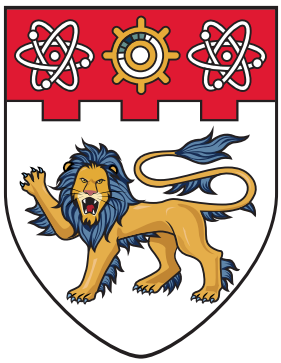


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

VISUAL UNDERSTANDING AND PERSONALIZATION FOR AN OPTIMAL RECOLLECTION EXPERIENCE

ANA GARCÍA DEL MOLINO

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2019

VISUAL UNDERSTANDING AND PERSONALIZATION FOR AN OPTIMAL RECOLLECTION EXPERIENCE

ANA GARCÍA DEL MOLINO

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

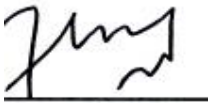
27 March 2019

. . . . . .
Date

. . . . . .
Ana García del Molino

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.



27 March 2019

. . . . . .
Date

. . . . . .
LIM Joo-Hwee

Authorship Attribution Statement

This thesis contains material from papers published in the following peer-reviewed conferences and journals where I was the first and/or corresponding author.

Chapter 2 is partly published as A. G. del Molino, C. Tan, J. H. Lim and A. H. Tan, "Summarization of Egocentric Videos: A Comprehensive Survey," in *IEEE Transactions on Human-Machine Systems*, vol. 47 (1), pp. 65-76, 2017.

The contributions of the co-authors are as follows:

- I did the literature research and prepared the manuscript draft and subsequent edits.
- Dr. Tan, Dr. Lim and Prof. Tan provided ideas and directions to organize and structure the prior research works, and edited the manuscript drafts in various capacities.

Chapter 4 is partly published as A. G. del Molino, J. H. Lim, and A. H. Tan, "Predicting Visual Context for Unsupervised Event Segmentation in Continuous Photo-streams," in *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 10-17, ACM, 2018.

The contributions of the co-authors are as follows:

- I argued the motivation and proposed solution; designed the model architecture and experimental setup; and performed all dataset analysis.
- I wrote the drafts of the manuscript and subsequent edits.
- Dr. Lim and Prof. Tan provided methodology suggestions and edited the manuscript drafts.

Chapters 5 and 7 are partly published as A. G. del Molino, X. Boix, J. H. Lim, and A. H. Tan, "Active Video Summarization: Customized Summaries via On-line Interaction with

the User," in *Thirty First AAAI Conference on Artificial Intelligence*, pp. 4046-4052, 2017.

The contributions of the co-authors are as follows:

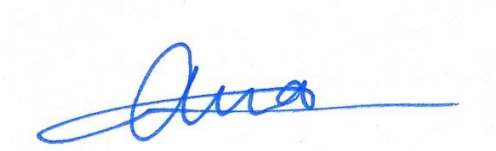
- I argued the motivation. Dr. Boix assisted in designing the model through insightful discussions.
- I designed the experimental setup, performed all dataset analysis and conducted the user studies.
- I implemented the inference framework, performed all experimental tests, designed the interactive GUI, and explored the optimal approach for the user interaction. Dr. Boix contributed the implementation and adaption of the Conditional Random Field.
- I prepared the manuscript drafts. It was then revised and completed by Dr. Boix.
- Dr. Lim provided insightful suggestions for the interaction module.
- Dr. Lim and Prof. Tan edited the manuscript drafts.

Chapter 6 is partly published as A. G. del Molino and M. Gygli, "PHD-GIFs: Personalized Highlight Detection for Automatic GIF Creation," in *Proceedings of the 26th ACM International Conference on Multimedia*, pp. 600-608, ACM, 2018

The contributions of the co-authors are as follows:

- I co-designed the study with Dr. Gygli.
- I implemented, trained and tested all possible methodologies. Dr. Gygli assisted in designing the model through insightful discussions.
- I performed all dataset curation and analysis.
- I prepared the manuscript drafts and subsequent edits. The manuscript was revised and completed by Dr. Gygli.

27 March 2019



. . . . . .
Date

. . . . . .
Ana García del Molino

Abstract

The affordability of wearable cameras such as the Narrative Clip and GoPro allows mass-market consumers to continuously record their lives, producing large amounts of unstructured visual data. Moreover, users tend to record with their smartphones more multimedia content than they can possibly share or review. We use each of these devices for different purposes: action cameras for travels and adventures; our smartphones to capture on the spur of the moment; a lifelogging device to record unobtrusively all our daily life activities. As a result, the few important shots end up buried among many repetitive images or uninteresting long segments, requiring hours of manual analysis in order to, say, select highlights in a day or find the most aesthetic pictures.

The first natural step to organize our visual memory collections is to cluster them into distinct episodic events. Segmenting visual content into events provides semantic structures for indexing, retrieval, and summarization. Current methods to temporally segment egocentric or continuous video data usually compare visual features between frames in an unsupervised manner or analyze motion features. However, such methodologies are ineffective at dealing with heterogeneous events and short lapses between sight directions, *e.g.* navigating in a city or meeting different people at a social event. Contextual Event Segmentation (CES), as proposed in this thesis, addresses these limitations by modeling the visual sequence, predicting its visual context, and tracking its evolution.

In many scenarios, it is desirable to highlight the interesting and aesthetic segments or pictures in a data collection without human intervention. Highlight detection models are typically trained to identify cues that make visual content appealing or interesting for the general public. But the "interestingness" of a video segment or image is subjective, and such highlight models provide results of limited relevance for the individual user. To improve the prediction, Personalized Highlight Detection (PHD) is proposed. Rather than training one model per user, it is a global ranking model which can condition on a particular consumer's interests, effectively adapting its predictions given only a few user-specific examples.

Further from detecting highlights in our personal recordings, being able to obtain an automatic visual summary facilitates the browsing of extensive collections. Many state-of-the-art summarization tools select video segments or pictures by optimizing a predefined criterion, frequently related to story coherence and interestingness. Such methods rarely consider the nature of the media or the purpose of the recording, nor allow for modifications

of the result. Hence this thesis introduces Active Video Summarization (AVS), which is an interactive summarization framework to help the user summarize consumer videos in a fast and simple way. It provides a diverse skim of good visual quality, and relevant to the user's objective.

In summary, tackling challenges in end-to-end consumer video summarization, this thesis contributes to the state of the art in three major aspects: *(i)* Contextual Event Segmentation, an episodic event segmentation method that is able to detect boundaries between heterogeneous events and ignore local occlusions and brief diversions. CES improves the performance of the baselines by over 16% in F-measure, and is competitive with manual annotations. *(ii)* Personalized Highlight Detection, a highlight detector that is personalized via its inputs. The experimental results show that using the user history substantially improves the prediction accuracy. PHD outperforms the user-agnostic baselines even with only one single person-specific example. *(iii)* Active Video Summarization, an interactive approach to video exploration that gathers the user's preferences while creating a video summary. AVS achieves an excellent compromise between usability and quality. The diverse and uniform nature of AVS summaries makes it also a valuable tool for browsing someone else's visual collection. Additionally, this thesis contributes two large-scale datasets for First Person View video analysis, CSumm and R3, and a large-scale dataset for personalized video highlights, PHD^2.

Lay Summary

The introduction of wearable video cameras such as the Narrative Clip and GoPro in the market allows us to continuously record our lives, producing large amounts of unstructured visual data. Moreover, we tend to record with their smartphones more multimedia content than they can possibly share or review. We use each of these devices for different purposes: action cameras for travels and adventures; our smartphones to capture on the spur of the moment; a lifelogging device to record unobtrusively all our daily life activities. The few important shots end up buried among many repetitive images or long uninteresting segments, requiring hours of manual analysis to, *e.g.* , select highlights in a day or find the most aesthetic pictures.

The first natural step to organize our visual memory collections is to group them into distinct life events. Current methods usually compare consecutive frames or analyze the event motion. However, such methodologies are ineffective at dealing with events with too much movement, *e.g.* navigating in a city or meeting different people at a social event. Contextual Event Segmentation (CES), as proposed in this thesis, addresses these limitations by predicting the social context or activity of the scene and detecting unexpected changes.

In many scenarios, it is desirable to highlight the interesting and aesthetic segments or pictures without needing human intervention. Highlight detection models are typically designed to identify cues that make visual content appealing or interesting for the general public. But this "interestingness" of a video segment or image is subjective. Thus, such highlight models provide results of limited relevance for the individual user. To improve the prediction, Personalized Highlight Detection (PHD) is proposed. Given only a few examples of what a user has previously found interesting, PHD can provide better results than previous models.

Further from detecting highlights in our personal recordings, being able to obtain an automatic visual summary facilitates the browsing of extense data collections. Many state-of-the-art summarization tools select video segments or pictures by optimizing a predefined criterion, frequently related to story coherence and relevance. Such methods rarely consider the nature of the media or the purpose of the recording, nor allow for modifications of the result. Hence Active Video Summarization (AVS), an interactive summarization framework to help the user summarize consumer videos in a fast an simple way, providing a diverse visual summary of good image quality, and relevant to the user's objective.

Tackling challenges in end-to-end personal video summarization, this thesis contributes to the state of the art in three major aspects: *(i)* An episodic event segmentation method that is able to detect boundaries between heterogeneous events and ignore artifacts blocking the field of view and brief diversions. CES improves the performance of the baselines and is competitive with manual annotations. *(ii)* A personalized highlight detector that uses the subject's history to substantially improve the prediction accuracy. PHD outperforms the non-customized baselines even with only one single person-specific example. *(iii)* An interactive approach to video exploration that gathers the user's preferences while creating a video summary. AVS achieves an excellent compromise between usability and quality. The diverse and uniform nature of AVS summaries also makes it a valuable tool for browsing someone else's visual collection. Additionally, this thesis contributes two large-scale datasets for First Person View video analysis, CSumm and R3, and a large-scale dataset for personalized video highlights, PHD^2.

Acknowledgements

I would like to thank my supervisor Joo-Hwee Lim for taking me as his student and offering me the opportunity to find my research interest before starting my Ph.D. I will always deeply appreciate his undisrupted availability to discuss whatever new idea or problem, his allowing me to explore a wide range of research lines, and the respect he has shown me. I also want to thank my co-supervisor Ah-Hwee Tan for his accademic guidance and confidence in me. My appreciation also goes to the thesis examiners, for their time, effort and valuable suggestions.

To Ferran Marqués and Xavier Giró-i-Nieto. I would most likely not have started this venture without their mentorship and, most importantly, unconditional sponsorship. I also want to thank Xavi Boix and Michael Gygli for believing in my research and sharing with me their time and knowledge. They boosted me up when I needed it the most, empowering me both technically and research-wise. Also deserving special mention are Stephane, Keng Teck, and Qianli, for so many conversations and valuable discussions.

To my fellow Ph.D.'s and soon-to-be-Ph.D.'s, Artsiom, Justin, Martin, Matthieu, Michal, and Thomas, without you, these years at the lab would have been really boring! Also to those at home, Ariesha, Esther and Laura, and to the frequent visitors, Rajeev and Devika, for putting up with all my reviewing and rants, for making it a pleasure to just be at home, and for sharing so many Netflix hours with me. And of course to the *upesé* family, for being a terrific support group despite the distance. I also want to thank my *oh-so-loyal* long-time friends that still manage to make time to catch-up whenever I show up unannounced in Barcelona for just a few days. They make it feel as if I had never left. I can finally allow them to ask the fearful question "And how is the thesis going...?" I guess trickier questions will be getting in the pipeline now!

Most importantly I want to thank my family, always concerned about my well-being being so far away while never pressuring me to be closer. Thank you for making it such a breeze. To my parents and brother, for their patience with my absences –frequently not just physical– and their full commitment to my personal growth. And last but never less, I want to thank Joaquim for being there every step of the way encouraging me to do –and be– even better.

Contents

List of Tables

List of Figures

List of Abbreviations

CES	Contextual Event Segmentation
AVS	Active Video Summarization
PHD	Personalized Highlight Detection
FPV	First Person View
TPV	Third Person View
LTR	Low Time Resolution
HTR	High Time Resolution
VCP	Visual Context Predictor

C3D	Convolutional 3D
CRF	Conditional Random Field
DCNN	Deep Convolutional Neural Network
DoG	Difference of Gaussians
DPP	Determinantal Point Process
EEG	Electroencephalogram
FNN	Feedforward Neural Network
GRU	Gated Recurrent Unit
HOG	Histograms of Oriented Gradients
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
RNN	Recurrent Neural Network
R-CNN	Region Convolutional Neural Network
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features

Chapter 1

Introduction

This Chapter introduces the motivation for this thesis, as well as the definition and characteristics of First Person Vision. It then presents the thesis contributions, and the thesis outline.

1.1 Motivation

The market of wearable devices has exploded in the past years and many of those devices include a video recording component. The affordability of devices such as the Narrative Clip and GoPro cameras, and the great memory capacities of current smartphones, allows mass-market consumers to continuously record for many hours, producing huge amounts of unconstrained data. As a result, our life is becoming heavily expressed in the digital substrate, potentially extending our memory to elements that we would have forgotten otherwise. But even if this multimedia data provides a better way to recall or re-experience our life events, the device wearer (the person recording the video) may never revisit much of those recorded visual memories. There is a need to identify and locate the meaningful life segments and make browsing and retrieving fast and efficient, as well as piecing segments together into a coherent summary for an optimal recollection experience.

1.1.1 Autobiographic Multimedia Memories

Life-logging devices taking pictures at a fixed interval can be used in cognitive therapy or as a means of memory preservation [34, 35]. There is a wide range of applications that emerge: from highlighting certain life events to identifying periodic patterns of life and behavior and deviations from them. However, lifelogging has an overloading problem both in time and space: These Low Time Resolution (LTR) cameras take a minimum of 2 pictures per minute, which can add to more than $1,000$ pictures a day, *i.e.* $100Gb$ per year. Such vast load of data requires hours of manual analysis to, for example, select your day's highlights, check what you ate and drank the past month, or monitor your grandparent's routines. Hence, automatic tools to extract highlights and life patterns, and retrieve specific past events, are needed [52, 53, 129].

1.1.2 Personal and Consumer Videos

The main usage of personal video cameras is preserving memorable events such as celebrations and holidays. But recording the experience generally means not being able to fully be a part of it. Wearable cameras allow the cameraman to capture the moment while enjoying with the others. Afterward, we hope to be able to share our experience with our friends, family, acquaintances. But we usually do not have the time to edit all our pictures and videos and fail to create that holiday video. Moreover, we would have to create different videos for different people, as not everybody is interested in the same things, and we may not want to share the more personal experiences with everybody. There is a need for a tool to assists in the process, which will understand what is the intention of our summary, and propose what to add next in the memory wrapper that is being created.

1.2 First Person View Summarization

First Person View (FPV) –or egeocentric– recordings comprise images and videos taken with (hands-free) wearable cameras and approximate the wearer's visual experience [1]. As such, FPV differs from Third Person View (TPV) in major aspects related to the content of the recordings, their intention, and visual quality. Consumer and egocentric videos are inherently personal, created to preserve or share our experiences. Since they are recorded using hand-held devices or body-mounted cameras, the videos result unstable, with a substantial presence of visual occlusions, walls and ceilings in the field of view, and frequent changes of visual orientations. The LTR nature of lifelogs further adds dramatic visual changes between consecutive frames even if these correspond to the same event. Thus, summarizing FPV content entails challenges not faced by TPV summarization methods.

1.2.1 Characteristics of First Person View Video

Egocentric videos are unconstrained in nature, lacking a proper structure for the purpose of the video. Moreover, they are manipulated by spontaneous human attention, recording just what the wearer saw, with all body/head associated movements. We can observe and highlight the following principal distinctive characteristics of FPV [26, 121]:

[1] Videos recorded with devices such as the Narrative Clip, Autographer, Looxcie, Google Glass, GoPro, Tobii, etc., are typical examples of FPV videos.

Intention In general, there is no specific intention in the recording, nor focus on the relevant thing the wearer wanted to keep documented. However, the spontaneous nature of the recordings can offer important cues that provide critical knowledge for video summarization, *e.g.* attention can be inferred from head motion.

Content Lifelogging is a hands-free action, so the wearer may record everything while being free to fully enjoy that life experience. As a consequence, most of the logged data could very well be repetitive or irrelevant. Moreover, the video results in a continuum of consecutive events with smooth transitions from one to another.

Quality FPV videos tend to contain many blurry and shaky segments due to ego-motion and are frequently unaligned due to head tilt. In the case of Lifelogs, the camera is usually clipped to the chest and sometimes blocked by the wearer's arms and clothing such as scarfs. When in resting positions, the camera is often facing ceilings and walls.

1.2.2 Challenges and Opportunities

The highly unconstrained nature of FPV content makes traditional TPV summarization methods difficult to apply, since these are generally domain-specific, designed for sports, news, movies, TV dramas, *etc.* The analysis benefits from the rigid structure of those contexts [55, 89], relying on speech excitement, applause, flash lights or "score" cuts, laughs for sitcoms, *etc.* These cues are mostly absent in egocentric video [6, 9, 76], and so are not available for its analysis. Moreover, such long streams of data with very subtle boundaries (both temporal and spatial) add an additional challenge to FPV video segmentation, and the low quality of the recordings hampers accurate feature tracking. Applying TPV summarization techniques over FPV videos provide inaccurate results, even performing worse than uniform sampling in some cases [75].

As many of the state of the art works point out [48, 80, 103], the ideal summary is context dependent. As such, it is important to summarize each kind of video differently. Unlike for TPV, where the context is generally known beforehand and common throughout the video, FPV faces the problem of having to deal with a possibly unknown and diverse context. Therefore, algorithms need to predict the summarization objective, which may change during the recording.

Yet, FPV offers a great advantage over TPV, which is the personal nature of FPV videos. The egocentric point-of-view allows for a privileged peek into interactions with objects, animals and other people. The video captures the wearer's ongoing activities and goals. Following his or her gaze and attention patterns can allow the system to detect highlights [100, 125, 126, 132].

1.3 Research Goals

In order to provide a compact summary of past life events, one needs to be able to separate them. A life event is distinctive from the rest from its overall context: the activity, people, location, *etc.* This thesis explores using the visual semantic context to separate and differentiate episodic events.

Based on the assumption that not all users are interested in the same content, and that users may have different motivations when reviewing their media, the need for customized video summarization arises. In the case of egocentric and consumer videos, and due to their personal nature, this seems of utter importance. Incidentally, there are universal preferences that most humans share and particular items for which most disagree. Customized video summarization methods could predict a given user's preferences. This thesis explores the suitability of an interestingness predictor that can detect what each different user considers a highlight.

Since one may also want to refine an automatically generated summary, a method for the user to give preferences in an active fashion could result in an efficient and usable editing tool. Such a method should put together the highlights of our recent life experience while also conveying a good story-telling summary. Moreover, the time needed to edit the video in this fashion should be considerably inferior to manual editing, while achieving similar quality.

1.4 Thesis Contributions

This thesis presents the following main contributions:

Contextual Event Segmentation (CES) [27] CES is a novel event segmentation methodology for egocentric video content that is inspired in the human perceptual reasoning. CES estimates the event representation at each timestep to infer the presence of an event or shot boundary. The event representation from the past sequence of frames is compared to the event representation given the future sequence (*c.f.* Figure 1.1). An LSTM-based generative model, that we coin Visual Context Predictor (VCP), is used to predict the event representation. It is able to model our daily activities and learn the associations between different scenes, *e.g.* a train commute will include corridors, stairs, a platform, the interior of a wagon, *etc.* As a result, CES improves with respect to the state of the art at detecting boundaries between heterogeneous events and ignoring local occlusions and brief diversions.

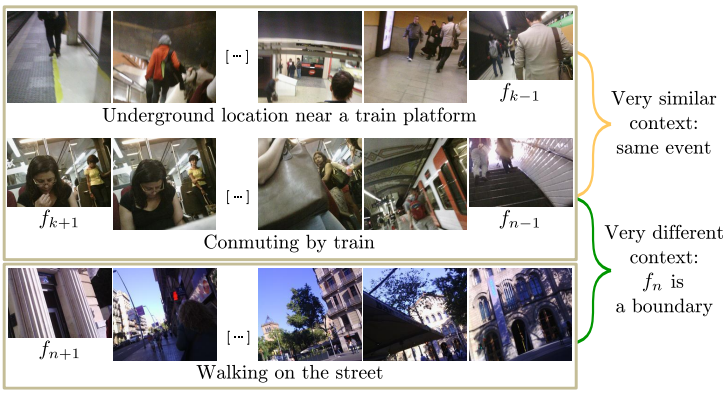


Figure 1.1 Contextual Event Segmentation. As humans, we define a new event when a new sequence of frames differs from our understanding of the previous frame sequence. CES models such intuitive framework of perceptual reasoning by predicting the event representation of the photo-stream. At each timestep, it compares the event representation predicted from the past sequence to the event representation predicted from the future sequence.

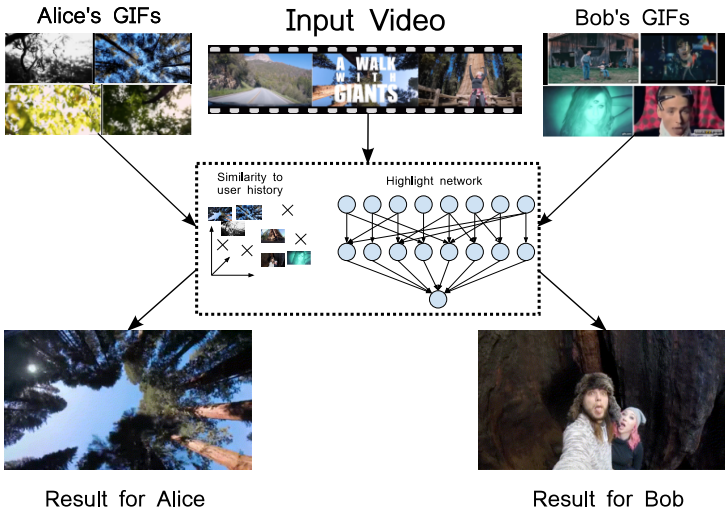


Figure 1.2 Personalized Highlight Detection. The notion of a highlight in a video is, to some extent, subjective. While previous methods trained generic highlight detection models, our method takes a user's previously selected highlights into account when making predictions. This allows to reduce the ambiguity of the task and results in more accurate predictions.

Personalized Highlight Detection (PHD) [22] Highlight detection models are typically trained to identify cues that make visual content appealing or interesting for the general public. However, this "interestingness" of a video segment or image is subjective. PHD is a global ranking model which can condition on a particular user's interests to customize the prediction (*c.f.* Figure 1.2). Rather than training one model per user, PHD is personalized via its inputs, which allows it to effectively adapt its predictions, given only a few user-specific examples.

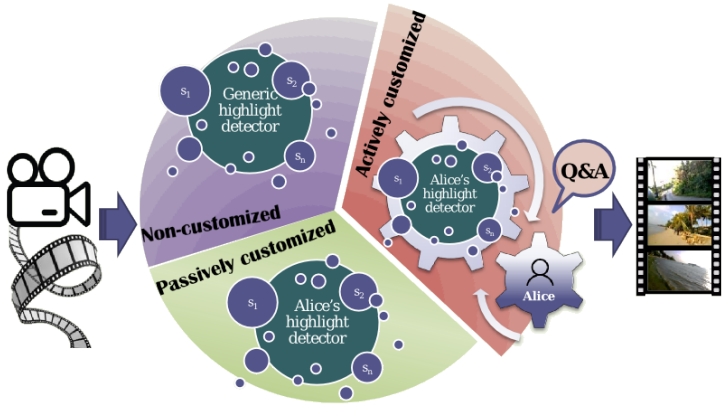


Figure 1.3 Active Video Summarization. The aim of AVS is to provide a customized summary with as little effort as possible from the user side. The system first asks for the user's initial preferences. Then, AVS asks questions about the summary to update it online until the user is satisfied. To minimize the interaction, the best segment to inquire next is inferred from the previous feedback.

