ROC curve is used as a more general metric of the classifier to function at various "operation points," corresponding to different thresholds applied on the posterior outputs of the positive class.

From the aforementioned performance metrics, F1 macro average and the overall accuracy provide a general evaluation metric of the classification task under study, with F1 being more suitable as it takes into account the class imbalance of the task. Positive class recall and precision are mostly provided as indicative measures of the selected operation point of the classifier. For example 50% precision and 60% recall in the positive class, means that 1 out of 2 from the detected 1-sec segments are indeed informative, while 6 out of 10 real informative segments are detected. AUC is also useful for quantifying the general ability of the classifier to discriminate between the two classes, regardless of the adopted probabilistic threshold.

6.1.1 Results

Table 6.2 shows the calculated AUC and F1 scores for six different classification methods and for the 3 individual modalities (audio, visual and fusion). Random Forest seems to be the best choice, since it achieves the best AUC score and one of the best F1 scores. However, AUC is more important in our case as it incorporates the ability of the classifier to function at different operation points, i.e. different

probabilistic thresholds. It is also important to note that the classifiers based on the visual modality are always at least 4% relatively better than the audio-based classifiers, while the relative improvement is almost 3% better in fusion compared to the visual modality.

Table 6.2: Segment Level Metrics

Classifier	ROC AUC			F1 macro averaged		
	Audio	**Visual**	**Fused**	**Audio**	**Visual**	**Fused**
Random	49.7%			47.6%		
Naive Bayes	59.5%	64%	63.4%	51.7%	48.3%	51.6%
KNN	59.3%	60.7%	62.6%	54.6%	56.3%	57.7%
Log Reg	62.8%	67.2%	67.4%	41.4%	44.6%	49.4%
Decision Tree	60.6%	66.3%	66.5%	41.8%	45.6%	45.6%
Random Forest	**66.7%**	**69.8%**	**71.8%**	57.8%	**60.4%**	60.6%
XGBOOST	65.3%	66.8%	69.6%	**59.8%**	**60.4%**	**62.3%**

Table 6.3 presents the final Precision, Recall, F1 score and Accuracy for the best classification method (Random Forest), when also using the post-processing approach described in Section 5.1.2, for different values of the N_1 and N_2 parameters of the filtering process. It can be comcluded that for $N_1 = 3$ and $N_2 = 5$, F1 is relatively boosted by almost 4%. In terms of the positive class recall and precision rates: the first is increased and the latter is decreased as expected, given that the filtering process removes positive predictions that do not match the aforementioned criteria. Overall, the 63% F1 macro-averaged score achieved by the method is reasonable, since the human performance on that metric, as measured from the agreement between the annotators, is 72.8%.

Table 6.3: Random Forest performance metrics for different parameters of the post-processing technique described in Section 5.1.2

thresholds (med (N_1) - hard (N_2))	**precision**	**recall**	**f1 macro**	**accuracy**
no	42.2%	69.9%	60.6%	62%
3 - 3	43.7%	69.7%	62%	63.8%
3 - 5	**44.9%**	**66.9%**	**63%**	**65.3%**
5 - 3	43.4%	70.8%	61.8%	63.4%
5 - 5	44.2%	69.9%	62.5%	64.3%

In addition to the methodology for composing the summary of a video, emphasis was given on recognizing the features that contributed the most to the final result of the random forest classifier. By using the Recursive Feature Elimination (RFE), a feature selection algorithm, able to rank features with recursive feature elimination, we were able to find the 10 most crucial audiovisual features out of the 224. In Table 6.4 the 10 features with estimated rank 1 by RFE are presented. The purpose of the RFE feature selection procedure, was to find out the key features on which our proposed video summarization classifier algorithm is based on.

Overall, 3 out of 10 features were from the audio domain and 7 from the visual. From the audio domain, two spectral and cepstral delta features were selected, along with the mean statistic of the spectral flux feature, which is something that makes sense, if we consider that spectral flux is a measure of spectral changes in successive audio frames (and delta features are, by definition, measuring changes in the respective features). With regards to the visual domain, 3 out of 7 features were related to motion and/or frame-level changes: frame-level diff, standard deviation of the magnitude of the flow vectors and shot duration (shot-detection is also extracted based on a set of thresholding-rules, related to movement and frame-level changes). The rest 4 significant visual features were not related to movement and frame diffs: the 1st and 4th histogram bins of the grayscaled values of the frames and the 2nd and 6th of histogram bins of the saturation values correspond to the percentages of (a) very dark, (b) significantly light, (c) very unsaturated (i.e. almost grayscale) and (d) very saturated (colorful) images, respectively. Therefore, it seems that information about extremely colorful and bright (and their complementary) is meaningful to the selection of the informative frames for the summary of the video.