Active Video Summarization (AVS) [25] Since the user may have different motivations to summarize a video, the same user preferences might not apply for all cases. AVS is an interactive approach to video exploration that gathers the user's preferences while creating a video summary (*c.f.* Figure 1.3). The same probabilistic model is used to generate a diverse summary and to infer the best interaction with the user. AVS asks questions about the summary to update it online until the user is satisfied. The experimental results show that AVS achieves an excellent compromise between usability and quality. Moreover, it can help the user generate a summary of a video that he has never seen before.

1.5 Thesis Outline

Figure 1.4 presents an overview of this thesis, which is organized as follows:

Chapter 2 presents the most relevant techniques for egocentric video summarization up to date, pointing out their main strengths and shortages. It also presents a framework for first-person view summarization, and compares the features, segmentation methods and selection algorithms used by the related work in the literature.

Chapter 3 describes the most relevant datasets for egocentric video summarization. It then presents three large scale datasets contributed by this thesis, namely CSumm, R3 and PHD^2, as well as insights into the interests and video highlight preferences of over $15,000$ anonymous users.

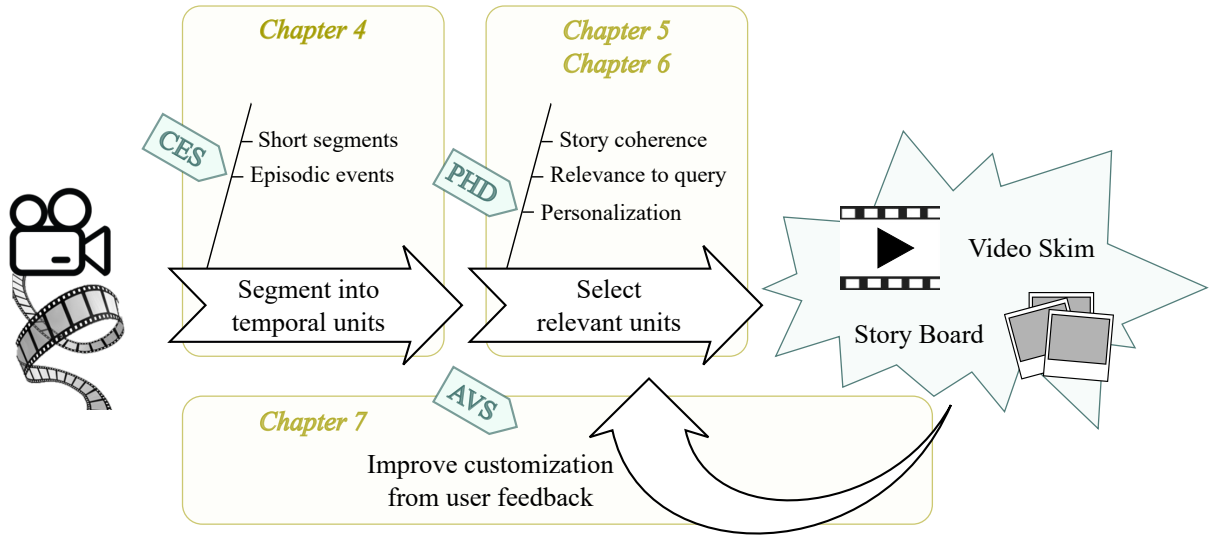


Figure 1.4 This thesis presents contributions in end-to-end video summarization, from temporal unit segmentation to key-frame or segment selection.

Chapter 4 introduces the motivation behind Contextual Event Segmentation, and describes the model design, training methodology and experimental results to validate it. CES is tested in four challenging video datasets consisting of both Lifelog and egocentric consumer videos. Additional experimental results are included in Annex A.1.

Chapter 5 presents two methods for egocentric visual memory summarization. The first uses CES to generate diverse summaries from Lifelog data. The second employs a Conditional Random Field to model High Time Resolution (HTR) egocentric consumer videos and select the most diverse and representative segments.

Chapter 6 introduces the motivation behind Personalized Highlight Detection, and describes the model design, training methodology and experimental results to validate it. Additional experimental results and architectures are included in Annex A.2.

Chapter 7 motivates the design behind Active Video Summarization, which builds on top of the CRF-based video summarization model presented in Chapter 5. This Chapter describes two scenarios in which AVS is of use in practice, and the experimental methodology that validates the proposed interactive summarization approach.

Finally, **Chapter 8** concludes this thesis by giving an overview of the achieved results and insights into the research challenges that remain to be addressed.

Chapter 2

Related Work

The increasing flow of First Person View video has led to a growing need for automatic video-summarization adapted to the characteristics of personal and egocentric content. Since organizing personal and egocentric videos cannot be properly tackled with traditional Third Person View summarization techniques, this topic has attracted a lot of interest recently. This Chapter presents the most relevant video summarization techniques related to FPV, pointing out their main strengths and shortages. It also presents a framework for FPV summarization, and compares the features, segmentation methods and selection algorithms used by the related work in the literature.

2.1 Introduction

First Person View videos are long, personal, and unconstrained in nature. The management of such large amounts of data can be targeted by providing indexing and retrieval systems [3, 16, 23, 33, 41, 107, 122, 130]. Alternatively, the content of the videos or image sets can be summarized, so that the user can appreciate the overall meaning and experience of the recorded memory in a much shorter time [2, 7, 47, 48, 64, 70, 75, 76, 80, 83, 91, 93, 101, 125, 126, 129, 132, 145]. Even though retrieval methods can be used to provide personalized summaries, their use as a tool for summarization is not well explored yet.

Early summarization systems for egocentric video were bottom-up, relying mostly on low-level features and ego-motion characteristics [23, 47, 48, 101, 129, 145]. Some other works applied supervised learning [75, 76, 80, 83, 125, 126, 130] and exploited physiological data such as EEG signals [2, 91] and gaze [132] to select the relevant segments. Recently, however, a new strand of methods use deep learning to obtain summaries from consumer videos [49, 54, 85, 94, 96, 109, 127, 134, 136, 140, 144, 146]. Tables 2.1a and 2.1b summarize the characteristics of each relevant method. Table 2.2 presents in a comprehensive and schematic way the features used in each analyzed paper, the cues used for the segmentation of events, and the objectives to be met when selecting subshots to represent the video.

Table 2.1 Relevant Video Summarization Methods.

(a) Evaluated on FPV

		Dataset			Evaluation		
		UTEgo	Other FPV	Task	Subjective	Annotation	NLP
Aizawa *et al.*, 2001	[2]		✓	V		*none*	
Ng *et al.*, 2002	[91]		✓	V		*none*	
Lee *et al.*, 2012	[76]	✓		SB	✓	✓	
Lu *et al.*, 2013	[83]	✓	✓	V	✓		
Xiong *et al.*, 2014	[129]	✓		SB		✓	
Zhao *et al.*, 2014	[145]		✓	V		✓	
Okamoto *et al.*, 2014	[93]		✓	FF	✓		
Kopf *et al.*, 2014	[70]		✓	FF	✓		
Lee *et al.*, 2015	[75]	✓	✓	SB	✓		
Varini *et al.*, 2015	[126]		✓	V	✓		
Varini *et al.*, 2015	[125]		✓	V	✓		
Xu *et al.*, 2015	[132]		✓	V		✓	✓
Poleg *et al.*, 2015	[101]		✓	FF		*other*	
Joshi *et al.*, 2015	[64]		✓	FF		*other*	
Bettadapura *et al.*, '16	[7]		✓	SB	✓		
Jeong *et al.*, 2016	[59]	✓	✓	SB	✓		✓
Zhang *et al.*, 2016	[143]	✓		SB, V	✓	✓	
Zhang *et al.*, 2016	[144]	✓		SB, V	✓	✓	
Yao *et al.*, 2016	[136]		✓	V, FF	✓	✓	
Sharghi *et al.*, 2016	[108]	✓				✓	
Halperin *et al.*, 2016	[50]		✓	FF		*other*	
Silva *et al.*, 2016	[110]		✓	FF		✓	
Plummer *et al.*, 2017	[99]		✓	V		✓	
Sharghi *et al.*, 2017	[109]	✓		V		✓	

(b) Evaluated on TPV

		Dataset		Method			Objective			
		SumMe	Other TPV	Task	Model	Learning	Diversity	Representative	Interestingness	User/context driven
Wang *et al.*, 2014	[128]		✓	V	L	U		✓		
Gygli *et al.*, 2014	[47]	✓		V	L	S			✓	
Sun *et al.* , 2014	[116]		✓	V	L	S			✓	C
Chu *et al.* , 2014	[17]		✓	V	L	U			✓	C
Gong *et al.* , 2014	[42]		✓	V	DPP	S	✓		✓	
Kim *et al.* , 2014	[66]		✓	B	Graph	U	✓	✓		
Panda *et al.* , 2014	[95]		✓	SB	Graph	U	✓	✓		
Gygli *et al.*, 2015	[48]	✓		V	SO	S		D	L	
Potapov *et al.* , 2015	[103]		✓	V	L	S			✓	C
Lin *et al.*, 2015	[80]		✓	V	SVM	S			✓	
Yang *et al.* , 2015	[134]		✓	V	ED	U			RNN	C
Zhang *et al.* , 2016	[143]	✓	✓	B	DPP	U		✓	D	C
Xu *et al.*, 2016	[131]		✓	V	L	WS			✓	
Gygli *et al.*, 2016	[49]		✓	V	FNN	S			✓	
Marvaniya *et al.*, '16	[86]	✓		V	L	U	D		L	
Zhang *et al.* , 2016	[144]	✓	✓	B	DPP	S	✓	RNN		
Vasudevan *et al.*, '17	[127]		✓	SB	SO	WS	D	D	RNN	U
Zhao *et al.* , 2017	[146]	✓	✓	B	RNN	S		✓		
Mahasseni *et al.* , '17	[85]	✓	✓	B	GAN	U	✓	RNN		
Ho *et al.* , 2017	[54]	✓	✓	B	RNN	WS	✓	✓	ED	
Yu *et al.* , 2017	[140]		✓	C	RNN	U	✓	✓		
Panda *et al.* , 2017	[96]		✓	V	FNN	U			✓	C
Panda *et al.* , 2017	[94]		✓	V	ED	S		✓	✓	
Ji *et al.* , 2017	[60]	✓	✓	V	SO	U	D	D	D	C

Task: **S**tory**B**oard; **V**ideo skim; **B**oth; **C**aptioning; **F**ast **F**orward; **O**ther.
Model: **L**inear ML; **S**ubmodular **O**ptimization; **D**issimilarity measures; **E**ncoder-**D**ecoder
Learning: **S**upervised; **U**nsupervised; **W**eakly **S**upervised.

2.1.1 First Person View Video Summarization Framework

When approaching the summarization problem, different possible outputs are considered from storyboards to video fast-forwarding (*c.f.* Table 2.1a). The election of one output or another is generally task-oriented:

Static Story Boards Though being a minority, some works summarize egocentric videos extracting the highlights in the form of a set of key frames to obtain a sort of photo album [7, 75, 76, 129]. This is preferred for lifelogging data, where the input is a set of pictures instead of video recordings [11, 32, 33, 41, 74, 120].

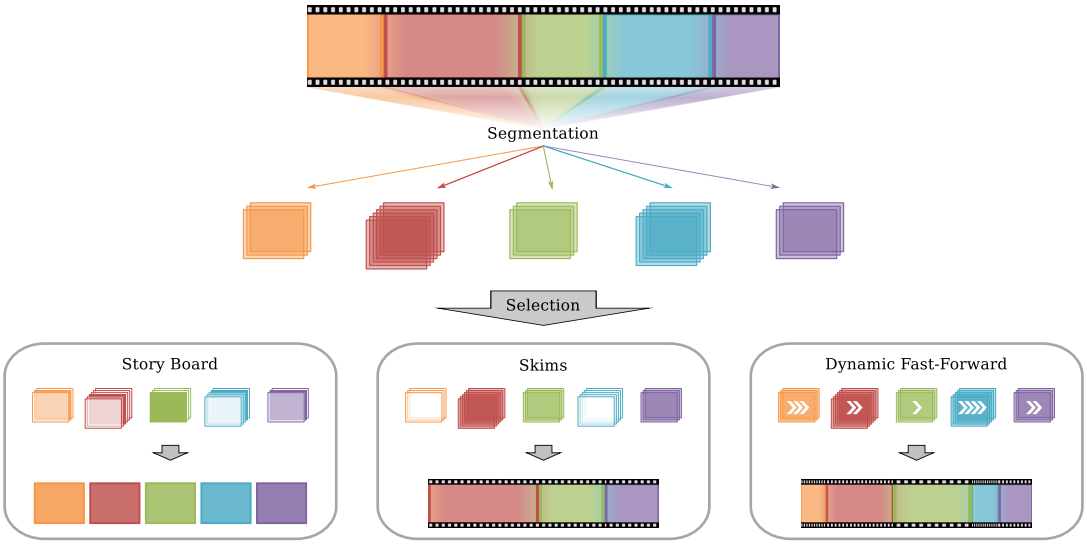


Figure 2.1 A general framework for egocentric video summarization.
The video is first segmented into scenes or subshots, from which: *a)* a key frame is extracted for story boards; *b)* the most relevant subshots are selected for skim summaries; or *c)* the subshot speed is determined for dynamic fast forwarding (traditional fast forwarding does not need prior segmentation).

Dynamic Video Skimming For personal recordings from head-mounted cameras (such as holiday or home videos), retaining the original video structure is generally preferred. The summary is done by selecting the most relevant segments of consecutive frames (subshots) to represent the full video [2, 47, 48, 80, 83, 91, 125, 130, 132, 145].

Fast Forward Despite the shaky nature of egocentric videos and its inherent challenge for a faster video browsing, preserving the whole video content is important for adventure and extreme sports videos. An stabilized fast forwarded version of the original recording is proposed for these cases [64, 70, 93, 101]. A variant of this is dynamic fast-forwarding, where each subshot in the video is played at a different speed [93], as opposite to traditional fast-forwarding, where a constant speed is set along the video.

Figure 2.1 shows the typical framework for each approach, which begins by segmenting the video into different events or small subshots, in order to select the most adequate ones afterwards. In the case of Lifelog summarization, the process is usually preceeded by a filtering of low quality pictures (*i.e.* blurred, dark or with presence of occlusions).

2.1.2 Summarization Objectives

Being able to record what we see without compromising our mobility or the use of our hands clearly opens a wide range of opportunities for law enforcement and security, healthcare, navigation and remote assistance, *etc.* In general, a summary should be uniform and diverse (it should convey a coherent story), and contain relevant keyframes or shots. However, the intention of the recording constraints the objectives to be met by the summary, and what is relevant for one application may not be for another. Here, some examples of applications of FPV summarization are listed, with the cues that would be relevant for them:

Security and law enforcement Currently, videos from body-mounted cameras are used as evidence of what happened on specific incidents. In a future, however, a summarization algorithm will be able to find behavior patterns and detect dangerous situations, to assess the police force beforehand, ignoring aesthetics or emotionally pleasing constraints.

Instructional video A summary of an instructional video should be complete, selecting each action change, and ignoring the intermidiate repetitive steps within the action.

Sports and adventure Systems to summarize videos of sports or adventure experiences should discriminate the thrilling or visually attractive shootings from shaky or dull ones.

Caregivers supervision and memory digitalization Daily recordings can be used to get an overview of life patterns, retrieve life events, or pass along our memories to the next generation. The summary objectives in each case vary from selecting a representative variety of actions and interactions to finding just one kind of activity (*e.g.* eating), or locationg emotions along the different events.

Celebrations and holidays In a summary of a personal life experiences, either as a short video or a photo album, we would expect to find happy faces, emotional moments, interactions with other people or animals, and beautiful scenery.

2.2 Event Segmentation

Whereas TPV segmentation approaches typically try to identify the shot boundaries comparing consecutive frames, or even using the frames' time-stamp [55, 89], this methodology cannot be directly applied over FPV, since egocentric videos consist of a single shot with extremely smooth transitions between consecutive events. Similarly, photo-streams entail a different

Table 2.2 Features, Segmentation and Selection Methods Used in the Reviewed Papers

		Image/video **features**							**Segmentation**					**Selection** (Objectives)						
		Low-level (e.g. color)	Hand-crafted	Saliency	Motion, blur	Deep learned	Objects/people	Metadata*	Color/GIST similarity	Ego-motion	Machine/Deep learning	Attention/fixation	Metadata**	Diversity/Uniqueness	Representativeness	Aesthetics	Importance/Interestingness	Attention (wearer)	Query (user input)	Notes
Aizawa *et al.*, '01	[2]	✓			✓			S	✓	✓								✓		
Ng *et al.*, '02	[91]	✓			✓			S	✓	✓								✓	✓	
Kitani *et al.* , '11	[67]		✓		✓					✓	✓									FPV action recognition
Lee *et al.*, '12	[76]	✓	✓				✓		✓				T				✓			
Lu *et al.*, '13	[83]	✓	✓		✓		✓		✓	✓				✓			✓			
Wang *et al.*, '14	[128]	✓			✓								U		✓	✓				Highlights major semantics
Xiong *et al.*, '14	[129]	✓	✓		✓			S	✓				T			✓				
Gygli *et al.*, '14	[47]	✓		✓	✓		✓			✓							✓			Rule-based
Zhao *et al.*, '14	[145]		✓		✓								U	✓						Unsupervised Dictionary
Okamoto *et al.*, '14	[93]		✓		✓								U				✓			
Poleg *et al.*, '14	[100]				✓					✓										
Lee *et al.*, '15	[75]	✓	✓				✓		✓				T	✓			✓			
Varini *et al.*, '15	[126]		✓		✓					✓		✓		✓				✓	✓	
Varini *et al.*, '15	[125]		✓		✓			S		✓		✓		✓				✓	✓	
Gygli *et al.*, '15	[48]	✓				✓							U		✓		✓			
Xu *et al.*, '15	[132]					✓		G				✓		✓	✓			✓		
Lin *et al.*, '15	[80]		✓										U				✓			Context-driven S-SVM
Poleg *et al.*, '15	[102]					✓					✓									
Bettadapura *et al.*, '16	[7]	✓	✓		✓			S	✓				L			✓	✓			
Jeong *et al.*, '16	[59]	✓	✓		✓				✓				T		✓	✓				
Zhang *et al.*, '16	[144]					✓					✓			✓	✓					DPP from LSTM embedding
Xu *et al.*, '16	[131]	✓	✓		✓								U		✓	✓	✓			
Yao *et al.*, '16	[136]					✓					✓						✓			C3D, VGG + ranked FNN
Marvaniya *et al.*, '16	[86]	✓			✓								U	✓	✓	✓				
Sharghi *et al.*, '16	[108]		✓					G					U	✓					✓	Seq-Hierarchic DPP
del Molino *et al.*, '17	[25]	✓			✓	✓				✓			T	✓	✓	✓			✓	Feat dist. + CRF
Plummer *et al.*, '17	[99]	✓				✓							U		✓		✓			[48] w/ vision-language feats
Sharghi *et al.* , '17	[109]		✓					G					U	✓					✓	Memory Net + DPP
Ho *et al.* , '17	[54]					✓							U		✓		✓			E-D embedding + biLSTM
Dimicoli *et al.* , '17	[29]					✓					✓						–			AC+ADWIN for LL
Bhatnagar *et al.* , '17	[8]	✓			✓						✓						–			FPV feat. representation

Different cues are used to define, segment and summarize the video, mainly according to color and motion properties. When selecting the relevant segments, a mixture of different objectives is maximized, including importance (predicted, from the wearer attention, or a particular user's query) in most reviewed papers.

* **G**aze; **S**ensor data. ** **U**niform Length; **T**emporal proximity; **A**ttention/Fixation; **L**ocation.

clustering problem due to their low time resolution: while many approaches to segment High Time Resolution (HTR) video use motion between frames to infer the wearer's activity, motion cues are not available in photo-streams.

2.2.1 Event Segmentation in High Time Resolution Videos

As can be observed in Table 2.2, FPV segmentation is still mostly based on raw features, not considering human or perceptual cues. The second column section of this table provides an overview of the cues used for event or subshot segmentation, either visual or from metadata. These cues go from deterministic length or temporal proximity to image processing techniques to attention analysis.

Egocentric HTR video is frequently segmented according to either visual similarity [2, 7, 25, 75, 76, 83, 91, 129, 132] (*e.g.* color, GIST, R-CNN hash, etc) or motion cues [2, 25, 47, 83, 91, 100, 102, 125, 126] (*e.g.* optical flow, blurriness). Motion features are generally used to predict the wearer's activity or attitude patterns and then segment the videos accordingly. Examples of these methodologies are the use of GIST difference over a given window [7]; similarity between the R-CNN hashes extracted from the fixation region [132]; the *Cumulative Displacement Curves* in [100]; a Super Vector Machine–Hidden Markov Model pipeline in [126]; a 3D Deep Convolutional Neural Network in [102]; and KTS by Potapov *et al.* [103], a kernel-based change point detection algorithm. Recently, many authors chose to segment the video deterministically, set to a specific number of frames or time [48, 54, 80, 86, 93, 99, 108, 109, 130, 131, 145].

2.2.2 Event Segmentation in Low Time Resolution Videos

While event segmentation is needed for a complete, informative and diverse summary that includes most life events in the recording, little work has been done to that effect in LTR lifelogging collections [9, 45]. Given the limited amount of annotated data, events are typically segmented using unsupervised techniques such as K-Means and other hierarchical clustering algorithms on visual cues (*e.g.* color, GIST features, CNN hashes) [12, 24, 31–33, 78, 79, 133].

An exception to these unsupervised methods is [39], in which a personal location classifier is trained for each user, and events are segmented according to changes in the wearer's location. Since these methods often ignore the semantic nature of the frames, Dimiccoli *et al.* [29]

propose defining the frames with semantic and contextual cues defined by CNN features and linguistic information. The relation between frames is assessed using a WordNet [88] based knowledge graph, and the event boundaries are found using a graph-cut algorithm integrating an agglomerative clustering. Such segmentation methodology relies on the cross-analysis of consecutive frames, and cannot detect change points between two events with heterogeneous visual content, nor ignore small and isolated visual changes within an event.

2.3 Selection of the Relevant KeyFrames or Subshots

Once the video is segmented, the next natural step is to select the most appropriate parts for the summary. This is done by maximizing a combination of objectives, as summarized in the last group of columns in Table 2.2. These are grouped into video coherence (such as diversity of events or temporal uniformity), and inherent importance of the segment, either from a general point of view (*e.g.* visual pleasantness), for the viewer (the user to watch the generated summary) or the wearer (the person recording the original video). Even if importance is the main target for most works, each objective has its shortcomings, and so the election of objectives must be consistent with the type of input data and the purpose of the summary. Submodular Maximization [48, 127] and DPPs [143, 144] have been used in the literature to balance the contribution of each objective.

Features such as color, SIFT, DoG, HoG or HoF are frequently used to analyze video coherence [48, 75, 76, 83, 93, 125, 126, 131, 145], as well as the use of deep features [23, 25, 48, 102, 132, 144]. Aesthetics is generally estimated using features such as color, SIFT, GIST and blurriness [7, 24, 78, 129]. Both the video coherence and quality assessments are normally driven by unsupervised learning techniques.

On the other hand, importance is generally predicted using supervised learning. Such predictors may be based on impersonal cues such as saliency [47], location [7], and people and object interaction [47, 75, 76, 83, 126], or Deep Learning on highlight detection datasets [54, 109, 136, 144]. When evaluating the importance from the wearer's point of view, the use of sensors [2, 91, 132] and motion analysis [100, 125, 126] has also been used. Importance can also been understood as the attention a human would pay to each segment. As such, Ma *et al.* [84] explore diverse cues for attention (*e.g.* motion, aural, music, faces) to better understand the video and its interestingness.

2.3.1 Video Coherence

To consider the whole, diversity is often defined as the dissimilarity between elements in the summary [76, 83, 125–127], and representativeness is often achieved by selecting instances the most similar to the rest of the video [25, 48, 83, 99, 127, 131]. The video frames or segments are usually described using hand-crafted features such as color histograms, GIST or SIFT, or deep learned features. Uniformity (temporal coherence), on the other hand, is generally related to the frame index [48, 76, 99]. Albeit considering the overall narrative, following these principles alone may lead to summaries of poor visual quality, even if informative enough, or of very weak interest.

Zhao *et al.* [145] force diversity and representativeness via a dictionary of video sentences (using HoG and HoF features). If it is impossible to reconstruct it by using the learned dictionary, the segment is added to the summary and the dictionary is updated with the new features. Xu et al.'s algorithm [132] encourages both representativeness and diversity by maximizing the entropy of the segments' descriptors (defined with R-CNN hash computed around the frames' centroids). Using a skeleton graph, Panda *et al.* [95] use random walk to find summaries of scalable length. A graph representation based on feature distances is also used by Kim *et al.* [66] to summarize photo streams leveraging on video story-telling sequences.

To achieve history coherence while constraining the quality or relevance of the elements, Submodular Maximization [48, 127] and DPPs [143, 144] have been used. Alternatively, a combination of contrast, blur, and other image quality assessment may be used to remove bad-quality elements from the pool of elements to be summarized [9, 12, 24, 59, 78]. Frames may then be clustered according to their SIFT or deep feature sparse coding, where the best quality and most representative of each cluster is selected as a key-frame.

Since DPPs consider each element as randomly permutable, they fail to capture the inherent sequential nature of video content. To overcome this deficiency, Gong *et al.* [42] propose a sequential DPP that uses supervised learning. The transformation matrix to be applied on the hand-crafted features, as well as the parameters of the neural network if using a non-linear hidden representation, are trained with user-generated summaries.

2.3.2 Segment Importance

The importance value of a frame or segment can be estimated for any user with generic predictors that do not consider the wearer's or specific viewer's interest. However, the absolute importance of each segment is context dependent, and cannot be equally estimated for, e.g., extreme sports and law enforcement video. As such, different cues are used for training based on the method's final objective.

To achieve this requisite, most methods use supervised learning. Since obtaining the annotation needed for such endeavor (*e.g.* human-generated summaries) is usually hard and costly, recent approaches leverage on edited content to train weakly-supervised models.

Supervised Learning The interestingness can be predicted from regions containing important people and objects, using low-level features (*e.g.* SIFT, DoG, region sizes) [48, 75, 76]. Alternatively, the presence of factors such as saliency, edges and colorfulness, object interactions, and the presence of landmarks, people or faces can be used to predict each frames' interestingness [47]. Similarly, [7, 62, 86, 129] base their quality or intentionality prediction on cues for composition such as picture alignment or accelerometer data; the artistic *rule-of-thirds*; symmetry (on local SIFT features) and color vibrancy; head tilt; camera motion; etc.

Context dependent, Okamoto *et al.* [93] train a model for navigation instructions using a crosswalk detector and ego-motion cues. Pairs of raw and edited videos are used in [116] to train a different SVM highlight detector for each considered context. The authors of [80] follow the same idea, training jointly both context and highlight structured SVM models, so the algorithm can adapt according to the detected context.

Similarly, recent methods for highlight detection have used deep neural networks with ranking loss to score interesting segments higher than non-interesting ones [49, 136]. While Gygli *et al.* [49] train the net with only spatio temporal features (C3D), Yao *et al.* [136] also consider the spatial information. Their model fuses the highlight estimation of two different ranked FNN models: one trained on AlexNet CNN features (objects and animals) and another on the C3D output.

Unsupervised and Weakly-supervised Learning Chu *et al.* [17] propose a Maximal Biclique Finding algorithm to find co-occurences between the target video and edited content (disregarding the frequently unavailable original videos). Zhang *et al.* [143] infer the optimal

summary using a determinantal point process (DPP) fed with a kernel matrix. Such matrix is the result of modeling the unseen video according to the pairwise similarity to the training examples from same-category edited content.

Context-dependent, Panda *et al.* fine-tune a 3D-CNN model trained on VCF101 to learn highlights from their presence in same-context web videos [96]. In [94], C3D features are used to predict the highlights of a video given sparse optimization on the reconstruction error for videos of the same kind. To ensure that the summary is also diverse and representative of the given video, a joint maximization problem is proposed using the summary reconstruction error.

2.3.3 Deep Summarization Architectures

Recent methods using deep learning predict the interestingness of each segment and the most coherent summary jointly with end-to-end training.

Supervised Learning In [144] the video segments are encoded with a bi-directional LSTM network. Two approaches are explored. In the first, $vsLSTM$, one hidden layer is used to predict the relevance of each frame from the LSTM encoding. In the second, $dppLSTM$, the output of $vsLSTM$ is used to weight a separate representation of the bi-LSTM embedding, which is then used as input for a DPP. To reduce the complexity of $vsLSTM$, Zhao *et al.* [146] propose aggregating the frames in each segment using a previous LSTM layer (most works simply average the frames' descriptors).

Ji *et al.* [60] use a biLSTM-based encoder-decoder architecture to achieve representativity in the generated summary. They define the frame importance as the level of attention. To model the attention of each video frame, a hidden layer is included between the encoder and the decoder. Frame-level importance annotations are used to train all parameters.

Semi-supervised Learning Given that the amount of FPV datasets is very limited, and the annotated summaries almost inexistent, Ho *et al.* [54] train a model using triplets of highlight and non-highlight samples from TPV annotated videos, a few FPV annotated samples, and many unsupervised FPV examples. An FNN encoder-decoder is first used to embed the input C3D feature into an space in which a highlight segment will be closer to other highlights than to non-highlights. As such, the loss function evaluates the distances between the embedding of a given FPV positive/negative segment and that of both a FPV/TPV highlight and non-highlight.

The embedded feature is then concatenated to the C3D feature and fed into a biLSTM, which explores the sequential nature of the video to predict the best summary with a single hidden layer scoring network. Unlabeled data is used to fine tune the parameters, mapping each segment as positive or negative according to its representation in the embedding space.

Yu *et al.* [140] encode each frame within the album using a GRU-RNN, then an FNN is used to predict the relevance of each frame to generate a caption of the photostream in 5 sentences. GRU is also used as unit in a Bidirectional Multi-thread RNN to preserve the sequential information in photostreams [81]. A variation of such model, but preserving the history coherence within a paragraph is coined Bidirectional Attention RNN in [82]. Park *et al.* [18] use a context sequence memory model with ReLU to personalize the image caption to each user's previous vocabulary.

Unsupervised Learning The method in [134] uses a context-dependent LSTM-based autoencoder (trained with edited videos) to predict the highlight score of the segments. For the highlights to also be representative of the video, the reconstruction error is assessed. Similarly, Mahasseni et al. propose a sequential generative adversarial framework, consisting of a summarizer and a discriminator [85]. The summarizer is an LSTM that outputs the scalar prediction of the relevance of each frame. An LSTM-based encoder-decoder is then used to reconstruct the original video, which is then fed to the discriminator.

2.4 Personalization

The aforementioned methods share a common limitation, as they generate generic summaries. This limits their potential performance, as not all users share the same interests [113] and are thus editing video in different ways [25]. One user may edit basketball videos to extract the slams, another one may just want to see the team's mascot jumping. A third may prefer to see the kiss cam segments of the game. An automatic method should therefore adapt its results to specific users.

The summary can be customized either for the person viewing the summary, or from the wearer's point of view. User profiles [51, 58, 118, 125], queries [108, 126, 135] and previous user's behaviour [87, 97, 139] can be used to adapt the summary to the viewer. On the other hand, EEG signals [2, 91], gaze [132] and motion patterns [125, 126] have been used in the literature to infer the wearer's interest.

2.4.1 Passive

Without active user input at inference time, personalization can be achieved by training with user-specific information acquired unobtrusively. Using annotated meta-data (rather than using the audio-visual inputs only), [1, 4, 58] build a user profile and use it for personalization.

2.4.2 From User Input

Query oriented, both [23, 130] propose systems that can retrieve subshots from the stored videos given a video [23] or a story-based query [130]. Sharghi *et al.* [108] estimate the relevance of the video segments given a user query in the form of text, given 1, 2 or 3 concepts (*e.g.* car, flowers). In [127], the relevance to a text query is given by the distance between the semantic embeddings, via an LSTM, and a quality score given by a CNN model. In [125, 126], the summary is personalized by looking for scenes relevant to the cultural interest of the viewer or the wearer, using DBpedia. Two recent interactive approaches [25, 112] require the user to give feedback on individual proposals [25] or pairwise preferences [112], which is then used to propose a refined summary.

Given a text query, Sharghi *et al.* [108] find the best summary using a Sequential and Hierarchical DPP. SH-DPP improves DPP in that it preserves the sequential structure of the input video, and allows for larger videos. In [109], the SH-DPP is fed with the output of a memory network embedding the frames with the user query. Both the DPP and the memory network's parameters are trained using user summaries in the training set.

2.4.3 From the Wearer's Perspective

Importance can also be inferred from the wearer recording patterns (such as time spent at a certain place, or interacting with a certain item or person) [125, 126]; from physiological data (such as gaze and brain waves) recorded alongside the video [2, 91, 132]; or the user's behavior while visualizing similar content [87, 90, 97, 139]. Supervised learning is used to train these models.

2.5 Evaluation Methodology

There is no such a thing as an objective best video–summary ground truth: each person may like a summary differently, and several segments could be interchangeable whithin a video summary. Moreover, video summaries are generally task-dependent, and they should be evaluated differently according to the intention. Therefore, the evaluation of video summarization is still a great challenge in video analysis. When evaluating FPV video summaries, the personal nature of the original recording makes it even more complicated: who should be the judge of the summary, who should annotate the key items? The wearer, the viewer, or both? Their understanding of a good summary may be completely different.

Unlike for image memorability [57], for egocentric summarization there is no empirical evidence showing that inter-subject consistency is actually relatively high. To the contrary, Gygli et al. [47] evaluated the "human performance" when summarizing their SumMe dataset, computing the F-measures of each human summary against all the other participants', achieving measures between 0.1 and 0.5 –mean of 0.25– on their egocentric videos. Moreover, the problem with objective ground-truth based evaluation is that it may not properly reflect what users truly want from a summary. Perhaps because of this, methodologies are evaluated as a relative comparison with other techniques, not considering the standalone performance.

Video summaries can be objectively evaluated by measuring precision and recall over the presence of key objects, people or events [20, 76, 99, 108, 109, 129, 132, 136, 145], or using Natural Language Processing (NLP) techniques with textual annotation of the whole video parts [59, 132] (refer to the datasets presented in Chapter 3). However, these key items and video parts must be previously annotated according to subjective criteria. Thus, the evaluation is not totally objective. Because of this, most authors conduct users studies to evaluate the proposed methods in a subjective way [7, 75, 83, 93, 125, 136, 143, 144]. In such cases, the evaluation can be crowdsourced using platforms like Amazon Turk and CrowdFlower, where a large and diverse number of users can annotate comparisons between two given summaries [43].

Chapter 3

First Person View Video Datasets

This Chapter describes the most relevant datasets for egocentric video summarization. The datasets must be recorded by people with head or chest mounted cameras, in totally unconstrained environments, and long enough as to compress a wide variety of sub-activities. Three large scale datasets, namely CSumm, R3 and PHD^2, are then introduced, as well as insights into the interests and video highlight preferences of over $15,000$ *anonymous subjects.*

3.1 Introduction

There is a small but growing number of datasets available for egocentric video analysis. Among them, many are not suitable for FPV summarization, since for this purpose they must contain videos recorded by people with head or chest mounted cameras (to see exactly what the wearer sees even with subtle sight movements), in totally unconstrained environments, and long enough as to compress a wide variety of sub-activities. Even if recorded with head-mounted devices, this is the case of CMU-MMAC [21], recorded in a staged kitchen; GTEA [36] and GTEA-gaze [38], which contain very specific videos, sometimes staged and short; and UEC [67], which is a compilation of short videos from YouTube and recordings of choreographed activities.

Table 3.1 lists the most popular datasets for egocentric video understanding and summarization, including the number of videos, wearers and typical length; the year of release, original task for which they were recorded and the available annotation; and the works in which they have been used.

3.1.1 Public High Time Resolution Video Datasets

The most suitable (and publicly available) egocentric datasets for the task of HTR video summarization are the following:

Table 3.1 First person view video datasets used for the summarization task.

Dataset name	Year	Device	Short description	Num. subjects	Num. vids	Typ. length	Original task: Summary	Original task: Activity rec.	Original task: Other	Annotation
UT Ego [76]	'12	Looxcie	Unconstrained videos of natural daily activities	4	10	3-5h	✓			- People and objects: text and boundary * [138] adds text annotation
ADL [98]	'12	Go Pro (chest)	Predefined set of actions at home	20	20	30'		✓		- 18 actions - 42 objects data
GTEA-gaze+ [38]	'12	SMI eye-tracking	Cooking in a natural setting	10	30	10-15'		✓		- 100 actions - Gaze metadata * [132] adds annotation for summarization
FP Social Int. [37]	'12	GoPro	Experiences during a day at DisneyWorld	6	8	6-8h			✓	- Social interactions * [100] adds motion annotation for segmentation * [138] adds text annotation
Huji EgoSet [100]	'14	GoPro	Different activities in unconstrained settings	3	37	6-30'			✓	- 7 motion classes
Microsoft's [70]	'14	GoPro	Videos of different sports	1	5	3-13'	✓			(none)
EgoSum+gaze [132]	'15	SMI eye-tracking	Daily life in unconstrained setting	5	21	15'-1.5h	✓			- Summarization: 5-15 sets of segments / video - Gaze metadata Not Released
[125, 126]	'15	*head mounted*	Recorded by tourists visiting cultural spots	12	12	30'	✓			- 6 motion classes Not Released
[130]	'15	*egocentric camera*	Videos taken at Disneyland in unconstrained setting	10	10	>5'			✓	- Locations and events Not Released
[7]	'16	Contour Cam	Unconstrained videos taken during a vacation trip	1	26.5h		✓			- Diverse metadata Not Released
CSumm		Google Glass	Unconstrained activities ranging from lunch to holidays	3	65	15'-1h	✓			- Manual annotations for 10 videos, objective oriented
EDUB [119]	'15	Narrative Clip	Unconstrained daily activities at 2 frames per minute	7	66	1k	✓	✓		- Events segmentation
CLEF [20]	'17			3	79	1k	✓	✓		- 51 topics
NTCIR [44]	'17			2	91	1k	✓	✓		- 39 topics
R3				30	1.7k	1k			✓	(none)

- **UT Egocentric** [76]: Originally recorded to summarize FPV based on the presence of important objects and people, this dataset contains long videos of many different daily activities. Unlike all the other reviewed datasets, this one was recorded at low-quality frame rate (15fps), and the video data includes objects annotation.

- **Activities of Daily Living** [98]: To record this dataset, users were asked to perform a set of pre-defined activities at their homes in a continuous way, wearing a chest-mounted GoPro camera. Videos are densely annotated with objects, interactions and the actions performed. It is, therefore, a suitable dataset to evaluate context-based egocentric summarization through user studies or the presence of key objects and people.

- **GTEA-gaze+** [38]: Intended to improve the data collected for GTEA-gaze, this dataset contains videos of subjects preparing meals out of 7 different food recipes in a natural kitchen setting. The dataset is recorded with SMI eye-tracking glasses and contains annotations for around 100 different actions. Summarization annotation was added later by [132]. However, due to the constrained nature of the videos, it is only suitable to evaluate summarization approaches tailored to cooking settings.

- **First Person Social Interactions Dataset** [37]: This dataset was originally recorded to evaluate social interactions during a full day at an amusement park. Due to its highly unconstrained nature and the long duration of its videos, it can be used to evaluate the egocentric summarization task through user studies. The different social interactions are annotated, as well as the events that happen within the video. Most social interactions are longer than 16 seconds, with only 10% longer than a minute. On the other hand, most events are longer than one minute, with 10% over six minutes.

- **Huji EgoSet** [100]: This dataset was recorded to test motion segmentation on any kind of activity, location, and illumination setting. It is in continuous development and, to this date, contains 44 videos of unconstrained daily activities (driving, chilling, walking, etc) taken with head mounted GoPro cameras. It also includes egocentric videos extracted from Youtube. The videos are annotated with motion and activity patterns, and 14 of the videos contain more than two different events. Most of these events are longer than 40 seconds, with 10% over four minutes. This dataset can also be used to evaluate the egocentric summarization task through user studies.

- **Microsoft's sports dataset** [70]: Used for fast-forwarding objectives, it was recorded

with a GoPro camera on a helmet and includes adventure activities such as mountain biking or climbing.

In addition, **SumMe** [47] is a dataset of consumer and egocentric videos annotated for the summarization task. It includes four egocentric videos, of about two minutes long each. As such, these videos are too short to be useful to evaluate summarization methods designed for longer personal egocentric events.

As can be observed, this is a very limited resource, and extensive work towards a unified benchmark for a wider range of tasks is needed.

3.1.2 Public Lifelogging Datasets

A wide effort has been done by the community to curate extensive and diverse Low Time Resolution datasets. The LTR datasets listed in Table 3.1 have all been recorded with the Narrative Clip, a wearable camera that takes one picture every 30 seconds:

- **EDUB-Seg** [119] and **EDUB-SegDesc** [13]: These two datasets have been recorded in Barcelona by 7 different subjects, adding to a total of 66 days, both working days and holidays. Annotations for the boundaries of the existing life events are provided, according to up to three different annotators. The events are defined as "a semantically perceptual unit that can be inferred by visual features, without any prior knowledge of what the camera wearer is actually doing" [119]. For the main annotator, most events in the dataset span longer than 20 minutes, with 10% of the events being longer than 42 minutes.

- **CLEF** [20]: This dataset "consists of data from three lifeloggers for a period of about one month each. The data contains $88,124$ [images], an XML description of 130 associated semantic locations (e.g. Starbucks cafe, McDonalds restaurant, home, work) and the four physical activities: walking, cycling, running and transport of the lifeloggers at a granularity of one minute" [20]. Annotations for 51 life events and activities are provided.

- **NTCIR** [44]: This dataset contains recordings from two users during one and two months each, for a total of $114,547$ images. During that time, the subjects traveled and worked at different locations. They also annotated metadata for each day, including what they ate, drank, and listened to, as well as 138 locations and physical activities.

Biometrics data such as heart rate, galvanic skin response, steps, daily blood pressure, weekly cholesterol *etc.* is also provided. For training and evaluation purposes, it includes annotations for 39 semantic life events.

However, the size of these publicly available LTR datasets is very limited: 170 days in CLEF [20] and NTCIR [45], and 66 in EDUB-Seg [29] and EDUB-SegDesc [13], spanning a total of $2,700$ hours and $261,845$ images. We can also resort to other popular HTR FPV video datasets such as the First Person Social Interaction Dataset [37], Huji EgoSet [100], and UTEgocentric [76], that cover 28, 15 and 16 hours, respectively. Down-sampled at $2fpm$, the accumulated length of these datasets is under $10,000$ images. This amount of information is insufficient to train effective deep learning models.

3.2 CSumm: an Extensive Dataset for Egocentric Video Summarization

Since current public datasets that provide annotations of the summary contain 1 to 5 minutes videos (*e.g.* SumMe [47], MED [103]), and video summarization is of most use for longer videos, we have obtained annotations for 10 shots of 15 to 30 minutes from CSumm, a novel dataset presented in this thesis.

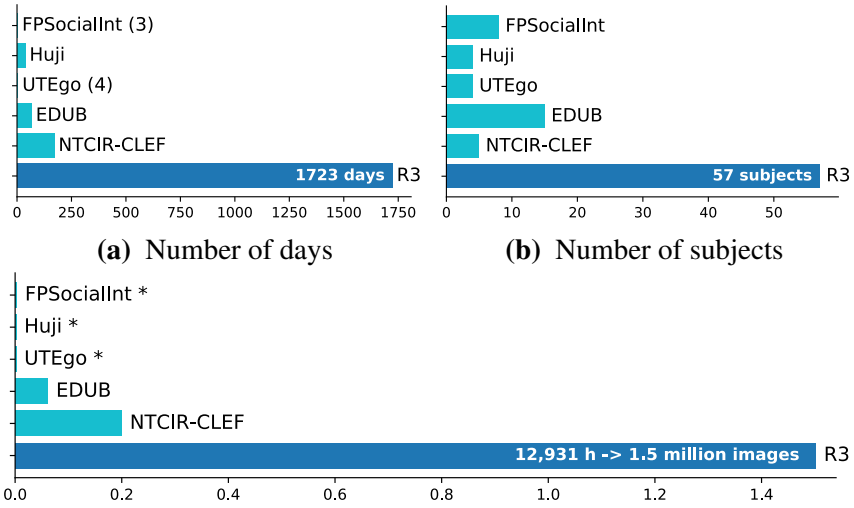
CSumm was recorded from October 2014 to March 2015 with a Google Glass (29 fps, with resolution of 720×1280 pixels). Due to hardware constraints (battery life and overheating), each recorded session is within one hour long, and the videos are saved in segments of 15 minutes. CSumm includes 202 such segments, from a total of 65 recording sessions. Most recording sessions are between 30 and 45 minutes long, and add up to a total of more than 40 hours of egocentric video. Alongside the video, a metadata file is generated, containing information from the sensors of the Google Glass such as timestamp and GPS. Those sensors are triggered at a frequency of $10Hz$.

The videos were taken by three different subjects during office hours, weekends and holidays. As such, they are of unconstrained nature and depict a large selection of activities, being a wide representation of the wearer experiences during the said time: lunch at the office, dinner with friends, commuting to and from home, practicing or watching sports, enjoying nature, *etc.* The videos include a wide range of viewpoints and motion, as they are FPV and a large amount of irrelevant moments alongside the recording. This makes the dataset challenging for video summarization, which is supported by our results in Chapters 5 and 7.

3.3 R3: a Large-scale Dataset for Lifelog Analysis and Understanding

Since the size of current publicly available LTR Lifelogging datasets is very limited, this thesis introduces *R3*, which is a large scale lifelogging image dataset suitable for deep unsupervised learning. It has been captured by 57 users during 1,723 days for a total of almost 13,000 hours, resulting in over 1.5 million images. A comparison of the size of *R3* with respect to the other Lifelogging datasets is presented in Fig. 3.1.