To sum up, the fundamental conclusions from the previously described experimental procedures are the following:

- Random forest achieves the best classification performance in terms of AUC for the binary classification task in all three modalities (visual, audio and multimodal).

- Visual - based classifier is always almost 4% relatively better than audio-based.

Table 6.4: The 10 most valuable features for prediction

Feature Name	Description	Modality
spectral_flux_mean	Mean spectral Flux value	audio
delta spectral_spread_std	Delta spectral spread standard deviation	audio
delta mfcc_5_std	Delta MFCC 5 standard deviation	audio
hist_v0	1^{st} bin of grayscaled value	visual
hist_v3	4^{th} bin of grayscaled value	visual
hist_s1	2^{nd} bin of saturation value	visual
hist_s5	6^{th} bin of saturation value	visual
frame_value_diff	Frame value difference	visual
mag_std	Magnitude flow standard deviation	visual
shot_durations	Current shot duration	visual

- Fusion - based classifier is always almost 3% relatively better than visual-based, which indicates that the two modalities both contain useful information for the summarization task.

- The final performance of the binary classifier after applying the proposed post-processing technique reaches almost 45% precision and 67% recall rate at a 1-second segment level.

- Motion-related features seem to be among the most significant features with regards to the classifiers' decision, along with some spectral domain audio features and color intensity and saturation features.

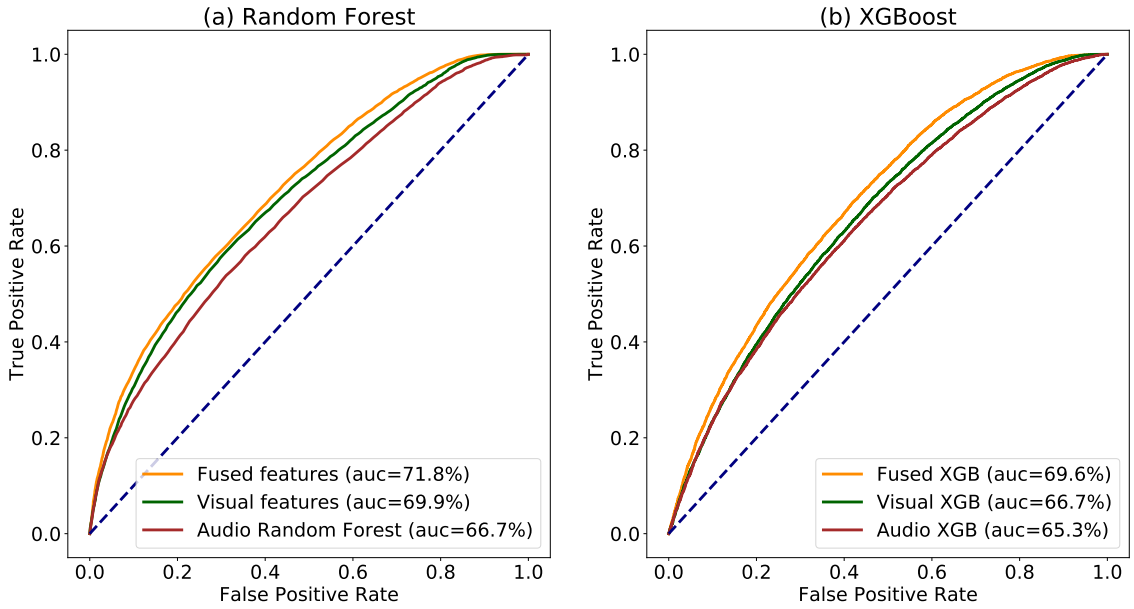

Figure 6.1: Receiver Operating Characteristic (ROC) curve of the (a) Random Forest classifier, (b) XGBoost classifier

Chapter 7

Conclusions and Future Work

7.1 Conclusion

In this work, we have presented an approach for video skimming that effectively used both audio and visual modalities and was applied in user-generated videos. We trained binary classifiers that learnt to discriminate between "important" (informative) segments, i.e., those that should be a part of the produced summary and "non-important" ones, i.e., those that should be discarded. A novel training and validation set was created by human annotators, for every 1-sec part of each video. The dataset contains user-generated videos, collected from YouTube, and recorded using either action cameras or smartphones. We used several annotators and filtered ambiguous or insufficient annotations, while we also measured inter-annotator agreement. The dataset has been made available for future use and comparison. We evaluated our approach using six classifiers trained on audio, video and fused features. Our experimental results indicated that both audio and visual features are important for classification.