The users volunteered to capture their daily lives as part of a memory-enhancement user study. They were asked to put on the wearable camera for most of their day during a whole month, and were free to withdraw from the study if they felt that wearing it was disrupting their routines. The volunteers are mostly seniors older than 50 years old and span a wide range of occupations and lifestyles. To protect their privacy, only the extracted visual features have been released.



(a) Number of days **(b)** Number of subjects

(c) Total amount of images (in millions).

Figure 3.1 Comparison of *R3* with respect to other popular FPV datasets.
* HTR datasets have been down-sampled at $2fpm$.

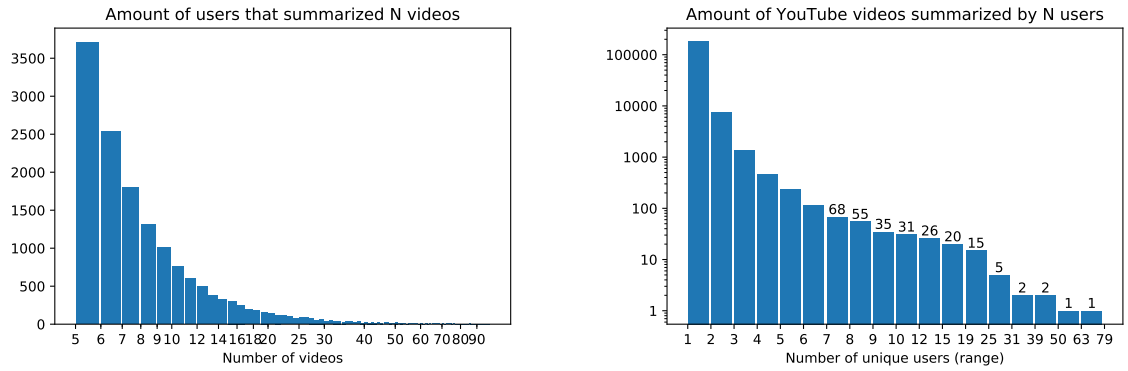
3.4 PHD^2: a Dataset for Video Highlight Analysis

One key challenge when training personalized models is obtaining a training set that provides useful user information. Different kinds of user information are possible, *e.g.* meta-data on the

user's age, gender, geographic location, what web editor was used and so on. To that end, PHD[2] contains information on what video segments a specific user considers a highlight. Having this kind of data allows for strong personalization models, as specific examples of what a user is interested in help the model obtain a fine-grained understanding of that specific user.

To obtain personalized highlight data, we have turned to *gifs.com* and its user base. *Gifs.com* is a video editor for the web and has a large base of registered users. When a user creates a GIF, *e.g.* by extracting a key moment from a YouTube video, that GIF is linked to the user. This allows to query for user profiles for users which have created several GIFs, *i.e.* contain a history that describes the user's interest. To have a reasonably sized sample of the users of interest, we restricted the selection to users that created GIFs from a minimum of five videos.

More than 15,000 users on *gifs.com* fulfilled these conditions. The dataset contains more than 350,000 annotations from 190,000 YouTube videos [1]. This is a significant leap with respect to other popular datasets such as the YouTube video highlight dataset [116], which contains about 4,300 annotations, and the Video2GIF dataset [49], which includes 100,000 annotations in the form of GIFs. The distributions for the number of videos per user in the full dataset are shown in Figure 3.2a. Note that a user may generate more than one GIF from the same video, and thus the total amount of GIFs is greater than the number of videos.



(a) Number of videos per user. **(b)** User queries per video.

Figure 3.2 Personalized Highlight Dataset in numbers. Most users in the dataset subset have made GIFs from more than 7 videos (3.2a). However, we observe that users, in general, have very different preferences in the videos they are interested in (3.2b). Out of almost 200,000 unique videos in the dataset subset, less than 10,000 have been queried by more than one frequent user. On the other end of the spectrum, two unique videos have been queried by 60 and 73 unique users, respectively.

[1] At the time of this report 36,000 of those videos have been removed from the platform, thus having almost 300,000 gifs from over 150,000 valid videos

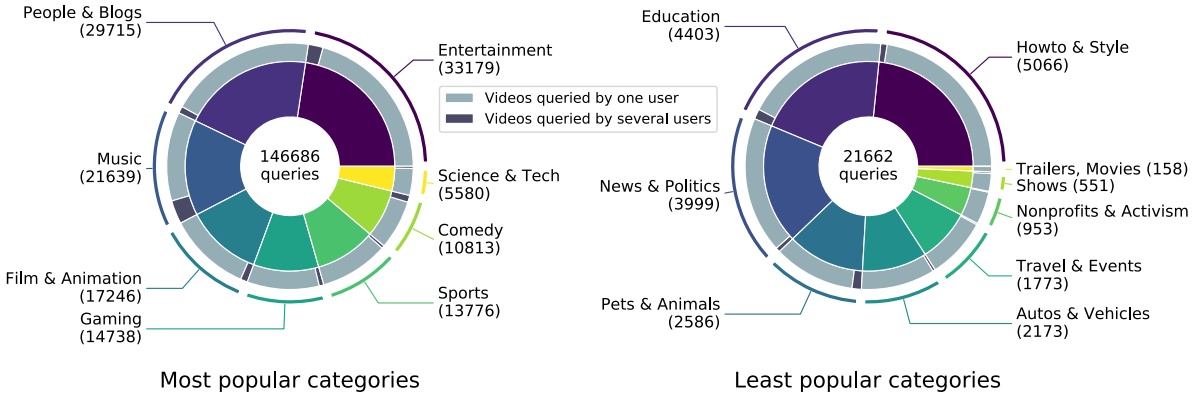


Figure 3.3 User queries per category. The most popular category is Entertainment, while very few users want to extract GIFs from Trailers, Movies or Shows.

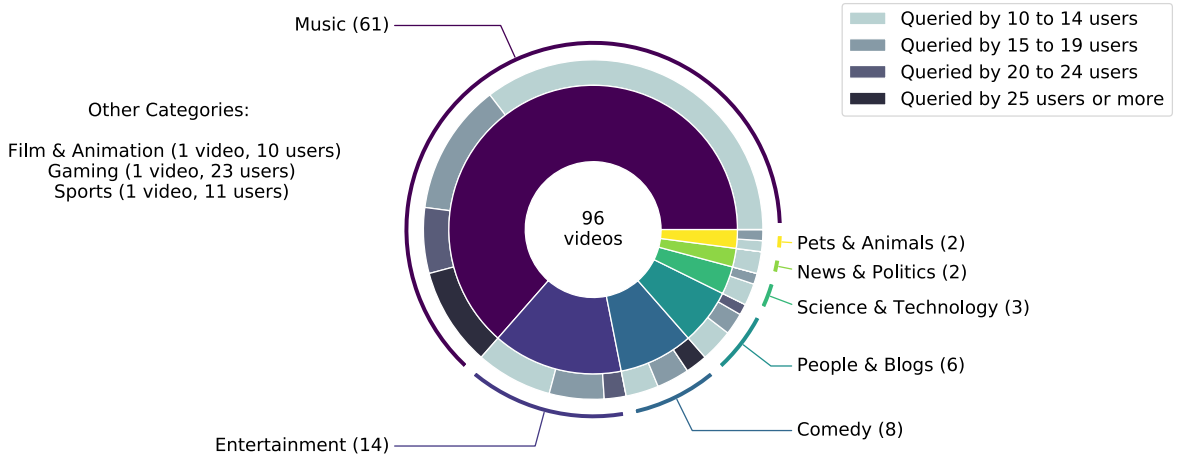


Figure 3.4 Category of the most popular videos. Most of the videos queried by at least ten users belong to the Music category. Some of the Comedy videos have also been queried by more than 25 users.

3.4.1 What things are generally more interesting?

Some videos are extremely popular, being queried by up to 73 individual users (*c.f.* Fig. 3.2b), whereas in general (95% of the annotated videos) videos are of interest only to one frequent user in the dataset. While the most popular category in YouTube for our dataset is Entertainment (*c.f.* Fig. 3.3), the most popular videos (queried by at least ten users) mostly belong to the Music category (*c.f.* Fig. 3.4). This is consistent with YouTube statistics, since Entertainment is a much broader category than Music but music videos are seen by a larger population than entertainment videos [2].

[2] `https://www.digitalmusicnews.com/2016/08/16/music-5-percent-youtube/`

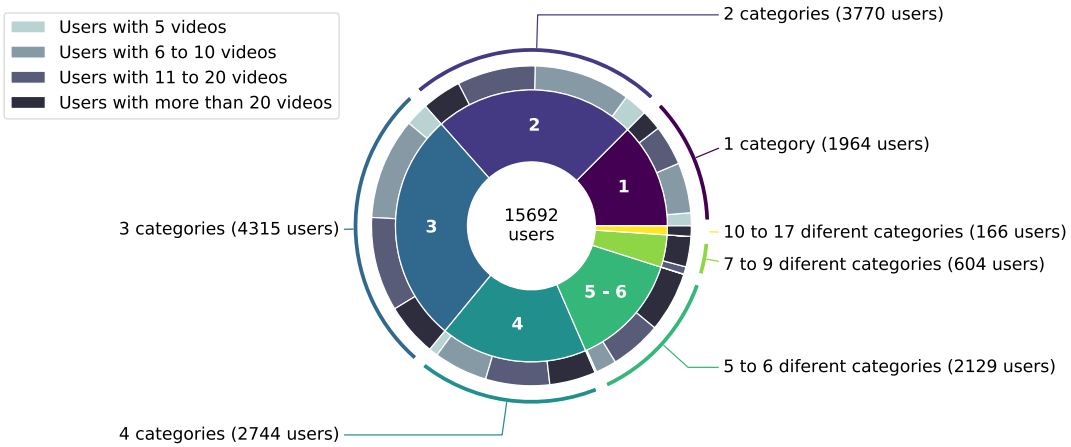
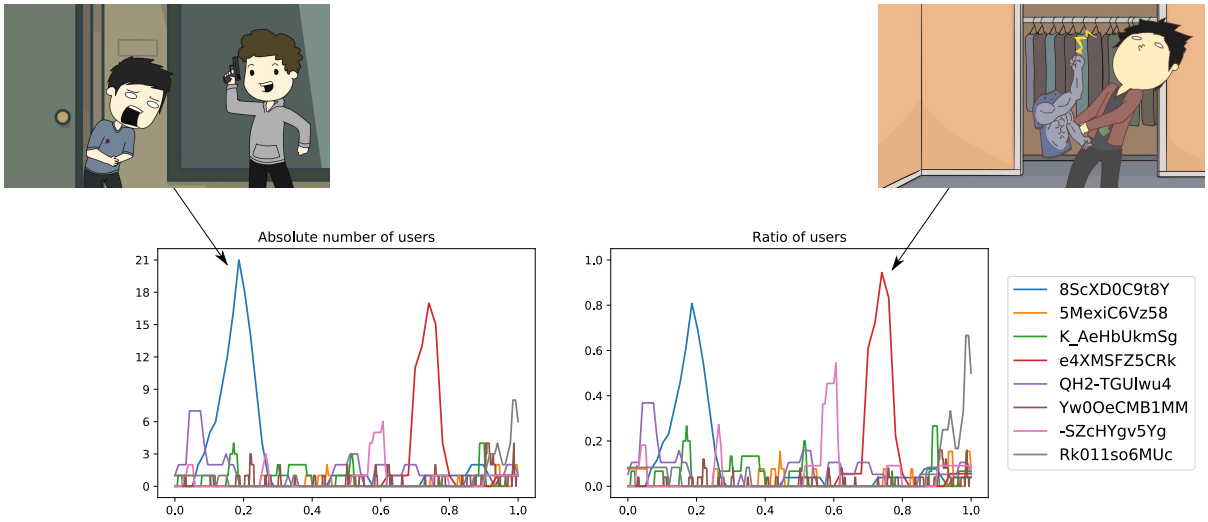


Figure 3.5 Number of categories browsed per user. Most users have made GIFs from three or less categories.

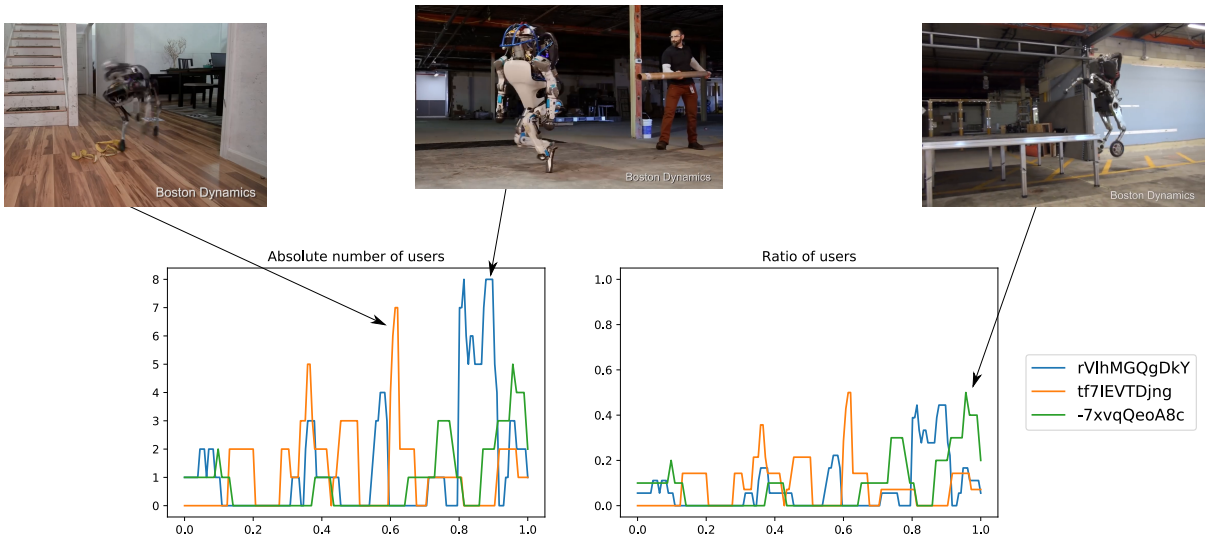
When analyzing the GIFs users made from these popular videos, we can observe that the most generalizable categories are Comedy and Science and Tech (*c.f.* Figure 3.6). In Comedy videos, the main gag is usually quite prominent, thus the consistent selection of such segment as a highlight. Boston Dynamics' YouTube channel is a very popular one, with people wanting to see how their robots slip, fall and get back on their feet, thus making it a very consistent highlight selection. On the other hand, for the most popular categories, users tend to have very diverse preferences of highlights (*c.f.* Figure 3.7). Music videos tend to be repetitive, and thus users may select any of the multiple segments in which the singer is dancing, for example. Even if a model could generalize in practice for that category obtaining good user feedback, automatic off-line evaluation methods will not be accurate at predicting the model's performance. The same happens for Entertainment and People and Blogs videos, in which the major user agreement happens for funny segments or famous people.

3.4.2 Are users consistent in their preferences?

As can be observed in Fig. 3.5, most users have made GIFs from three or fewer categories. While this would be normal for users with only five videos queried, most of those users have made GIFs from at least 11 videos, which shows they are consistent in the category they are interested in. Moreover, most of the users that have queried more than ten videos queried from less than four categories.

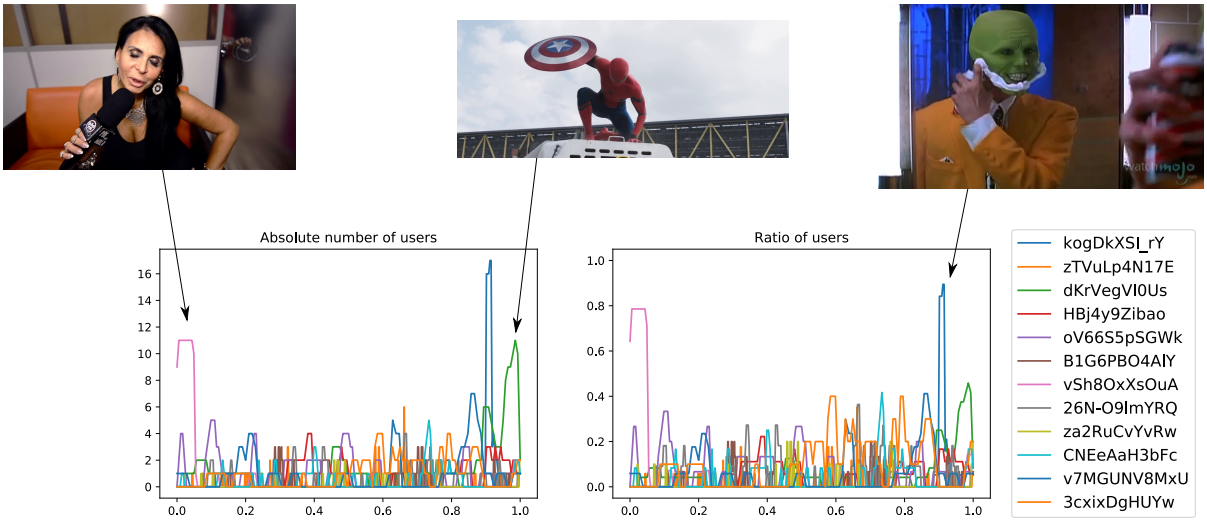


(a) Amount of users selecting each moment for Comedy.

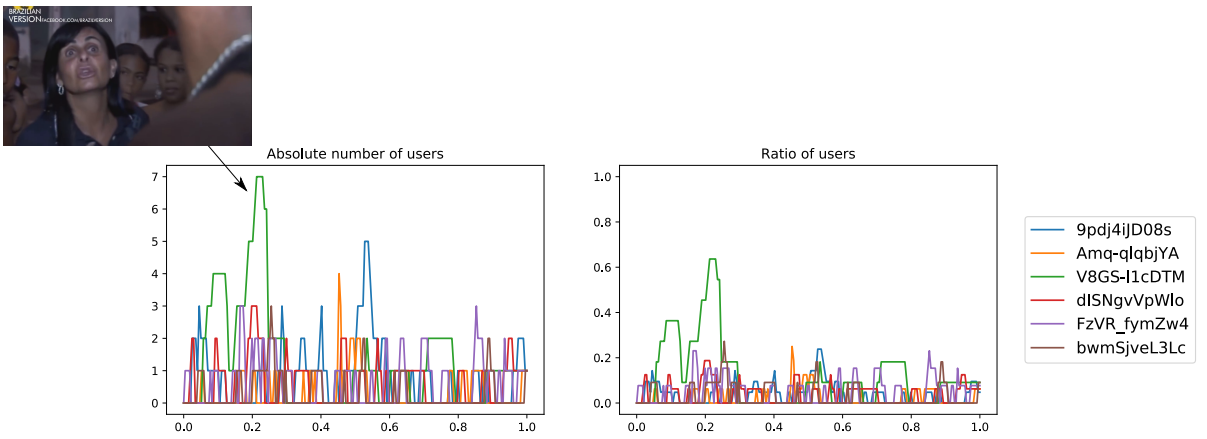


(b) Amount of users selecting each moment for Science & Technology.

Figure 3.6 Moments which several users agree are highlights.



(a) Amount of users selecting each moment for Entertainment



(b) Amount of users selecting each moment for People & Blogs

Figure 3.7 Moments that the users selected as GIFs from the most popular YouTube categories. We can observe that, in general, users have quite diverse interests for these two categories.

Chapter 4

Event Segmentation in First Person View Video

Segmenting visual content into events provides semantic structures for indexing, retrieval, and summarization. Current methods to temporally segment egocentric or continuous video data usually compare visual features between frames in an unsupervised way or analyze motion features. This Chapter presents Contextual Event Segmentation (CES), an episodic event segmentation method that is able to detect boundaries between heterogeneous events and ignore local occlusions and brief diversions. CES improves the performance of the baselines by over 16% *in f-measure and is competitive with manual annotations.*

4.1 Introduction

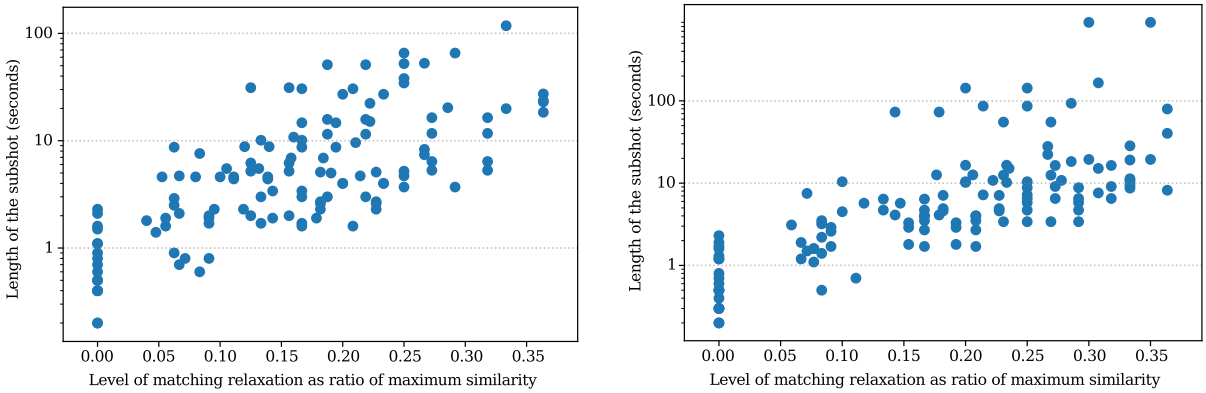
Segmenting video content into events provides semantic structures for indexing, retrieval, and summarization. TPV event segmentation approaches typically identify shot boundaries by detecting abrupt changes between consecutive frames [55, 89]. However, FPV content is not comprised of separate shots, but rather a succession of events with smooth transitions, where event boundaries are not well defined.

Event segmentation methods for High Time Resolution FPV video frequently use motion cues, both visual (*e.g.* optical flow, blurriness) [2, 25, 47, 83, 91, 100, 102, 126] and from sensors [2, 25, 114]. In the case of lifelog photo-streams (*i.e.* Low Time Resolution videos), frames can be up to 30 seconds apart. In such low temporal resolution, visual motion information is unavailable. Moreover, content may change a lot between consecutive frames even if they are part of the same event.

Visual similarity between groups of frames (*e.g.* color, GIST, CNN hash) is commonly used to detect activity changes [2, 12, 24, 32, 33, 75, 76, 78, 79, 83, 91, 103, 133]. While these methods are effective at detecting action change points, the granularity of such "activities" is much higher than that of episodic events. Therefore, such methods are not effective at separating one life episode from the next.

Query-based Event Segmentation One way to segment the visual content according to life events is to define the shot by what is really going on in the scene, who is part of it, or when and where it happened. Most works fail to provide a full semantic segmentation of the indexed video. Video units can be indexed and labelled using semantic features such as *day and time*, *illumination*, *location*, *motion*, *people*, *objects*, *scenario* and *situation* [23]. A shot can then be defined as the set of consecutive units (*e.g.* frames) sharing the same relevant labels. These *relevant labels* can be variable, and depend on each specific query, either video, image or text. The *relevant labels* are thus the tags shared between the given visual query and the stored video unit.

This methodology has been tested on a subset of CSumm (*c.f.* Chapter 3), for which shots were defined according to visual queries, *i.e.* video clips or images. This condition leads to segments of less than one second, on average, for the stored video most similar to the query. Slightly longer segments are retrieved from the second match, as to be expected given that the number of shared tags will be inferior and fewer changes between consecutive units are to be expected. Depending on the application, a longer video segment might be preferred. To obtain longer segments, different degrees of matching relaxation can be applied, such that not only the most similar frames are selected. Those frames with a similarity value inside the range of relaxation are also considered, obtaining a set of very similar consecutive shots. As shown in Figure 4.1, the greater the relaxation, the longer the retrieved segment. Needless to say, longer segments tend to include more irrelevant semantic meaning to the retrieved memory as intended by the query.



(a) Subshot length for the closest video.

(b) Subshot length for the second closest video.

Figure 4.1 Longest stream of consecutive frames for different levels of matching relaxation. A matching relaxation of 0 implies that only the frames with higher similarity value are considered.

Challenges of Event Segmentation in Low Time Resolution Content Given the limited amount of annotated data, event segmentation in LTR videos is very often unsupervised, performed via traditional clustering algorithms on visual cues (*e.g.* color, CNN hashes) [12, 24, 31–33, 78, 79, 133]. In retrieval systems, one can identify event boundaries by its similarity to the given query, constrained to a continuous timeline [24]. However, a query might not always be available or suitable. Under unconstrained settings, the methods mentioned above usually fail at detecting change points between two events with heterogeneous visual content, and ignoring small and isolated visual changes within an event.

While current event segmentation methods for continuous photostreams mostly rely on the cross-analysis of consecutive frames, this is not how humans understand episodic events. Inspired by our event perception cognitive system, this thesis presents CES, a novel event segmentation paradigm in which each frame is understood as part of a global sequence. As such, it is able to detect inconsistencies in the visual feed and accurately predict the presence of event boundaries of heterogeneous events, and discard visual changes due to temporary diversions.

4.2 Context-based Event Segmentation

As observed by Zacks et al., "perceptual systems continuously make predictions about what will happen next. When transient errors in predictions arise, an event boundary is perceived [141]." As intelligent beings, we learn to identify recurring patterns from our observations. Our sensory experiences allow us to build event models, *i.e.* perceptual representations of "what is happening now", which enable us to anticipate the future and, as a result, detect changes in everyday events. More notably, event models are robust against transient variations in the sensory input and hence ignore disruptions such as occlusions and short distractions. Event segmentation is thus dependent on the timescale. In order to detect coarse-grained events, the transient error in the prediction will have to be greater than that for fine-grained event segmentation.

Given a continuous stream of photos, we would identify the start of a new event if the new frame differs from our expectation of what should follow the preceding sequence. We would also check whether that frame is consistent with the subsequent scene. If the new scene spans a very short time and returns to the previous, we would ignore it as an extra event, but rather wrap it within the current event (*e.g.* going for a bottle of water while watching TV). In order to control the granularity of the event segmentation, we would have to frequently look forward and backward to verify whether it was a new event, or just a brief diversion or local outlier.

4.2.1 Overview of Contextual Event Segmentation

The proposed model is analogous to such intuitive framework of perceptual reasoning. It learns the event model representation from more than 1 million lifelog images captured during around $1,300$ days, using an encoder-decoder architecture. At each timestep t, two event representations are predicted: one given the previous sequence of images, *i.e.* the past, and another given the future sensory input, *i.e.* the ensuing frames. If the two predicted visual contexts, *i.e.* the event representations, differ greatly, CES will infer that the two sequences (past and future) correspond to different events, and will consider $frame_t$ as a candidate event boundary.

CES consists of two modules (*c.f.* Algorithm 4.1): First, the Visual Context Predictor, that predicts the event representation of the upcoming frame, either in the past or in the future depending on the sequence order. Second, the event boundary detector, that compares the event representation at each time-step given the frame sequence from the past, with the event representation given the sequence in the future.

4.2.2 Visual Context Predictor

Inspired by [8, 115, 140], we propose predicting the event representation from a sequence of frames with a Long Short-Term Memory (LSTM) network. LSTM networks are a type of Recurrent Neural Network (RNN) that learn long-time dependencies through four hidden layers, *i.e.* the gates. Thus, LSTMs can aggregate the information they receive by learning to forget. Their mathematical formulation can be expressed as

$$\begin{aligned} \begin{pmatrix} \underline{\mathrm{i}} \\ \underline{\mathrm{f}} \\ \underline{\mathrm{o}} \\ \underline{\mathrm{g}} \end{pmatrix} &= \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \left(\begin{bmatrix} \mathbf{x_t} \\ \mathbf{h_{t-1}} \end{bmatrix} \mathbf{W} \right) \\ \mathbf{c_t} &= \underline{f} \circ \mathbf{c_{t-1}} + \underline{i} \circ \underline{g} \\ \mathbf{h_t} &= \underline{o} \circ tanh(\mathbf{c_t}), \end{aligned} \tag{4.1}$$

where $\mathbf{x}$ is the input sequence; $\underline{i}$, $\underline{f}$, $\underline{o}$, and $\underline{g}$ correspond to the four gates of the unit (*input*, *forget*, *output*, and *input modulation*); $\circ$ is the element-wise product; $\mathbf{W}$ are the network weights [40]; and $\mathbf{c_t}$ and $\mathbf{h_t}$ are the cell state and the hidden state, respectively, at time-step t.

Algorithm 4.1: Overview of Contextual Event Segmentation

▷ *Get past and future event representations from the Visual Context Predictor:*
$\mathbf{rf}(t-1) \leftarrow$ predicted from $[\mathbf{x}_k]_{\forall\ 0 \leq k < t}$
$\mathbf{rp}(t+1) \leftarrow$ predicted from $[\mathbf{x}_k]_{\forall\ \mathbf{len}[\mathbf{x}] \geq k > t}$

▷ *Detect boundary candidates:*
$pred(t) = cos\_dist(\mathbf{rf}(t-1), \mathbf{rp}(t+1))$
$b = \{t \,|\, (\frac{\delta pred}{\delta t} = 0)\}$
▷ *Remove noisy candidates:*
$b = \{b_k \,|\, pred(b_k) \leq average(pred(b))\}$

The sequential and relational nature of lifelogging photo-streams allows us to train the weights of an LSTM-based aggregation network without ground truth annotations. To obtain the weights of our Visual Context Predictor, we train an encoder-decoder architecture that, given a sequence of visual feature vectors, learns to predict the subsequent sequence, as shown in Fig. 4.2. Since LTR video frames are visually highly different from adjacent ones, the model will learn the general context of the event at the same time as the estimation of the visual feature of the upcoming frame.

The auto-encoder is defined as

$$\begin{aligned} \mathbf{r_t} &= \mathbf{h_{t,encoder}}(\mathbf{x_t}) \\ \mathbf{\hat{x}_{t+1}} &= \mathbf{h_{t,decoder}}(\mathbf{r_t}), \end{aligned} \tag{4.2}$$

where $\mathbf{x_t}$ is the deep learned visual feature of frame t, $\mathbf{r_t}$ is the predicted event representation at time t, and $\mathbf{h_{t,encoder}}$ and $\mathbf{h_{t,encoder}}$ correspond to the models trained to encode and decode the visual feature, respectively. The objective function of the learning process is to minimize the mean squared error of the prediction, *i.e.* $mse(\mathbf{x_t}, \mathbf{\hat{x}_t})$.

VCP shares architecture and weights with the encoding model presented above, and is able to encode the event representation of lifelogging image sequences both feed forward and backward, *i.e.* in reverse time order. The chosen architecture for VCP (*i.e.* the encoder) is a single LSTM layer of 1024 neurons. The hidden state is then passed to the decoder, which has a corresponding LSTM layer.

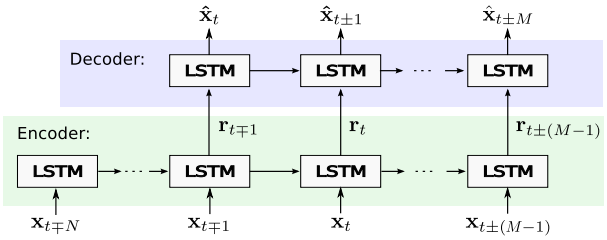


Figure 4.2 Training of the Visual Context Predictor. Given a sequence of features, the model learns to predict the visual feature of the following frame, either in the future or, if the sequence is in reverse order, in the past. The output of the encoder, $\mathbf{r}_t$, corresponds to the event representation at time-step t.

4.2.3 Boundary Detector

Given a frame $\mathbf{x_t}$, two different context predictions can be obtained from VCP. The first, the future context $\mathbf{rf_t}$ including the sequence of frames from the past ($\mathbf{x_{k|0\leq k\leq t}}$). The second, the past context $\mathbf{rp_t}$ including the frames in the future ($\mathbf{x_{k|T\geq k\geq t}}$), where T is the total length of the lifelog. Thus, at each time-step t, the future context given the past will be $\mathbf{rf_{t-1}}$, and the past context given the future $\mathbf{rp_{t+1}}$. Note that the frame $\mathbf{x_t}$ is not seen when predicting the future and past context at time t to avoid overlapping inputs in the prediction.

An event boundary will delimit sequences with very different event representation. Similar to [31], the boundary prediction function is defined

$$pred(t) = d(\mathbf{rf_{t-1}}, \mathbf{rp_{t+1}}), \tag{4.3}$$

where $d(\cdot,\cdot)$ is the cosinus distance.

Event representations will change gradually within the vecinity of a boundary. Therefore, the local maxima become boundary candidates. Since these will also be found for very slight changes in the event representation, only the candidates whose prediction value is over a certain threshold are kept . To maintain an unsupervised pipeline, such threshold is defined independently for each lifelog as the average of all candidates' values. A lower value can be used for coarse-grained segmentation, or the k-th candidate when only k events are wanted.

Variation for High Time Resolution Video Sequences In the case of HTR videos, transitions between events may last several frames, as the sampling is of higher frequency. In those cases, all frames that have an event representation imbalance greater than the threshold are considered to be part of the boundary between events.

4.3 Experiments

We compare the performance of CES on two Low Time Resolution datasets (*EDUB-Seg* and *EDUB-SegDesc*) and two egocentric High Time Resolution video datasets (*HujiEgoSet* and *Disney*). CES is compared against the state of the art. Further tested variations and ablation studies of the importance of VCP and the threshold are included in Annex A.1.

4.3.1 Implementation Details

Data setup The output of the pre-pooling layer of InceptionV3 [117] is used to describe the frames in the lifelog. To find the local maximums in the prediction signal of CES, as well as smoothing the K-Means clustering, a window of size 5 is chosen, so that it is consistent with the ground truth tolerance.

We use the available lifelogging video data from R3, CLEF and NTCIR (*c.f.* Chapter 3) to train the VCP model for LTR data:

- *Training of the VCP model:* 75% of R3 is used as training set for the Visual Context Predictor model. To ensure that the model is not biased toward this dataset, a 20% of both CLEF [20] and NTCIR [45] is also included in the training set. This joined set adds up to $1,207,483$ images. A separate 5% of R3 is used to validate the different configurations and select the best hyperparameters.
- *Testing set for the VCP model:* the remaining 20% of R3, and 80% of CLEF and NTCIR is kept as test to confirm that VCP is not overfitted toward R3 (*c.f.* Annex A.2).

Training methodology We explore several architectures and training parameters for the Visual Context Predictor model. Regarding the architecture, we can modify the number of neurons in the encoding LSTM layer, the number of frames seen before starting the future prediction (N), the amount of frames the decoder needs to predict (M), and whether the prediction will be conditional or not, *i.e.* whether the model gets further inputs past frame N. We investigate architectures between 256 and 1024 neurons, values of $N = M$ between 10 and 100, and the same range of M for $N = 1$. VCP is implemented using Keras with TensorFlow backend.

Concerning the training parameters, the loss is defined as the mean squared error of the prediction $\hat{\mathbf{x}}_\mathbf{t}$, and RMSProp without decay is used as optimizer. The learning rate is randomly

set in the range $[.0001, .001]$, and is reduced by half after every 4 epochs without significant improvement in the validation loss. Different batch sizes are used, between 250 and 1000 sequences at a time.

The best configuration is determined through a gridsearch on all the different parameters. We find that the best prediction performance (smaller validation loss) is achieved with 1024 neurons on a conditional architecture. The number of frames seen before starting the future prediction is set to $N = 10$, equal to the number of frames to predict ($M = 10$). We observe that training with longer sequences does not improve significantly the model performance (*c.f.* Table A.2), while making the training slower. At test time, one single frame ($N = 1$) is given to start the prediction of the whole day ($M = length(lifelog) - 1$).

Wider Boundary for HTR Videos The boundary window for the frames with contextual imbalance above the threshold is limited to a boundary period of one minute. This modification of the boundary detector is named *win* for the results reported in Tables 4.1 and 4.2.

4.3.2 Experimental Results

Evaluation metrics Following the literature, we report the averaged F-measure, precision, and recall for the tested models. For our evaluation, a detected boundary is considered a true positive if there is an element in the ground truth within a distance of tolerance, and the ground truth element is not already matched to any other detected boundary. Analogously, all elements in the ground truth for which no detected boundary is found within the tolerance are considered false negatives. This tolerance is set to 5 frames.

Baseline comparison We compare the performance of CES to that of the following baselines:

KTS. Kernel Temporal Segmentation with our deep feature as described in [103], and also with the unormalized feature (*unorm*).

GIST. Distance between GIST descriptors within a temporal window, as described in [7].

AC-Color. Agglomerative Clustering on the color feature of the frames, as done in [75] (only for LTR datasets).

SR-CNN. Semantic Regularized Clustering as described in [29], using visual features (only for LTR datasets).

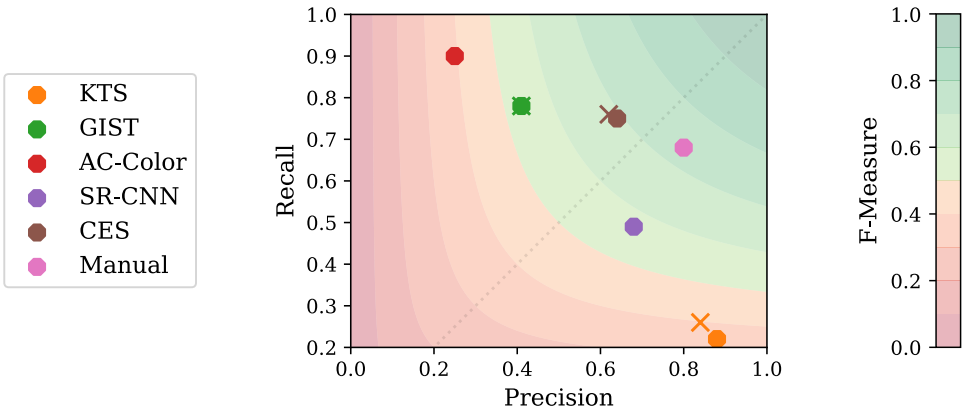


Figure 4.3 Precision-Recall curve for the tested unsupervised event segmentation algorithms on LTR datasets. The corresponding F-measure score is shown in the color space. Results on EDUB-Seg are represented with an octagon, whereas those from EDUB-SegDesc is depicted with an x.

Bias in the Ground Truth Since segmenting lifelogs into events can be a very subjective task, the curators of EDUB-Seg provide in [29] an extensive analysis on the uniformity among the ground truth annotated by different subjects. They conclude that visual lifelog event segmentation can be objectively evaluated, since different people (which are not the camera wearer) tend to segment the lifelogs consistently. For the purpose of our evaluation, we select the ground truth from the first annotator.

Performance of CES relative to manual annotations Since there is not just one correct way of segmenting video content into events, we have to compare the performance of CES relative to that of the average person. For the lifelogs that only included one annotation, we asked independent subjects to annotate the events, so that we would have at least two sets of annotations for each lifelog.

For each lifelog, we average the performance of all available annotations, as evaluated on the selected ground truth. Table 4.1 reports the performance of the manual annotations as an upper reference. We observe that subjects are only 5 points better than CES in terms of F-measure. Even though the precision of the manual annotations is very high, they also have worse recall than CES. This is due to some of the subjects selecting very general events, *e.g.* wrapping all working afternoon into the same event, disregarding the different meetings. Such annotation criteria yield many false negatives and therefore decreases the recall score. Analogously, in some other cases, subjects selected more details than the ground truth, and therefore their rate of false positives is not zero.

Table 4.1 Comparison to the state of the art. Averaged results (F-measure, Precision and Recall) for each Lifelogging dataset.

Sampling:	LTR: 30 sec.						
Dataset:	EDUB-Seg			EDUB-SegDesc			
method	F1	P	R	F1	P	R	notes
KTS (norm) [103]	0.34	0.88	0.22	0.37	0.84	0.26	normalized descriptor
KTS (unorm)	0.59	0.56	0.67	0.56	0.52	0.68	raw descriptor
GIST dist [7]	0.52	0.41	0.78	0.50	0.41	0.78	GIST dist. over window
AC-Color [75]	0.38	0.25	0.90		-		Agglomerative Clustering
SR-CNN [10]	0.53	0.68	0.49		-		
CES30	**0.67**	0.64	0.75	**0.66**	0.62	0.76	VCP trained with R3
CES30-win	0.63	0.53	0.82	0.62	0.52	0.84	Wide boundary
Manual segmentation	**0.72**	**0.80**	0.68		-		

CES segments, on average, into more events than the annotators. As a result, it is able to detect 10% more true boundaries than the average person (75% *vs* 68%) but will also find a relative 80% more incorrect events (36% *vs* 20%). Such a large increase is to be expected, as the selected ground truth is very exhaustive, and the annotators rarely identify boundaries not present in the ground truth. Overall, we can conclude that CES is a highly precise event segmentation algorithm. Given our ground truth, CES' F-measure is of 93% relative to the manual performance.

Results in Low Time Resolution (Lifelogs) Table 4.1 presents the results of CES and the baselines in EDUB-Seg and EDUB-SegDesc. The position of each method in the Precision-Recall curve is shown in Fig. 4.3. While most methods fall within the mid-range performance in terms of F-measure, CES stands out of the baselines, improving their performance by over 15%, and positioning itself on the upper range of the absolute spectrum. The performance of CES is even competitive with that of the manual annotations. We show in Fig. 4.4 the performance of CES applied to one of the tested lifelogs. We can observe that most elements in the ground truth fall on the spikes of the prediction signal, or very close to them. This confirms the suitability of using the predicted contexts as a boundary cue.

While the baselines fail at detecting boundaries between heterogeneous events, CES is capable of extracting the underlying context of each event and discern their disparity (*e.g.* shopping at the supermarket after riding a bike on the street). Moreover, in cases in which the camera wearer orientation changes within a static event (*e.g.* looking back from your food to your

Table 4.2 Comparison to the state of the art. Averaged results (F-measure, Precision and Recall) for each HTR dataset.

Sampling:	HTR: 2 sec.						HTR: 5 sec.					
Dataset:	Huji			FP Social Int			Huji			FP Social Int		
method	F1	P	R	F1	P	R	F1	P	R	F1	P	R
KTS [103]	0.31	0.45	0.27	0.08	0.07	0.11	0.34	0.88	0.22	0.09	0.20	0.06
GIST dist [7]	**0.32**	0.72	0.24	0.14	0.26	0.10	0.31	0.71	0.23	0.13	0.24	0.09
CES30	0.28	0.27	0.35	0.12	0.09	0.18	0.29	0.36	0.28	0.11	0.12	0.10
CES30-win	0.31	0.29	0.41	0.18	0.13	0.30	**0.35**	0.42	0.34	0.19	0.19	0.20
CES10-win	0.30	0.29	0.40	**0.18**	0.13	0.31	0.32	0.42	0.31	**0.23**	0.22	0.25
CES{2, 5}-win	0.28	0.23	0.45	0.15	0.09	0.33	0.34	0.39	0.35	0.20	0.18	0.24

colleagues), traditional segmentation methods detect such view change as an event boundary, whereas CES is able to detect the presence of a common visual path. However, if the view change spans longer than CES memory, CES will not be able to contextualize it within the event. An example for such a situation can be seen in Figs. 4.5b and 4.5c. We also note that the ability of CES to detect the general context of the visual sequence and track common cues sometimes misleads the prediction. When the ground truth of a boundary falls within the same physical space, or similar contexts, CES does not perceive their differences and thus does not detect the boundary. Arguably, such boundaries are also difficult to detect by external viewers. This may also occur when short transitions between events are considered events on their own.

Results in High Temporal Resolution (Egocentric Videos) CES has been tested on two HTR datasets, Huji Ego Set and the First Person Social Interactions Dataset (*c.f.* Chapter 3.1). The results are reported in Table 4.2. Since these datasets have been annotated for activity segmentation the event granularity is higher. For this reason, the VCP has been re-trained using videos the CSumm dataset, downsampled at different time rates, *i.e.* $T = 10$ seconds, $T = 5$ and $T = 2$, namely CES10, CES5 and CES2, respectively. Additionally, the original VCP model trained on R3 is used in CES30. At test time, the videos have been downsampled at one frame every two seconds ($T = 2$) or five seconds ($T = 5$).

We can observe that CES outperforms the baselines for long videos (*i.e.* FP Social Int), while being competitive for shorter videos (*i.e.* Huji Ego). In the case of shorter videos, with shorter events, the VCP model trained with R3 (CES30-win) outperforms training at higher time rates, while for longer events (as those in FP Social Int) training the VCP model with one frame

every 10 seconds is preferred. Higher frame rates in the training data result in worse contextual predictions. This is due to the VCP model learning a trivial representation, as the closer in time consecutive frames are, the more the prediction for $t+1$ will resemble the frame at time t. At test time, we also observe that the overall segmentation performance is better for the higher downsampling ratio of five seconds than for that of two seconds.

4.4 Summary

This Chapter introduced Contextual Event Segmentation, a novel unsupervised event segmentation method that uses the sequential nature of a photo-stream to infer the presence of event boundaries. At the core of CES is the Visual Context Predictor, a model that predicts the event representation of a sequence of frames. At each timestep t, the event representation given the previous sequence is compared to the representation given the future to determine whether there is a boundary at time t.

The Visual Context Predictor is trained with over one million images from R3. With such extensive dataset, it is able to model human activities given sequences of visual features. In a series of experiments, it is demonstrated that the event representation is a strong indicator of event changes. We conjecture that the event representation can also be useful for storytelling tasks and tracking of daily activities.

Leveraging on the event representation of the sequences allows CES to detect boundaries between heterogeneous events and ignore local occlusions and brief diversions. For Low Time Resolution videos, CES improves the performance of the baselines by over 16% in F-measure. Moreover, the performance of CES is competitive with manual annotations, for which the F-measure is only 5% better than that of CES.

In the case of High Time Resolution, the timescale for interactive activities is smaller than that of episodic events. While most events annotated for the EDUB dataset are longer than 20 minutes, most events in Huji and the FP Social Int dataset are shorter than a minute. As a result, the performance of CES in HTR is worse than in LTR. However, the results still show an improvement with respect to the state of the art.