7.2 Future work

As future work, it would be quite interesting to train the proposed method on a robust training dataset with a large number of videos from various categories annotated from an extensive number of users. An exploration in further modalities, such

as text, may accompany a video in places like the description, title, subtitles or even in the comments. The possible absence of such a huge dataset would be interesting to study and start as a data collection process in our methodology. Extension to the machine learning models, deep learning algorithms, such as time-series models. More specifically, from the sequence to sequence models, LSTM and Transformers will be used in the updated version of our work. A different pre-processing technique will be quite promising in the results, such as a deep neural network as a video feature extractor. Lastly, an A / B testing by measuring the final result of the method based on the user's point of view, who did not participate in the annotation process, could be explored as a totally different and novel evaluation method, evaluating the videos at the summary level.

References

[1] Youtube. Youtube in Numbers. https://www.youtube.com/intl/en-GB/about/press/, 2021. [Last accessed 20-February-2021].

[2] M. Furini and V. Ghini. An audio-video summarization scheme based on audio and video analysis. In *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.*, volume 2, pages 1209–1213. IEEE, 2006. ISBN 978-1-4244-0085-0. doi: 10.1109/CCNC.2006.1593230. URL http://ieeexplore.ieee.org/document/1593230/.

[3] Arthur G Money and Harry Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of visual communication and image representation*, 19(2):121–143, 2008.

[4] Ziyou Xiong, Regunathan Radhakrishnan, Ajay Divakaran, Zou Yong-Rui, and Thomas S Huang. *A unified framework for video summarization, browsing & retrieval: with applications to consumer and surveillance video.* Elsevier, 2006.

[5] Po Kong Lai, Marc Decombas, Kelvin Moutet, and Robert Laganiere. Video summarization of surveillance cameras. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 286–294. IEEE, 2016. ISBN 978-1-5090-3811-4. doi: 10.1109/AVSS.2016.7738018. URL http://ieeexplore.ieee.org/document/7738018/.

[6] GG Lakshmi Priya and S Domnic. Medical video summarization using central tendency-based shot boundary detection. *International Journal of Computer Vision and Image Processing (IJCVIP)*, 3(1):55–65, 2013.

[7] Hoang Trinh, Jun Li, Sachiko Miyazawa, Juan Moreno, and Sharath Pankanti. Efficient uav video event summarization. In *Proceedings of the 21st Interna-*

tional Conference on Pattern Recognition (ICPR2012), pages 2226–2229. IEEE, 2012.

[8] Evaggelos Spyrou, Giorgos Tolias, Phivos Mylonas, and Yannis Avrithis. Concept detection and keyframe extraction using a visual thesaurus. *Multimedia Tools and Applications*, 41(3):337–373, 2009.

[9] Yingbo Li, Bernard Merialdo, Mickael Rouvier, and Georges Linares. Static and dynamic video summaries. In *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, page 1573. ACM Press, 2011. ISBN 978-1-4503-0616-4. doi: 10.1145/2072298.2072068. URL `http://dl.acm.org/citation.cfm?doid=2072298.2072068`.

[10] R. Lienhart, S. Pfeiffer, and W. Effelsberg. The MoCA workbench: support for creativity in movie content analysis. In *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, pages 314–321. IEEE Comput. Soc. Press, 1996. ISBN 978-0-8186-7436-5. doi: 10.1109/MMCS.1996.534993. URL `http://ieeexplore.ieee.org/document/534993/`.

[11] Bor-Chun Chen, Yan-Ying Chen, and Francine Chen. Video to text summary: Joint video summarization and captioning with recurrent neural networks. In *Procedings of the British Machine Vision Conference 2017*, page 118. British Machine Vision Association, 2017. ISBN 978-1-901725-60-5. doi: 10.5244/C.31.118. URL `http://www.bmva.org/bmvc/2017/papers/paper118/index.html`.

[12] Michael A Smith and Takeo Kanade. *Video skimming for quick browsing based on audio and image characterization*. School of Computer Science, Carnegie Mellon University, 1995.

[13] Debashis Sen, Balasubramanian Raman, et al. Video skimming: Taxonomy and comprehensive survey. *arXiv preprint arXiv:1909.12948*, 2019.

[14] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[15] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In Bastian Leibe, Jiri Matas, Nicu

Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9911, pages 766–782. Springer International Publishing, 2016. ISBN 978-3-319-46477-0 978-3-319-46478-7. doi: 10.1007/978-3-319-46478-7_47. URL http://link.springer.com/10.1007/978-3-319-46478-7_47. Series Title: Lecture Notes in Computer Science.