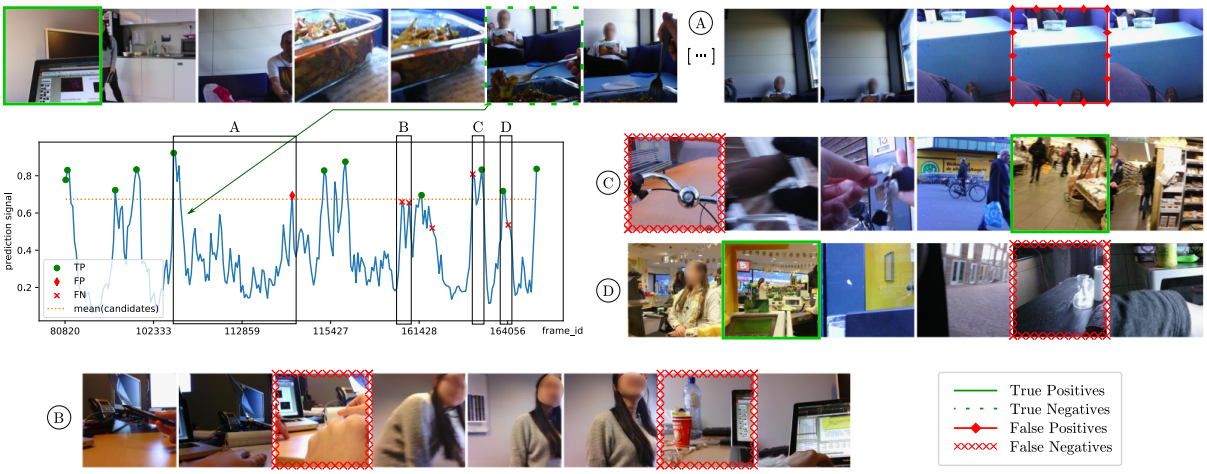


Figure 4.4 Qualitative example for one of the tested lifelogs. Those frames which are false positives or false negatives in the baselines are highlighted. We can observe that, unlike the baselines, CES is able to ignore occasional occlusions as long as the different points of view span fewer frames than CES' memory span (A). It is also capable of detecting boundaries that separate heterogeneous events such as riding a bike on the street and shopping at the supermarket (C, D). Most of the boundaries not detected by CES correspond to events that take place within the same physical space (B) and short transitions (C, D), *e.g.* parking the bike.

(a) True Positives: CES can model public transportation events, as well as street walking.

(b) True Negatives: CES remembers previously seen context, and is able to match future and past.

(c) False Positives will raise if the different sight positions span longer than CES' memory span.

(d) False Negatives: two events taking place in the same location could be understood as a single one.

Figure 4.5 Examples of the capacities of CES.

Chapter 5

Summarization of First Person View Content

In order to relive our recorded personal experiences, it is useful to obtain an automatic summary of them. Summarization also facilitates the browsing of extense data collections. This Chapter presents two methods for egocentric video summarization. For Low Time Resolution content, Contextual Event Segmentation is used to generate diverse summaries from Lifelog data. On the other hand, for High Time Resolution video, a Conditional Random Field is used to model the video and select the most diverse and representative segments.

5.1 Introduction

Lifelog images are typically characterized by excessive redundancy. The goal of lifelog summarization is to automatically select a small set of images to construct a compact story with significant events. Such images should be representative and informative on their own, as well as diverse. In the case of Hight Time Resolution personal videos, a skim is usually a preferred summary output. The relation between the video segments is thus of crucial importance to achieve an optimal storytelling video skim. Moreover, given the spontaneous and shaky nature of the original videos, the visual homogeneity and quality of each segment must be considered to obtain a pleasant summary, as opposed to a shaky and dizzying one.

Many state-of-the-art summarization tools select the segments to include in the summary by optimizing a pre-defined criterion. Such criteria frequently relates to story coherence such as diversity and representativity [19, 145]; interestingness from visual aesthetics, attention, importance to external human judges, *etc.* [17, 47, 61, 80, 84, 103, 136, 143]; or both [48, 83, 92]. Very few of those methods allow for personalization, either via queries, user profiles, or user feedback.

5.2 Personal Lifelog Summarization

Lifelog summaries can be generated following two strategies: (i) extraction of multiple types of events within a time period (e.g. *"What did I do last Saturday?"*), and (ii) extraction of multiple occurrences of the same type of event over a period of time (e.g. *"Gym exercises in the past three months"*). In the scope of this thesis, only summaries for the second strategy have been evaluated. However, the proposed model is equally suitable to select diverse key-frames within a time-constrained visual Lifelog.

5.2.1 Contextual Event Segmentation for Lifelog Summarization

The proposed summarization pipeline produces lifelog summaries from the raw lifelog data that are compact, diverse, and relevant to a given query. Being consistent with the summarization framework presented in Chapter 2, the process follows a series of operations including quality filtering, ranking, event clustering and key-frame selection (Figure 5.1).

Quality Filtering The incoming frame stream is pre-processed to evaluate the quality and informativeness of the images. All frames below a certain quality threshold are then discarded. The quality rate can be obtained by assessing the blurriness and color diversity of the image. Images with very homogeneous colors are deemed to be uninformative, since this usually means there are few different objects in the image. The RGB pixel-values are quantized into a 32-bin space, and the color diversity score is defined as the frequency of the predominant color. On the other hand, the blurriness is computed using Laplacian filtering to detect prominent edges.

Ranking Summaries are generated according to specific requirements. If the requirement is a diverse summary for a time-constrained Lifelog (case *(i)* above), the images can be ranked according to visual quality, memorability, or aesthetics. Alternatively (case *(ii)*), each image is scored according to its relevance to a query-given event [24, 78]. The top ranked Q images are retrieved from the whole lifelog dataset, where Q is larger than the summary length budget.

A linear model is used to obtain the relevance scores. The images are described using features from DCNN models trained on objects, locations, and people, as well as metadata provided by the users. The weights are learned from training data using a Conditional Random Field [78] (*c.f.* 5.3.1) in which each node corresponds to a feature dimension.

Event clustering The set of relevant images belong to various events. To avoid over-sampling images from a few dominant moments, and thus achieve better performance of diversity, event clustering is applied to the set of Q images. Traditionally, k-means clustering on the image descriptor has been used. Instead, we propose clustering into events using Contextual Event Segmentation (*c.f.* Chapter 4). CES is used to locate the time boundaries of events within the original lifelog, so that these boundaries can be applied to the subset of relevant pictures. This method assumes that events are contiguous in time, *i.e.* breakfast at home today will be a different event from breakfast at home tomorrow.

Key-frame selection Event clustering results in a set of image clusters $c_i = \{I_{n_i}\}$, where each cluster corresponds to an event. The images in each event cluster are sorted based on their relevance score. To achieve a balanced diversity of images from different events, the representative images are selected iteratively from the event clusters. The summary is initialized empty, $S = []$, and the following process is repeated iteratively until reaching the desired summary length X: The first available image in each cluster is selected to be part of the final summary $s = \{f_k \mid \forall_i, k = 0\}$, and discarded from the bag of available frames. Then, the selection is sorted according to each frame's relevance score, so that the most relevant are first in the generated summary. The sorted sequence is added at the end of the summary, $S = S^\frown s$. Note that to force that each event will be represented if selecting a summary shorter than X, the sequence to be sorted is the newly drawn s, and never S. If at the end of the drawing process the cadence of S is greater than X, the last elements of S are discarded. The proposed key-frame selection algorithm balances the significance of relevance with diversity under the constraint of compactness for the summarization.

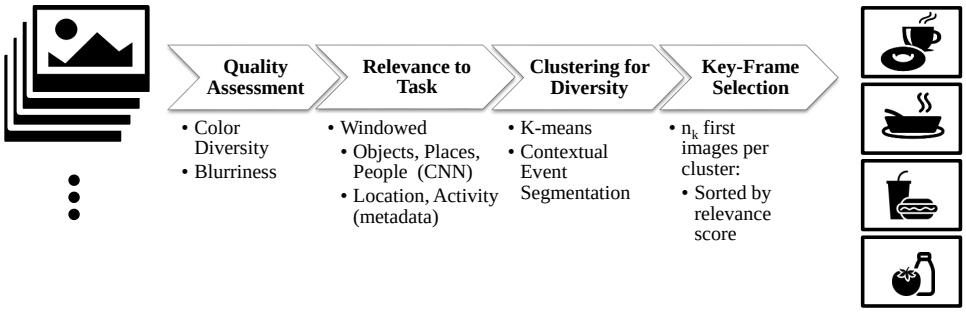


Figure 5.1 The input image stream is first analyzed to remove bad quality images. Then, the frames are assessed in terms of relevance to the task, and clustered into episodic events. The most relevant images in each cluster are selected for a compact, diverse and relevant summary.

5.2.2 Experimental Results

The proposed method is evaluated using the benchmark provided by the ImageCLEF 2017 LifeLog Task [20]. The objective of the lifelog summarization task is to generate a summary on a query event, *e.g.*, *"In a Meeting: Summarize the activities of user 1 (u_1) in a meeting at work"*, by retrieving all the relevant images from the CLEF dataset (*c.f.* Chapter 3).

The proposed pipeline is compared to the best performing run of the competition (denoted as *I2R_S1*), two submissions from the organizer's team [20, 148] (*Org_A* and *Org_SA*), and one from the *UPB* team [30]. All the approaches employ visual concepts learned from ImageNet, and additional metadata. In *Org_A* the query topics are parsed into keywords by using natural language processing (NLP). Alternatively, in *Org_SA*, keywords are selected manually. The *UPB* method uses WordNet vocabulary to automatically filter the topics for keywords. The official performance metric in the benchmarking is F1@10. We additionally report the F1@50 to evaluate the impact of each method in the *extensive run* (*i.e.*, for larger summaries).

While the best method (*I2R_S1*, as reported in [24]) achieves an F-1 performance of 0.497, it involves a high amount of human input with a trial-and-error selection of keywords, and setting the weights of semantic aspects, requiring human intelligence and skills of online tuning. The proposed method, on the other hand, learns the relevance weighting automatically. Its performance is 16% higher than the *Org_SA* score of 0.319 [148] and outperforms the baseline (*Org_A*: 0.19) and the *UPB* method (0.13) [30] by large margins. The same query parsing is evaluated for summaries generated using k-means segmentation, and Contextual Event Segmentation. Comparing the two methodologies, we can observe that using CES to cluster the relevant pictures consistently improves the performance of clustering with k-Means, proving its suitability for lifelog summarization.

Table 5.1 Comparison to the state of the art. Averaged results (F-measure, Precision and Recall).

Summary Size:	$X = 10$			$X = 50$			
method	F1	P	R	F1	P	R	notes
Org_A [148]	0.19	N.A.			N.A.		Automatic (NLP)
Org_SA [148]	0.32	N.A.			N.A.		Keywords
UPB [30]	0.13	N.A.			N.A.		WordNet filter
I2R_KM [24]	**0.50**	0.70	0.43	**0.51**	0.53	0.58	Human intervention
CRF_KM	0.30	0.41	0.28	0.37	0.34	0.49	Relevance learned from
CRF_CES	**0.37**	0.53	0.33	**0.39**	0.34	0.54	WordNet propagation

5.3 Consumer Video Summarization

As opposed to Lifelogs, personal HTR videos involve motion, and a video skim is a preferred output. For the summarization of egocentric and consumer HTR video, this thesis proposes using a probabilistic approach based on active inference in Conditional Random Fields (CRFs) [105]. It can model the video contextual dependencies and infer the most diverse and representative summary. CRFs are sound probabilistic models that have been successfully applied in many computer vision and multimedia problems [72]. The proposed formulation with the CRF allows for user-driven passive personalization (*e.g.* from queries), as well as active customization by modifying the CRF parameters during on-line interaction [25], as shown in Chapter 7.

5.3.1 Conditional Random Fields for Video Summarization

Let $\mathbf{s} = \{s_i\}$ be the set of random variables that represent the summary of the video by indicating whether a segment (or subshot) of the video appears in the summary, or not. Thus, $s_i \in \{0,1\}$, where s_i is equal to 1 when the segment is included in the summary, and 0 otherwise. We denote $P(\mathbf{s}|\theta)$ as the probability density distribution of how likely the summary $\mathbf{s}$ is preferred by the user. This distribution is modeled with a CRF, and θ are the values of its parameters, that depend on the input video and, when available, the user's preferences.

A CRF models the probability density with a Gibbs distribution [69]. Therefore, $P(\mathbf{s}|\theta)$ can be written as the normalized exponential of an energy function, which is denoted as $E_\theta(\mathbf{s})$. The energy function is the sum of a set of potentials, which are functions that take as input a subset of $\{s_i\}$. The summary of the video, which is denoted as $\mathbf{s}^\star_\theta$, is obtained by inferring the Maximum a Posteriori (MAP), *i.e.* $\mathbf{s}^\star_\theta = \arg\max_{\mathbf{s}} P(\mathbf{s}|\theta)$, or equivalently, maximizing the energy function $E_\theta(\mathbf{s})$[1].

We follow most methods in the literature, that select representative and diverse segments with as little motion as possible. To do so, we define the energy function of the CRF as

$$E_\theta(\mathbf{s}) = \lambda \sum_i \underbrace{\phi_u(s_i)}_{\text{unary}} + \sum_{ij} \underbrace{\phi_p(s_i, s_j)}_{\text{pairwise}}, \tag{5.1}$$

[1] Note that we omit the dependency of the potentials on θ for simplicity, and the parameters that we introduce in the following should be considered as part of θ. See Section 5.3.3 for details on the values of the parameters of the potentials.

where the unary potentials enforce the selection of static segments, the pairwise potentials encourage segments with diverse semantic content, and λ is a parameter that weights the unary potentials with respect to the pairwise. There is a unary potential for each segment of the video and one pairwise potential for each pair of similar segments. The length of the summary is controlled during the inference of the MAP summary by adding additional constraints to the energy function that control the length of the summary, as we show below.

Unary Potentials The unary potentials, $\{\phi_u(s_i)\}$, encourage selecting segments that the user will probably like. $\phi_u(s_i)$ is equal to $Q_i\mathbf{I}[s_i = 1] + L\mathbf{I}[s_i = 0]$, in which: $\mathbf{I}[a]$ is an indicator function that is 1 if a is true and 0 otherwise; Q_i is a function representing how well that segment relates to the requirements, if available, or its visual quality otherwise; and L is a constant offset that is set during the MAP inference of the summary in order to adjust the summary length.

Pairwise Potentials The pairwise potentials, $\{\phi_p(s_i, s_j)\}$, are defined between each pair of similar segments, and enforce selecting segments with diverse content.

Let $d(\psi_i, \psi_j)$ be the Euclidean distance between the descriptors of two segments (refer to the specific implementation details in section 5.3.3). The pairwise potential enforces that similar segments should not be included in the summary. To do so, we define a potential that is weighted by the distance between descriptors, *i.e.* $\phi_p(s_i, s_j) = \exp(-d(\psi_i, \psi_j))\phi'_p(s_i, s_j)$, in which $\phi'_p(s_i, s_j)$ enforces that both segments should not be selected at the same time, and the term $\exp(-d(\psi_i, \psi_j))$ reduces the effect of $\phi'_p(s_i, s_j)$ when the segments are dissimilar. In this way, only a representative segment among similar segments is selected.

By default, $\phi'_p(s_i, s_j)$ is defined as

$$\phi'_p(s_i, s_j) = \begin{cases} L\alpha & \text{if } s_i = s_j = 0 \\ -L\beta & \text{if } s_i = s_j = 1 \\ \gamma & \text{if } s_i \neq s_j \end{cases}, \tag{5.2}$$

where γ is the cost of selecting only one segment in the pair, α and β are the cost to discard or select both segments, respectively, and L is a variable parameter that controls the length of the summary. To obtain short, diverse, and representative summaries, α, β and γ are pre-defined possitive scalars. The formulation with the CRF allows for them to become matrixes when the user gives active instructions, as described in Chapter 7.

5.3.2 MAP Inference of the Summary

There are many off-the-shelf algorithms to obtain the MAP summary from the CRF with the energy function introduced in Eq. (5.1). We use the implementation of Belief Propagation (BP) [137] implemented by Boykov and Kolmogorov (2004), using a maximum of five iterations.

The summary is generated using a line search algorithm that optimizes the values of L and λ to yield the desired summary duration and balance between visual quality (unary potentials) and diversity (pairwise potentials). Recall that the parameter L encourages excluding segments from the summary when $L > 1$: $s_i = s_j = 0$ is further encouraged and $s_i = s_j = 1$ is further penalized (due to the negative sign). Thus, when L is increased, the summary is shorter; otherwise, it is longer. Additionally, the parameter λ is increased when the segments selected do not meet the minimum quality criteria or to better meet the initial requirements, and decreased to facilitate diverse content.

5.3.3 Implementation Details

The specifics of the implementation and the values of the different constants are presented below. These values were manually set during development.

Video Segmentation The subshot boundaries used for the summarization are estimated with the motion status and changes on the environment. In CSumm, these are obtained through the gyroscope from Google Glass to infer motion, and the illumination sensor to identify abrupt changes in the lighting condition. Each segment is set to be around 2.5 seconds long, and its boundaries to match a change in illumination or motion pattern. When the sensor data is not available, segments are equally set to be around 2.5 seconds long, with its boundaries matching a change in the image overall illumination, obtained from a quantization of the image mean intensity.

Segment Descriptors The frame descriptors, ψ_i, are based on the output of a neural network for object recognition, extracted for each frame. Specifically, we use the last layer from AlexNet [71], trained in Places dataset [147] and in ImageNet [106]. We concatenate the output of the neural network for the categories of objects (including animals) and places. Finally, we average the value for each item along all the frames in the video segment.

Unary Potentials Recall that Q_i represents the quality of the video segment for the user. Initially and by default, Q_i depends on the motion and blur. Q_i is inversely proportional to the amount of motion in the segment, which is estimated from a blur detector, and from the gyroscope if such sensor data is available. Q_i is normalized to take values in $[0,1]$.

Additional passive preferences can be added by the user beforehand. Such preferences are included in the model as constraints for this potential. The preferences can be obtained from user profiles, or as queries. The framework also allows for the user to select relevant and irrelevant items from a list. Such items are the top ranked *objects* and *places* categories in the video (*i.e.* the categories with higher accumulated activation). Then, Q_i is increased or decreased, respectively, depending on the activation value of such items in segment i. This is done by multiplying Q_i by $1+\sum_{j\in\text{relevant}}\psi_i(j)-\sum_{j\in\text{irrelevant}}\psi_i(j)$, in which $\psi_i(j)$ is the output of the neural network for the category indexed with j.

Pairwise Potentials To enforce representativeness of the segments in the summary, we set $\alpha=5$, $\beta=1$ and $\gamma=1$. We can observe by analyzing Eq. (5.2) that these parameters penalize selecting both segments ($\beta=1$, and the negative sign). Also, these parameters encourage that both segments are discarded ($\alpha=5$), or that only one of them is selected ($\gamma=1$). Note that α is bigger than γ because in most cases the pair of segments in the pairwise potential should be discarded, as only few segments should be selected in the final summary.

To reduce the computational cost of the MAP inference algorithm, we discard the pairwise potentials with smallest influence. Specifically, we discard 30% of the pairwise potentials that encode the largest distances between segments.

Duration of the Summary The duration is variable depending on the length of the original video. It is set to be around 0.1% of the video length, with a minimum of three to four video segments.

5.3.4 Experimental Results

The proposed video summarization method has been compared to the state of the art in two challenging datasets: eight videos from CSumm of a maximum of 30 minutes, and the videos of UTEgo split to be of at most 3 hours long, generating seven videos.

Comparison to the state of the art The methods used for the evaluation are the following:

Uniform. Summary from uniformly sampled segments.

VMMR. Video Maximal Marginal Relevance, a summarization method which rewards diversity [77].

Lee *et al.* [76]. Key-frames extracted with the method presented by Lee et al. (2012). These summaries are only available in UTEgo.

Manual. In CSumm, where results by Lee et al. (2012) are not available, such baseline is replaced with a manual annotation of the best segments, with a length constraint of three to five key-frames.

User survey The summaries generated with each summarization method have been compared by 15 independent judges that responded an on-line survey. Given the constraints of the on-line surveying tool, the summaries were given in the form of key-frames. For each video, the subjects were asked to assessed the quality of the four summaries by ranking them from *worst* to *best* on a four-level scale. They were requested to rate at least one as *worst* and one as *best*, and were allowed to rate two summaries as equally good or bad, *i.e.* give the same score to both. A summary had to be rated better than another if it was more informative and complete, or if it was as informative and complete, but shorter in number of key-frames. The subjects had never seem the full videos before, and could only assess the quality of one summary relative to another.

Results Table 5.2 reports the number of videos for which most queried users consider the summary generated with one method better than another. On the tested datasets, the proposed method, namely *CRF*, generates summaries more informative than VMMR, and equally informative to uniform sampling for CSumm. As expected, manual annotations are deemed more informative than the automatic summaries. In UTEgo, the method by Lee et al. is also the best ranked one. We can observe that the performance of the proposed method is slightly better in CSumm than for the longer videos from UTEgo. This might be due to two reasons: First, a longer graph takes longer to converge, and might not reach an optimal output. Second, the videos of UTEgo are of low visual resolution, and as such the descriptor might be less representative and discriminative.

Note however that for three videos in CSumm most users reported uniform sampling being more informative than the manual summarization. Most notably, two other videos got

Table 5.2 Number of videos for which the method on the left is ranked better than the one on top by most surveyed users.

	CSumm				UTEgo			
	Unif.	Manual	VMMR	CRF	Unif.	Lee *et al.*	VMMR	CRF
Uniform	-	3	3	**4**	-	2	2	4
Manual/Lee *et al.*	3	-	6	5	5	-	5	5
VMMR	3	1	-	3	5	2	-	3
CRF	**4**	2	**5**	-	3	1	**4**	-

the same informativeness score for uniform sampling and manual editing. We can therefore conclude that egocentric video summarization is a highly subjective task, making uniform sampling competitive with manual annotations. Adapting the summary to the user's preferences is required to achieve higher user satisfaction.

5.4 Summary

This Chapter presented two methods to summarize FPV content, one tailored for Low Time Resolution videos, the other for High Time Resolution videos.

For LTR content, a method based on Contextual Event Segmentation is proposed. Given a query, *e.g.* "What did I eat during the past month", the proposed framework ranks all images in the lifelog according to their relevance to the query and their visual quality. The lifelog is segmented into episodic events using CES, so that the top-ranked images can be clustered together according to their belonging to each event. The most relevant image from each cluster is then selected as a candidate key-frame. The top ranked images withing the length budget will then comprise the final summary. In a series of experiments, CES is demonstrated to be more suitable to obtain diverse lifelog summaries than k-means.

For HTR content, a method modeling the video with a Conditional Random Field is proposed. The most diverse summary is found maximizing an energy function that weighs the visual quality of each video segment and the similarity between pairs of segments. While the results obtained are not better than the state of the art, we conjecture that the proposed method is suitable for FPV video summarization if appropriately modified. In particular, the CRF formulation allows for parameter-tunning according to user feedback. The suitability of this methodology for active user customization is demonstrated in Chapter 7.

Chapter 6

Passive Customization without User Intervention

The usability of video summarization can be significantly improved if customizing the summary. This Chapter presents Personalized Highlight Detection, a highlight detector that is personalized via its inputs. The experimental results show that using the user history as a personalization cue substantially improves the prediction accuracy. PHD improves over generic highlight detection even when only one user-specific example is available.

6.1 Introduction

A personalized video summary should only contain segments that will be of interest for a particular user. This can be achieved via user queries, but in most automatic video summarization scenarios it is desirable to avoid user intervention. Previous models for personalized interestingness prediction learn user-specific models directly from annotation of a particular user [58, 113]. While this approach works in principle, it has two important practical issues: First, its computational cost. Having a model per user is often infeasible in practice, due to the cost of training and storing models. Second, limited user data. Typically, only a small number of examples per user are available. This limits the class of possible methods to simple models that can be trained from a handful of examples. Ren et al. [104] improve upon these methods by proposing a generic regression model which is personalized with a second, simpler, model that predicts the residual for a specific user. While this method can handle users with no history, it still requires (re-)training a model for each new user.

In contrast to that, Personalized Highlight Detection is a global model that is personalized via its inputs, by using information of the GIFs that a user previously created. As the user information is an input and is not embedded into the model parameters, PHD does not need retraining as new user information arrives, and is able to perform well even for users that have no previous history (*cold start* problem). Moreover, the model is trained for all users jointly, allowing for more complex architectures. Furthermore, it does not need user intervention before or during testing: PHD uses the user's history as the signal for personalization, as a user's GIF history allows for a fine-grained understanding of their interests.

6.2 Personalized Highlight Detection

PHD builds on the success of deep ranking models for highlight detection [49, 136], but makes the crucial enhancement of explicitly taking a user's interests into account. In particular, the proposed model predicts the score of a segment as a function of both the segment itself and the user's previously selected highlights. As such, the model learns to use the *user history* to make accurate personalized predictions.

Let V be the video from which a user U wants to generate a GIF, and s the segments that form it. A ranking approach [63] is used to train the model, which learns to score positive video segments higher than negative segments from the same video. A segment is a positive if it was part of the user's GIF and a negative otherwise, as in [49]. In contrast to previous works [49, 116, 136], the predictions are not made based on the segment alone, but also take a user's previously chosen highlights, *i.e.* their history, into account. Thus, the learning objective is

$$h(s^+, \mathscr{G}) > h(s^-, \mathscr{G}), \quad \forall \left(s^+, s^-\right) \in V, \tag{6.1}$$

where s^+, s^- are positive and negative segments coming from the same video V, $\mathscr{G}$ denotes all the GIFs that user U previously generated, *i.e.* the *user's history*, and $h(s, \mathscr{G})$ is the score assigned to segment s. This formulation allows the model to personalize its predictions by conditioning on the user's previously selected highlights.

While there are several ways to do personalization, making the user history an input to the model has the advantage that a single model is sufficient and that the model can use all annotations from all users in training. A single model can predict personalized highlights for all users and new user information can trivially be included. Previous methods instead embedded the personal preferences into the model weights [104, 113], which requires training one model per user and retraining to accommodate the new information.

Two models are proposed for $h(\cdot, \cdot)$, which are combined with late fusion. One takes the segment representation and aggregated history as input (**PHD-CA**), while the second uses the distances between the segments and the history (**SVM-D**). Next, these two architectures are presented in more detail. In all models the segments s and the history elements $g_i \in \mathscr{G}$ are described using C3D [123] (conv5 layyu7er). These vector representations are denoted $\mathbf{s}$ and $\mathbf{g_i}$, respectively.

Model with aggregated history We propose to use a feed-forward neural network (FNN) similar to [49, 136], but with the history as an additional input. More specifically, the history representations $\mathbf{g_i}$ across examples are averaged to obtain $\mathbf{p}$. The segment representation $\mathbf{s}$ and the aggregated history $\mathbf{p}$ are then concatenated and used as input to the model:

$$h_{FNN}(s,\mathscr{G}) = FNN\left(\begin{bmatrix}\mathbf{s}\\ \mathbf{p}\end{bmatrix}\right). \tag{6.2}$$

The model is a small neural network with 2 hidden layers with 512 and 64 neurons [1].

Distance-based model The assumption behind using a model of the form $h(s,\mathscr{G})$ is that the score of a segment depends on the similarity of the segment to a user's history. That assumption is explicitly encoded into the model. Specifically, we create a feature vector that contains the cosine distances to the k most similar history elements g_i. This feature vector is denoted $\mathbf{d}$. Using this representation a linear ranking model (ranking SVM [73]) is trained to predict the score of a segment, *i.e.*

$$h_{SVM}(s,\mathscr{G}) = \mathbf{w^T d} + b, \tag{6.3}$$

where $\mathbf{w}$, b are the learned weights and bias. While the distance features could directly be provided to the model introduced in Section 6.2, we find that training two separate models and combining them with late fusion leads to improved performance (*c.f.* Table A.3). This is in line with previous approaches that found this method to be superior over fusing different modalities in a single neural network [15, 111].

Model fusion The two models introduced above (eq. 6.2 and 6.3) are combined with late fusion. As the models differ in the range of their predictions and their performance, we apply a weight for the model ensemble. To be concrete, the final prediction is computed as

$$h(s,\mathscr{G}) = h_{FNN}(s,\mathscr{G}) + \omega * h_{SVM}(s,\mathscr{G}), \tag{6.4}$$

where ω is learned with a ranking SVM on the videos of a held out validation set.

[1] While different aggregation methods are possible, averaging the history works well in practice. Alternative ways to aggregate were also tested, such as learning the aggregation with a sequence model (LSTM), but these lead to inferior performance (see Annex A.2).

6.3 Experiments

The proposed method, called **PHD-CA + SVM-D**, is evaluated on a subset of the dataset PHD^2 (*c.f.* Chapter 3). It is compared against the state of the art for non-personalized highlight detection, as well as several personalization baselines. The ground truth was acquired before the experiments, and no user feedback was used to refine the experimental results. As such, one cannot know if any of the candidate GIFs would be as good as the GIF a user manually generated, or if said user would have selected it if given the suggestion. Thus, the results presented can only be considered as a lower bound, *i.e.* a worst-case scenario. For more details on the contribution of the different inputs and architectural choices, as well as variations of the method, see Annex A.2.

6.3.1 Implementation Details

Data Setup The models are trained on a subset of PHD^2. 13,822 users, of which 11,972 are used for training, 1,000 for validation, and 850 for testing. At both training and test time, the goal of the model is to predict what part of video V a user chooses, given his history $\mathscr{G}$. As such, V corresponds to the last video from each user, and all other videos are used to build each user's history $\mathscr{G}$. The validation set is used to find the best hyper-parameters for the highlight models and also to find the right weight ω for Eq. 6.4.

To train the model, five positive-negative pairs (s^+, s^-) are sampled from each user's video V, where a positive example s^+ is a shot that was part of the user's GIFs for that video (see Figure 6.1), and a negative example s^- is a shot that was not included in any GIF. To split the user selected segments into shots, we use the shot detection of [46] and deterministically split shots longer than 15 seconds into 5 second chunks. For the user history $\mathscr{G}$, we use a maximum of 20 shots, which are selected at random ensuring that there is at least one shot from each of the last $k = 20$ videos in the user's history. Since a user may generate several overlapping GIFs before being satisfied with the result, $\mathscr{G}$ (and analogously the ground truth for V) does not correspond to each of the user-generated GIFs independently, but rather their union.

At test time the videos are segmented into fixed segments of 5 seconds to be able to compare to [49]. Furthermore, [46] may predict short shots and gaps (due to slow scene transitions), which, when used at test time, would lead to noise in the evaluation. The user's full history is used when making predictions. Since the distance-based models require a k-dimensional

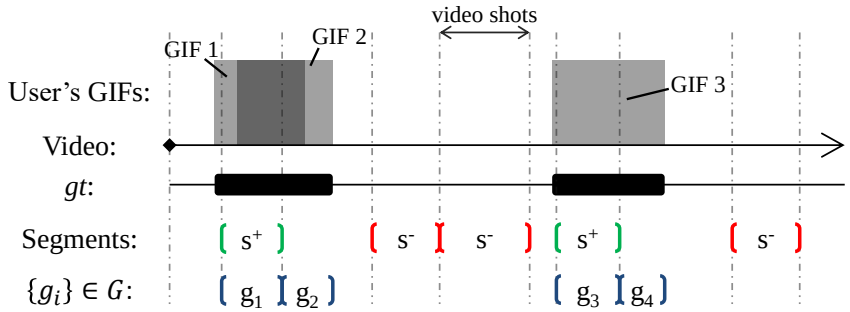


Figure 6.1 Procedure to obtain the pairs of segments (s^+, s^-) in V, the user selection gt for the evaluation, and the user *history* $g \in \mathcal{G}$ from any other *video* $\neq V$.

input, the distance vector is filled with zeros if $|\mathcal{G}| < k$, and the oldest elements, *i.e.* those GIFs generated long ago, are discarded if $|\mathcal{G}| > k$.

Training methodology The network parameters are optimized using grid search over different possible FNN architectures. Different dropout values (random search between .5 and .8 for the input layer, and .1 to .5 for the intermediate ones) and activation functions (*ReLU* and *SELU* [68]) were explored, as well as the use of batch normalization [56] after each layer. Using *RMSProp* as optimizer and a weight decay between 1e−3 and 2, the initial learning rate (randomly set between 1e−2 and 1e−4) is decreased by half every four epochs, for a total of 16 epochs per search iteration. The pairwise loss function used for all models is l_1.

For learning the combination of the user profile and the segment information, we ran hyper-parameter search and varied the number of hidden layers of the FNN from one to three. We tested layers having up to 512 neurons, where each following layer would have the same number or fewer neurons. We find that smaller architectures perform best. The final model **PHD-CA** consists of two hidden layers of 512 and 64 neurons.

6.3.2 Experimental Results

Following [49], PHD is evaluated according to its mean Average Precision (mAP) and normalized Meaningful Summary Duration, which rates how much of the video has to be watched before the majority of the ground truth selection was shown, if the shots in the video had been re-arranged to match the predicted ranking order. In addition, Recall@5 corresponds to the ratio of frames from the user-generated GIFs (the ground truth) that are included in the 5 highest ranked GIFs.

Baseline comparison *PHD-CA + SVM-D* is compared against several strong baselines:

Video2GIF [49]. This work is the state of the art for automatic highlight detection for GIF creation. We evaluate the pre-trained model which is publicly available. As the model is trained on a different dataset we additionally provide results for a slight variation of [49], trained on our dataset, which we refer to as *Video2GIF (ours)*.

Highlight SVM. This model is a ranking SVM [73] trained to correctly rank positive and negative segments as per Eq. (6.1), but only using the segment's descriptor and ignoring the user history.

Maximal similarity. This baseline scores segments according to their maximum similarity with the elements in the user history $\mathscr{G}$. The cosine similarity as used as the distance measure.

Video-MMR. Following the approach presented in [77], $\mathscr{G}$ is used as the query so that the segments that are most similar are scored highly. Specifically, we use the mean cosine similarity to the history elements g_i as an estimate of the relevance of a segment.

Residual Model. Inspired by [104], we include a residual model for ranking. [104] proposes a generic regression model and a second user-specific model that personalizes predictions by fitting the residual error of the generic model. To adapt this idea to the ranking setting, we propose training a user-specific ranking SVM that gets the generic predictions from *Video2GIF (ours)* as an input, in addition to the segment representation $\mathbf{s}$. Thus, a user's model is defined as

$$h_{res}(s,\mathscr{G}) = \mathbf{w}_{\mathscr{G}}{}^{\mathbf{T}} \begin{bmatrix} \mathbf{s} \\ h_{V2G}(\mathbf{s}) \end{bmatrix} + b, \tag{6.5}$$

where $\mathbf{w}_{\mathscr{G}}$ are the weights learned from the history $\mathscr{G}$.

Ranking SVM on the distances. This model corresponds to the model defined in eq. 6.3.

Results Quantitative results are reported in Table 6.1. Qualitative examples are provided in Figure 6.2. When analyzing the results, we find that PHD outperforms [49] as well as all baselines by a significant margin. Adding information about the *user history* to the highlight detection model (**CA + SVM-D**) leads to a relative improvement over generic highlight detection (**Video2GIF (ours)**) of 5.2% (+0.8%) in mAP, 4.3% (-1.8%) in mMSD and 8% (+2.3%) in

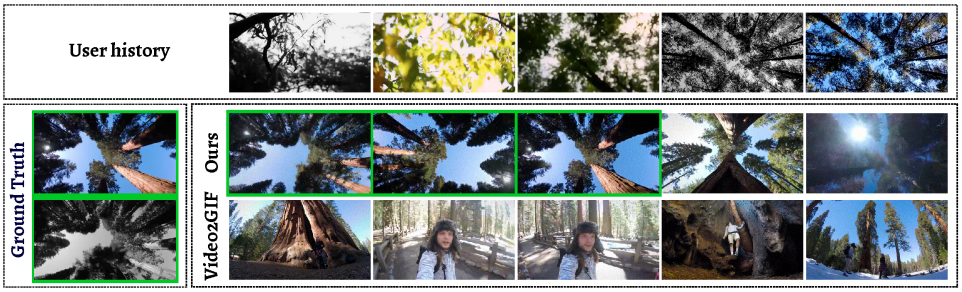
Table 6.1 State-of-the-art comparison (videos segmented into 5-second long shots). For mAP and R@5, the higher the score, the better the method. For MSD, the smaller is better.

	Model	mAP ↑	nMSD ↓	R@5 ↑	Notes
Non-personal	Random	12.97%	50.60%	21.38%	
	Video2GIF [49]	15.69%	42.59%	27.28%	Trained on [49]
	Highlight SVM	14.47%	45.55%	26.13%	
	Video2GIF (ours)	**15.86%**	**42.06%**	**28.42%**	
Personal	Max Similarity	15.49%	44.22%	26.44%	unsupervised
	V-MMR	14.86%	43.72%	**28.22%**	unsupervised
	Residual	14.89%	47.07%	26.05%	
	SVM-D	**15.64%**	**43.49%**	28.01%	
PHD (CA + SVM-D)		**16.68%**	**40.26%**	**30.71%**	

Recall@5. This is a significant improvement in this challenging high-level task and compares favorably to the improvement obtained in previous work [49]. The improvement of PHD over using the user history alone is even larger, thus reinforcing the need to train a personalized highlight detection model that uses the information about all users jointly.

Models using only generic highlight information or only the similarity to previous GIFs perform similar (15.86% for **Video2GIF (ours)** *vs.* 15.64% mAP for **SVM-D**), despite the simplicity of the distance model. Thus, we can conclude that these two kinds of information are both important and that there is a lot of signal contained in a user's history about his future choice of highlights. This concurs with the qualitative analysis in Chapter 3, where we find that that most users in our dataset are consistent in the kind of highlights they select.

Given that the combination of the two kinds of information improves the final results, we conclude that they are complementary to each other and that it is beneficial to use models that consider them both. The residual model also combines generic highlight detection and personalization. However, it estimates model weights per user, which leads to inferior results on the test set, due to the small number of training examples per user. Indeed, the **Residual** baseline is outperformed by the generic highlight detection and the personalization baselines, in particular **SVM-D**. PHD, on the other hand, performs well in this challenging setting and outperforms all baselines by a large margin.



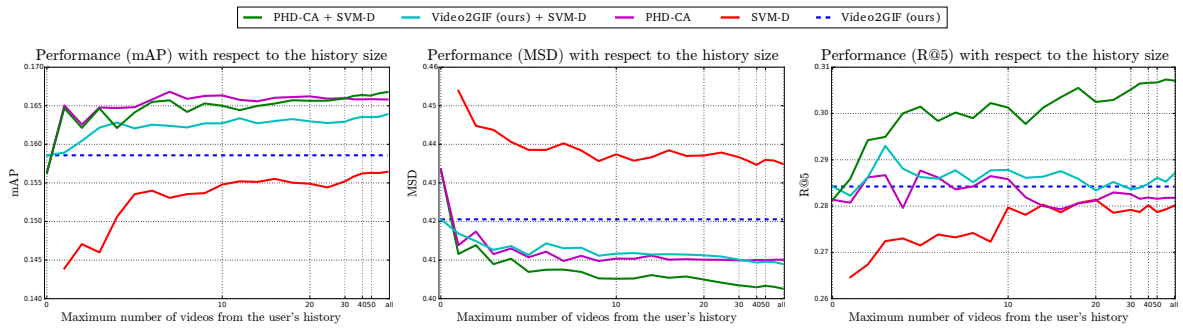
(a) User with interest in forests

(b) User favoring knock-outs

(c) User creating previews for TV shows.

Figure 6.2 Qualitative Examples. We compare our method (**PHD-CA + SVM-D**) to generic highlight detection (**Video2GIF (ours)**). Videos for which personalization improves the Top 5 results are shown in **(a)** and **(b)**. In both cases the users are consistent in what content they create GIFs from. Thus, personalization provides more accurate results (correct results have green borders). In **(c)** we show a failure case, where the user history is misleading the model.

To better understand how the model works, Figure 6.2 shows qualitative results for PHD and the non-personalized state of the art, along with the user history. As can be seen from 6.2a & 6.2b, PHD effectively uses the history to make more accurate predictions. Figure 6.2c shows a failure case, where the history is not indicative of the highlight chosen by the user.



(a) **(b)** **(c)**

Figure 6.3 Performance of different methods as a function of the history size. We observe that our method improves over generic highlight detection with as little as one history element per user. Furthermore, performance has not saturated even when using the full history, thus indicating that our method can effectively use longer histories as well. Interestingly, we find that only models including the distances to the history as a feature improve Recall@5, *i.e.* provide better results at the top of the ranking.

How much does personalization help for different history sizes? We are interested in how well the model performs when very little user-specific information is available. To do so, we restrict the history provided to the model to the last k videos a user created GIFs from, rather than providing the full history[2].

The performance of different tested models is presented as a function of the history length k in Figure 6.3. From this plot, we make several important observations. (i) Adding personalization helps even for small histories. Recall@5 improves by 5.6% (+1.6%) over the generic model for a history size of $k = 4$, for example. Even for $k = 1$, *i.e.* a single history video, our method outperforms generic highlight detection across all metrics. Having a model that performs well given few history elements is important, as the history size follows a long tail distribution (*c.f.* Figure 3.2 in Chapter 3). Indeed, more than 90% of the user profiles were discarded when creating the dataset, as they had a history of fewer than 5 elements. (ii) While **PHD-CA** quickly improves mAP as the history grows, only the model including the distances significantly improves Recall@5. This is consistent with our experiments in Table A.3. Improving the ranking of the highest scoring segments is challenging, as they often have only subtle differences. The similarity to a user's history allows to capture these differences and thus obtain a better ordering of the top elements. (iii) Performance is not yet saturated for the history lengths in the dataset. Thus our model is not only able to make use of small histories, but can also effectively use larger histories to further improve prediction accuracy.

[2]Note that some users may have less than k videos in their history, and only $n < k$ videos can be considered.

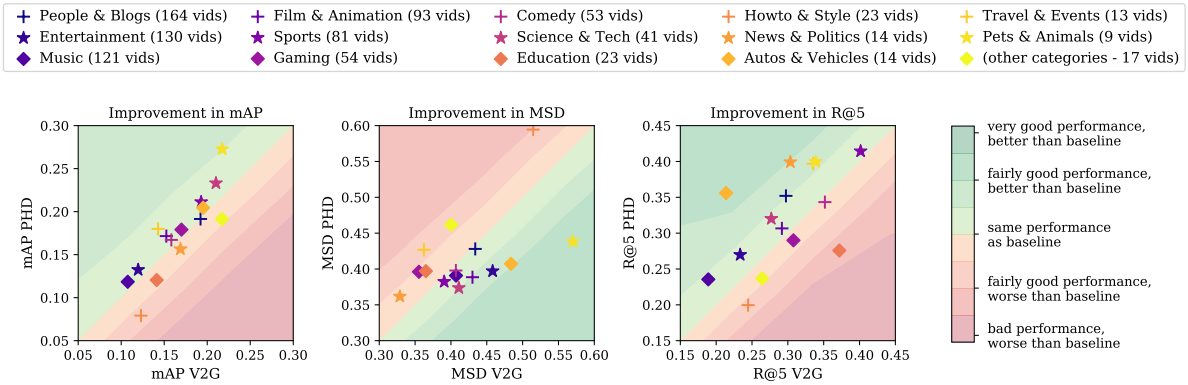


Figure 6.4 Performance of the proposed method (PHD) against the state of the art (V2G) for each YouTube category. The green areas correspond to better performance for PHD than Video2GIFs, with darker green representing a big improvement with respect to the baseline. On the other hand, points located in darker areas correspond to categories in which PHD provides much worse predictions. The most popular categories (purple to dark blue markers) are mostly located in green areas.

Which video categories are more predictable? The test set contains 850 YouTube videos from all existing categories on the full set. Figure 6.4 shows the performance of the proposed model (**PHD-CA + SVM-D**) for each category with respect to the baseline (**Video2GIF (ours)**). The amount of videos for each category is reported in the figure's legend, and is consistent with that observed for the whole dataset (*c.f.* Figure 3.3 in Chapter 3).

We can observe that the most popular categories (those with a color marker in the purple family) mostly fall within the green range, *i.e.* better performance of PHD. While one could think that this is due to the training set also having more samples from these categories, both **PHD-CA + SVM-D** and **Video2GIF (ours)** have been trained with the same dataset, and thus both should be equally affected by this imbalance. In particular, predictions from the Pets & Animals and Autos & Vehicles categories seem to benefit greatly from personalization. Videos from Music and Entertainment also get slightly better predictions when the user history is available. On the other hand, the predictions for videos from Education, Gaming and How-to & Style are in general misled by the user history. From Figure 6.4 we can also observe that, for this test set, the most predictable categories (either with or without history) turn out to be Sports, Science & Tech, and Pets & Animals. How-to & Style is the least predictable category.

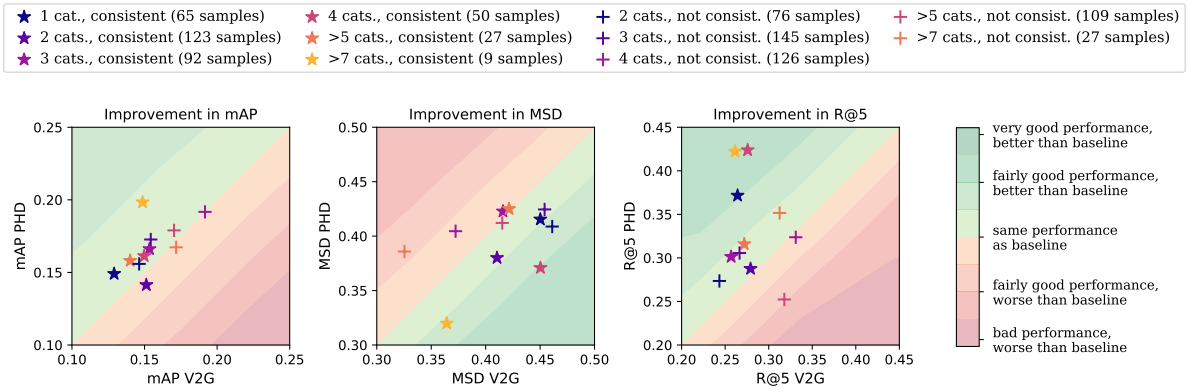


Figure 6.5 Performance of the proposed method (PHD) against the state of the art (V2G) for users with a different number of unique categories in their GIFs. A user is consistent if their test video corresponds to the most frequent category amongst their GIFs. The green areas correspond to better performance for PHD than Video2GIFs, with darker green representing a big improvement with respect to the baseline. On the other hand, points located in darker areas correspond to categories in which PHD provides much worse predictions.

Are consistent users more predictable? Figure 6.5 shows the performance of **PHD-CA + SVM-D** with respect to the baseline (**Video2GIF (ours)**) for different groups of users. The users are grouped according to the number of different categories in their history and whether their last video, *i.e.* the test video, corresponds to their most frequently browsed category, in which case they will be a *consistent user*.

We observe that the greatest improvement by using the user's history is achieved by the group of 6 different categories with a consistent test sample. On the other hand, the worst case is for the analogous group, *i.e.* 6 categories but not consistent in test. PHD gets the largest improvement in Recall@5 for users with consistent test videos and either one, four, or more than seven categories browsed.