[16] Georgios Evangelopoulos, Athanasia Zlatintsi, Alexandros Potamianos, Petros Maragos, Konstantinos Rapantzikos, Georgios Skoumas, and Yannis Avrithis. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Transactions on Multimedia*, 15(7):1553–1568, 2013.

[17] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, Xiaokang Yang, and Chen Yao. Video summarization via semantic attended networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[18] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2141–2148. IEEE, 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5539893. URL http://ieeexplore.ieee.org/document/5539893/.

[19] George Pantazis, George Dimas, and Dimitris K Iakovidis. Salsum: Saliency-based video summarization using generative adversarial networks. *arXiv preprint arXiv:2011.10432*, 2020.

[20] Hugo Jacob, Flávio LC Pádua, Anisio Lacerda, and Adriano CM Pereira. *Journal of Intelligent Information Systems*, 49(2):193–211, 2017.

[21] Marcos Vinicius Mussel Cirne and Helio Pedrini. Viscom: A robust video summarization approach using color co-occurrence matrices. *Multimedia Tools and Applications*, 77(1):857–875, 2018.

[22] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, volume 11216, pages 358–374. Springer International Publishing, 2018. ISBN

978-3-030-01257-1 978-3-030-01258-8. doi: 10.1007/978-3-030-01258-8_22. URL http://link.springer.com/10.1007/978-3-030-01258-8_22. Series Title: Lecture Notes in Computer Science.

[23] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato, editors, *Computer Vision – ACCV 2016*, volume 10115, pages 361–377. Springer International Publishing, 2017. ISBN 978-3-319-54192-1 978-3-319-54193-8. doi: 10.1007/978-3-319-54193-8_23. URL http://link.springer.com/10.1007/978-3-319-54193-8_23. Series Title: Lecture Notes in Computer Science.

[24] Jiaxin Wu, Sheng-hua Zhong, Jianmin Jiang, and Yunyun Yang. A novel clustering method for static video summarization. *Multimedia Tools and Applications*, 76(7):9625–9641, 2017.

[25] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8694, pages 540–555. Springer International Publishing, 2014. ISBN 978-3-319-10598-7 978-3-319-10599-4. doi: 10.1007/978-3-319-10599-4_35. URL http://link.springer.com/10.1007/978-3-319-10599-4_35. Series Title: Lecture Notes in Computer Science.

[26] Yu-Fei Ma, Lie Lu, Hong-Jiang Zhang, and Mingjing Li. A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542, 2002.

[27] Yihong Gong and Xin Liu. Video summarization using singular value decomposition. 2:174–180, 2000.

[28] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial LSTM networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2982–2991. IEEE, 2017. ISBN 978-1-5386-0457-1. doi: 10.1109/CVPR.2017.318. URL http://ieeexplore.ieee.org/document/8099801/.

[29] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. TVSum: Summarizing web videos using titles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187. IEEE, 2015. ISBN 978-1-4673-6964-0. doi: 10.1109/CVPR.2015.7299154. URL http://ieeexplore.ieee.org/document/7299154/.

[30] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8695, pages 505–520. Springer International Publishing, 2014. ISBN 978-3-319-10583-3 978-3-319-10584-0. doi: 10.1007/978-3-319-10584-0_33. URL http://link.springer.com/10.1007/978-3-319-10584-0_33. Series Title: Lecture Notes in Computer Science.

[31] Yong Jae Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, 2012. ISBN 978-1-4673-1228-8 978-1-4673-1226-4 978-1-4673-1227-1. doi: 10.1109/CVPR.2012.6247820. URL http://ieeexplore.ieee.org/document/6247820/.

[32] Sandra Eliza Fontes De Avila, Ana Paula Brandao Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[33] Francesc Alías, Joan Claudi Socoró, and Xavier Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6(5):143, 2016.

[34] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12):e0144610, 2015.

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages

I–511–I–518. IEEE Comput. Soc, 2001. ISBN 978-0-7695-1272-3. doi: 10.1109/CVPR.2001.990517. URL http://ieeexplore.ieee.org/document/990517/.

[36] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981.

[37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot MultiBox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9905, pages 21–37. Springer International Publishing, 2016. ISBN 978-3-319-46447-3 978-3-319-46448-0. doi: 10.1007/978-3-319-46448-0_2. URL http://link.springer.com/10.1007/978-3-319-46448-0_2. Series Title: Lecture Notes in Computer Science.

[38] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[39] Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[40] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003.

[41] David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. *Logistic regression*. Springer, 2002.

[42] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[43] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[44] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[45] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, Cham, 2019.

[46] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[47] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

[48] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939785. URL `https://dl.acm.org/doi/10.1145/2939672.2939785`.