6.4 Summary

This Chapter presented Personalized Highlight Detection, an approach for personalized highlight detection in videos. The core idea of PHD is to use a model that is trained for all users jointly and which is customized via its inputs, by providing a user's previously chosen highlights at test time. Such an approach allows training a high-capacity model, even when few examples per user are available.

In a series of experiments, it is shown that a user's GIF history allows for a fine-grained understanding of their interests and is a useful signal for future selections, and thus personalized predictions. Incorporating that information into the highlight detection model significantly improves the performance: PHD outperforms generic highlight detection by 8% in Recall@5. When training a separate model per user, as done in previous work, personalization does not outperform generic highlight detection. PHD, on the other hand, works well even when given very few user-specific training examples. It outperforms generic highlight detection given just a single user-specific training example, thus confirming the benefit of the proposed architecture.

Chapter 7

Active Customization via User Interaction

Current approaches to customizable video summarization obtain the user's preferences prior to the summarization process. As a result, the user may need to manually modify the summary to further meet the preferences. This Chapter presents Active Video Summarization, which enriches the set of user's preferences while creating the summary. AVS is an interactive approach to video exploration that does not require extense knowledge of the user's preferences, nor that the user be familiar with the full content of the video.

7.1 Introduction

Customization might be crucial to effectively summarize consumer videos, as these videos are inherently personal. Previous work to customize video summaries obtain the user's preferences by analyzing data that the user provided previously to the summarization process. Several methods create a customized summary given a text query from the user [108, 126, 135] or a set of video segments that match a set of user's preferences [51, 124]. Another strand of methods assesses each segment's interest from the user's behavior while watching that video [97], or similar videos [87, 139]. However, such customization might not be sufficient. Indeed, it can be improved by actively interacting with the user.

AVS addresses this problem, as it is an interactive approach to gather the user's preferences while creating the summary. AVS asks questions about video segments and updates the produced summary on-line until the user is satisfied with the result. The probabilistic model described in Chapter 5.3 is used to infer both the customized summary and the next question to ask, which reduces the time the user needs to produce a summary.

AVS is evaluated on two challenging datasets for video summarization. The results show that the summaries generated with AVS exploit better the user's preferences than the state-of-the-art video summarization algorithms. With just six questions to the user, the average level of satisfaction for AVS is higher than that of all other tested algorithms. In 41% of the tested cases, the users consider the summary obtained with AVS better than any other summary, including the summaries generated with manual tools.

7.2 Active Video Summarization

The aim of AVS is to provide a customized summary with as little effort as possible from the user side. The system first asks for the user's initial preferences, selected from a set of items, *i.e.* the most frequent items in the original video. Then, the user's preferences are further refined through a question-asking inference.

AVS asks the user specific questions about segments of the video. It shows one selected segment, and asks the following two binary questions:

Q1: Would you want this segment to be in the final summary?, and

Q2: Would you want to include similar segments?

Additionally, the user can decide at any time to go through the segments in the summary and give such feedback about them. Although AVS is not limited to these two questions, experiments show that they are effective in practice, and they serve us as a proof of concept. Note that the original video is not shown to the user, as the segments shown during the interaction provide an accurate idea of the video content in much less time.

AVS can be divided into two independent inference problems (*c.f.* Algorithm 7.1):

(i) infer the customized summary, and

(ii) infer the next segment to show.

7.2.1 Conditional Random Fields for Active Video Summarization

A probabilistic approach based on active inference in Conditional Random Fields (CRFs) [105] is used to infer the most likely summary, and to estimate the next question to ask. As presented in Chapter 5.3, the energy function of the CRF is defined as

$$E_\theta(\mathbf{s}) = \lambda \sum_i \underbrace{\phi_u(s_i)}_{\text{unary}} + \sum_{ij} \underbrace{\phi_p(s_i, s_j)}_{\text{pairwise}}. \tag{7.1}$$

Different from other approaches like using a DPP [144] to model diversity and representativeness, CRFs allow for parameter manipulation for each node and pairwise connection. For a

Algorithm 7.1: Active Video Summarization

$\theta_1 \leftarrow$ initialization from the video
$t = 0$
while user does not stop the loop do

▷ *1.(i) Compute customized summary**
$\mathbf{s}^\star_{\theta_t} = \arg\max_{\mathbf{s}} E_{\theta_t}(\mathbf{s})$

▷ *Display* $\mathbf{s}^\star_{\theta_t}$

▷ *1.(ii) Compute the reward for each candidate**
forall candidate segments do
  $S_k = E_{\theta_{t+1}}\left[R\left(\mathbf{s}^\star_{\theta_{t+1}}, \mathbf{s}^\star_{\theta_t}\right) \mid k\text{-th candidate}\right]$
end
$k^\star = \arg\max_k S_k$

▷ *2. Get active feedback*
if users wants to review summary then
  Ask questions about segments in $\mathbf{s}^\star_{\theta_t}$
else
  Ask about $k^\star$-th candidate
end
$\theta_{t+1} \leftarrow$ update parameters according to user's answer
$t = t + 1$

end

* (i) and (ii) are independent from each other and can be run in parallel for a better user experience.

customizable and interactive formulation, the parameters become specific to each pairwise:

$$\phi_u(s_i) = \begin{cases} L & \text{if } s_i = 0 \\ Q_i & \text{if } s_i = 1 \end{cases} \quad \text{and} \quad \phi_p(s_i, s_j) = e^{-d(\psi_i, \psi_j)} \begin{cases} L\alpha_{ij} & \text{if } s_i = s_j = 0 \\ -L\beta_{ij} & \text{if } s_i = s_j = 1 \\ \gamma_{ij} & \text{if } s_i \neq s_j \end{cases}, \quad (7.2)$$

where $\mathbf{s} = \{s_i\}$ is the set of random variables that represent the summary of the video by indicating whether a segment (or subshot) of the video appears in the summary; ψ is the visual descriptor of the segment; Q is dependent on the segment's quality and user's preferences; γ_{ij} is the cost of selecting only one segment in the pair, and α_{ij} and β_{ij} are the cost to discard or select both segments, respectively; L is a variable parameter that controls the length of the summary; and λ is a parameter that weights the unary potentials with respect to the pairwise.

7.2.2 Update of the CRF Parameters During the Interaction Phase

The formulation with the CRF yields the following flow of the algorithm. Initially, the values of the CRF potentials are θ_1, which are estimated from the input video (see Chapter 5.3). Then, the summary $\mathbf{s}^\star_{\theta_1}$ (MAP summary) is shown to the user. The algorithm selects a segment to

Table 7.1 Update of the parameters for each possible user response to candidate segment s_k

		Q2 = Yes	**Q2 = No**
Q1 = Yes	$\triangleright \quad Q_{k,t+1} = \Delta Q_{k,t}$	$\{\gamma_{kj,t+1}\}_{\forall j} = \{-K\gamma_{kj,t}\}_{\forall j}$ $\{\beta_{kj,t+1}\}_{\forall j} = \{-K\beta_{kj,t}\}_{\forall j}$	$\{\gamma_{kj,t+1}\}_{\forall j} = \{K\gamma_{kj,t}\}_{\forall j}$
Q1 = No	$\triangleright \quad Q_{k,t+1} = \Delta^{-1} Q_{k,t}$	$\{\gamma_{kj,t+1}\}_{\forall j} = \{K\gamma_{kj,t}\}_{\forall j}$	$\{\gamma_{kj,t+1}\}_{\forall j,t+1} = \{-K\gamma_{kj,t}\}_{\forall j}$

query, and the values are updated, θ_2, to match the user's answer. Thus, after the t-th answer, the potential's values are θ_{t+1}. The parameters are updated as follows (*c.f.* Table 7.1)[1]:

Unary When the user recommends to include a candidate segment k (an affirmative response to *Q1*), Q_k is increased by Δ to enforce the selection of that segment; otherwise Q_k is decreased by Δ. At test time, Δ is set to 100 to ensure that the segments selected by the user appear in the summary, and the discarded do not.

Pairwise After each interaction, all the pairwise terms in which segment k appears are updated. When the user recommends selecting either the segment (*Q1*) or a similar segment (*Q2*), but not both, $\{\gamma_{kj}\}_{\forall j}$ is multiplied by a $K > 1$ to encourage that one of the two segments in the pair is selected. On the opposite case, if the user recommends discarding or selecting both segments at the same time, we multiply $\{\gamma_{kj}\}_{\forall j}$ by $-K$ to penalize selecting one of them, *i.e.* it penalizes $x_j \neq x_k$. Additionally, if the user recommends selecting both the segment and similar ones, $\{\beta_{kj}\}_{\forall j}$ is enlarged by $-K$, to cancel the negative sign in Eq. (5.2) and allow that multiple similar segments are selected. For the conducted user study the multiplier K is set to 5.

7.2.3 Inference on the Next Segment to Show

To infer the next segment to query, AVS ranks all possible questions with a score, and asks for the one with the highest rank. Let S_k be the score used to rank the k-th candidate segment. Following the dynamic programming formulation [5], the score is based on a reward function that evaluates the change produced in the summary given the answer of the user, *i.e.* it compares $\mathbf{s}^{\star}_{\theta_{t+1}}$ to $\mathbf{s}^{\star}_{\theta_t}$. Since the reward is obtained after an answer of the user, the algorithm can only

[1]The dependency of the potentials on θ is omitted for simplicity, and the parameters in eq. 7.2 should be considered as part of θ. See section 5.3.3 for details on the initial values of the parameters of the potentials.

estimate the expected reward to decide the candidate segment to query. Thus, the score S_k, is obtained evaluating the expected reward for the k-th candidate.

Let $R(\mathbf{s}^\star_{\theta_{t+1}}, \mathbf{s}^\star_{\theta_t})$ be the reward function, that compares the future summary $\mathbf{s}^\star_{\theta_{t+1}}$ to $\mathbf{s}^\star_{\theta_t}$. Since we want to prioritize the questions that may yield the largest changes in the summary, we define $R(\cdot,\cdot)$ as the Kendall τ correlation between $\mathbf{s}^\star_{\theta_{t+1}}$ and $\mathbf{s}^\star_{\theta_t}$ [28].

Also, we only evaluate the expected reward for the next candidate by discarding the reward of future segment queries that are not the next one. Thus, we define S_k as

$$S_k = E_{\theta_{t+1}}\left[R\left(\mathbf{s}^\star_{\theta_{t+1}}, \mathbf{s}^\star_{\theta_t}\right) \mid k\text{-th candidate}\right], \tag{7.3}$$

where the expectation is over all possible answers to querying the k-th candidate, and $\mathbf{s}^\star_{\theta_{t+1}}$ is the MAP summary for a user's answer of the k-th candidate.

Note that to compute the expected value in Eq. 7.3 we need an estimate of the probability of the user's answers. We can estimate this probability using BP (Sec. 5.3.2). BP obtains the MAP summary by approximating the marginals of the Gibbs distribution, *i.e.* BP approximates $\{P(s_i|\theta)\}$ and $\{P(s_i,s_j|\theta)\}$, and then, it takes the s_i's that maximizes $P(s_i|\theta)$, independently from the other segments, *c.f.* [137]. Thus, we can take the marginals estimated by BP to compute the probability of the user's answers. Note that $\{P(s_i|\theta)\}$ is the probability that the user recommends the i-th segment to be included in the summary (an affirmative response to *Q1*). We can estimate the probability that the user will recommend to include similar segments (*Q2*) by averaging the pairwise marginals, $\{P(s_i,s_j|\theta)\}$, that refer to the segments similar to s_i.

7.3 Experiments

The customization potential of AVS is evaluated through the quality of the final summary and the usability of the system. To that end, two scenarios in which AVS can be used in practice are analyzed. In the first scenario, the user has to summarize a video never seen before. The user has no knowledge of the video essence and thus does not know yet what are the relevant parts. AVS allows the user to discover his or her own preferences while exploring the video content. In the second scenario, the user already knows the content of the video (*e.g.* the user was the camera wearer), and already knows his or her preferences. However, due to the length of the original video, looking for such preferences in the video is very time-consuming. AVS allows the user to browse the video and find such events easier and faster.

7.3.1 User Study

Scenario 1: Discovery Task For this task, 30 independent participants were asked to summarize two videos they had never seen before. The subjects are between 20 and 40 years old, from different nationalities and backgrounds, and tech-savvy. They were given no constraints as to what had to be seen in the summary, other than whatever they were interested in. Then, they were asked to rate how good was that summary, by answering the question *"Did the system manage to provide your ideal summary for that video?"* with a scale of 1 (*"Not at all"*) to 5 (*"Absolutely"*).

To validate their responses on a semi-blind setting, a week after the experiment they were asked to compare the quality of the different baseline summaries. For the two videos that the subject has summarized, we showed the summaries generated with the baselines and the ones that the subject generated. Then, the subjects assessed the quality of the summaries by ranking each of them using one of the following tags: *best*, *good*, *acceptable*, *bad*, and *worst*. They were requested to rate at least one as *worst* and one as *best*. The subjects did not know the baseline corresponding to each summary, and the order was randomized among trials. Note that more than one summary can be rated with the same label, so that there may be more than one *best* or *worst* if these seem to be equally good or bad.

Scenario 2: Search Task To evaluate the efficiency of AVS, the same participants were asked to find a set of events in 2 videos. Such preferences are given in the form of key-frames, extracted from the original video, and a text description of what needs to be included in the final summary (an example can be seen in Fig. 7.1).

To do so, three independent subjects that do not participate in the user study agreed in the selection of four frames from each original video. This set of four key-frames is then used as guidance and scoring reference to summarize the video. In the user study, each user is asked to generate a summary which includes the four given events or items.

The subjects perform this task twice, once with AVS and the other with AVS with *random* questions, *i.e.* step *1.(ii)* in Algorithm 7.1 is replaced by a random choice. None of them knew anything about AVS during the experiment. The subjects also ignored whether they were using AVS, by randomly changing the order of the algorithms.

At the end of each summarization, the users were asked to rate how well the final summary represented the given constraints, on a scale from 1, *i.e.* none or only one of the events is found,

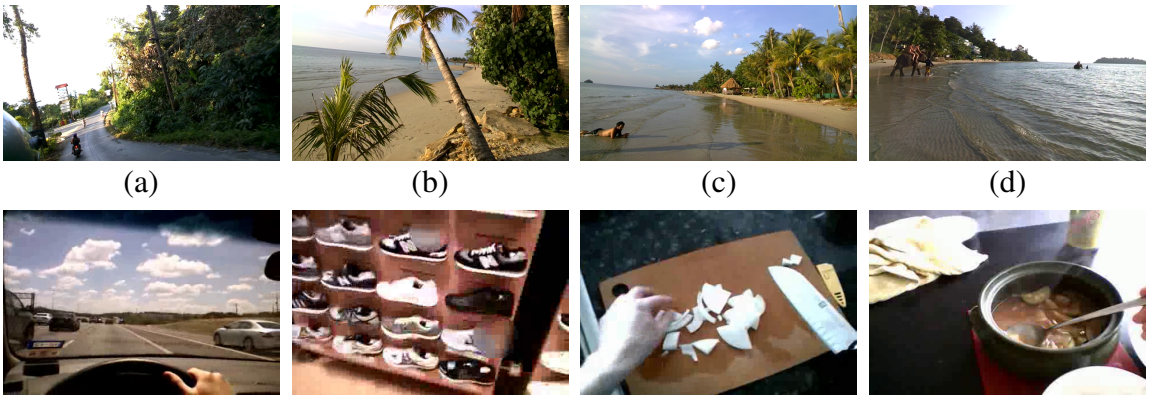


Figure 7.1 Items to be found in Scenario 2 for two example videos. CSumm: (a) Gas station by the road. (b) Beach viewed from the road. (c) Man lying at the shore. (d) Elephants in the water. UTEgo: (e) Driving in highway. (f) Shoe shopping. (g) Chopping vegetables. (h) Serving food.

to 5, *i.e.* all the given constraints are perfectly included in the summary. This experiment also allows obtaining an objective measurement from the amount of interaction needed to reach the target summary. Once performed the summarization task with both approaches, the users were asked to rate the usability of one approach against the other.

Finally, to obtain a blind test against the baselines, the user is also asked to rate the summary that another user generated, and the baseline summaries, using the same scale and criteria as used in his or her summaries.

7.3.2 Experimental Results

As for the non-personalized summaries evaluated in section 5.3.4, AVS is compared against the following baselines:

Uniform. Summary from uniformly sampled segments.

VMMR. Video Maximal Marginal Relevance, a summarization method which rewards diversity [77].

Lee *et al.* [76]. Summary extracted with the method presented by Lee et al. (2012). Since this approach obtains a set of key-frames, we have mapped each key-frame to its corresponding segment to obtain the video summary. These summaries are available in UTEgo.

Manual. In CSumm, where results by Lee et al. (2012) are not available, we have replaced such baseline with a manual annotation of the best segments, with a length constraint of 10 seconds. This was performed by two independent subjects (who did not participate in the rest of the user study), which were asked to manually summarize the given video to their own liking. The annotation to use as the baseline is chosen at random among both.

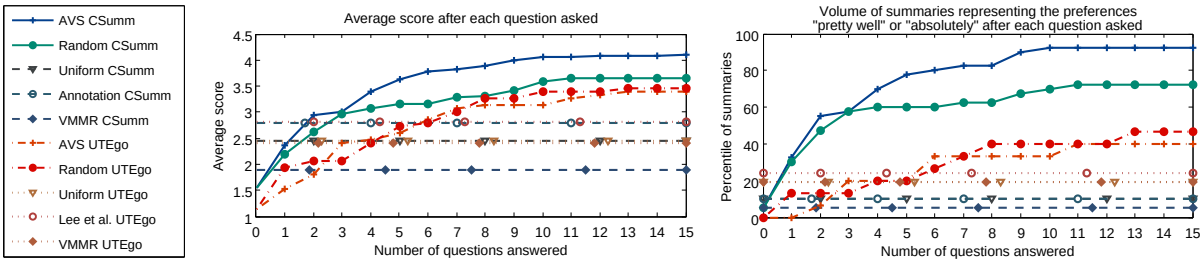
Additionally, the efficiency of the inference on the segment to query in AVS is compared to that of a random selection of segments (named *random*).

Results: Quality of the Summary Table 7.2 reports the percentage of times that the subjects have ranked a summary better than another summary in the discovery task (Scenario 1). We can see that AVS is largely preferred over two of the tested baselines, *uniform* and *VMMR* for both datasets. For half of the summaries in our dataset, AVS is preferred or equally preferred to *manual* annotation. In UTEgo, AVS is also preferred to the method by Lee et al. (2012).

Note that the comparison between *manual* and *uniform* in CSumm shows that in not all cases the subjects prefer the *manual* summary over *uniform*. This shows that the summarization of the videos in CSumm is highly subjective, as a subject may prefer the *uniform* summary over a *manual* annotation from another person (recall that the subject that is assessing the *manual* summary is not the author of this summary, but of the summary with AVS). This proves the challenging nature of CSumm, and it gives more reassurance that the inference of the user's preferences is a key component for video summarization.

When searching for specific events (Scenario 2), Fig. 7.2 reports the users' satisfaction after each question answered. We observe that the AVS summary after two questions is rated better than any of the baselines for CSumm. Moreover, one-third of the summaries represent the user requirements "pretty well" or "absolutely" with only one question, and the ratio goes up to 80% after five questions. With small interaction with the user, AVS achieves better results than any of the baselines.

For the videos of UTEgo, the user needs 6 questions to generate a summary more satisfactory than with the state of the art. We also observe that for this dataset AVS obtains only slightly better performance than AVS with *random* questions. We investigated this, and we found that the performance of AVS highly depends on the image quality of the input data. UTEgo has a resolution of 320×480px (more than four times inferior to the 720×1280px of CSumm). As a consequence, the output of the DCNN results in almost flat feature vectors, making it



(a) (b) (c)

Figure 7.2 Evaluation of the summary after each question in the Search task (Scenario 2). The score is given by the answer to the question *"Did the system manage to provide your ideal summary for that video?"*, which accepted as responses *"Not at all"* (1), *"Not much'* (2), *"So-so'* (3), *"Pretty much'* (4) *"Absolutely'* (5). (a) Legend. (b) Mean score of the summaries. (c) Percentage of summaries for which the summary obtained a score greater or equal to 4.

Table 7.2 Percentage of times each method on the left was ranked better than the one on top for the Discovery task (Scenario 1). Note that symmetric elements may not add up to 100%, since two summaries can be ranked equally.

	CSumm				UTEgo			
	Unif.	Manual	VMMR	AVS	Unif.	Lee *et al.*	VMMR	AVS
Unif.	-	28%	44%	25%	-	29%	41%	24%
Manual/Lee *et al.*	66%	-	78%	**50%**	59%	-	71%	41%
VMMR	47%	19%	-	19%	47%	24%	-	24%
AVS	**59%**	34%	**66%**	-	**71%**	**53%**	**76%**	-

difficult for AVS to distinguish among the different events. A different visual descriptor could be explored for low-resolution videos. Moreover, longer videos translate into longer graphs for the CRF, which needs longer to converge.

Results: Usability We compare the time needed to generate a customized summary with AVS and *manually* (only in CSumm, as we did not obtain the manual annotation for UTEgo) in the discovery task (Scenario 1). In Table 7.3, we can see that the users are 4 times faster creating the summary with AVS than with the *manual* baseline. This is a significant improvement of the usability of the *manual* annotation, since the quality of AVS is competitive with the quality of the *manual* annotation as shown before.

Looking for specific events (Scenario 2), we can compare AVS and AVS with *random* questioning under the same search constraints. We show such subjective assessment in Table 7.4.

Table 7.3 Usability of AVS. Time to generate a summary in CSumm. (Scenario 1: Discovery Task)

AVS	Manual
5.89 ± 3.85 min.	21.66 ± 6.59 min.

Table 7.4 Subjective perception of the usability of AVS against the random baseline: amount of summaries that obtained each possible score. (Scenario 2: Search Task)

	Much worse	Worse	Similar	Better	Much better
CSumm:	5.4%	16.2%	18.9%	**43.2%**	16.2%
UTEgo:	6.7%	13.3%	26.7%	**40%**	13.3%

We can see that the majority of the subjects prefer active inference over the *random* baseline in both datasets, which is in accordance with Fig. 7.2. These results demonstrate the usefulness of estimating the next questions to ask, as opposing to selecting *random* segments.

7.4 Summary

This Chapter presented Active Video Summarization, an approach to customize a video summary by interacting with the user. Current passive approaches are constrained by the initial feedback the user provided. By contrast, AVS enriches the set of user's preferences while creating the summary. It uses a Conditional Random Field to infer the best summary given the user's preferences, and to find the optimal interactive path. This approach reduces the time the user needs to produce a summary as opposed to selecting random questions.

In a series of experiments, it is demonstrated that AVS strikes a balance between usability and quality of the summary. The results show that the summaries generated with AVS exploit better the user's preferences than the state-of-the-art video summarization algorithms. Namely, AVS significantly reduces the time spent by the users to generate their preferred summary. With just six questions to the user, the average level of satisfaction for AVS is higher than those of all other tested algorithms. Also, in 41% of the tested cases, the users consider the summary obtained with AVS better than any other summary, including the summaries manually generated.

Chapter 8

Conclusions

This thesis has addressed three major challenges in end-to-end egocentric video summarization: First, segmenting the visual data stream into distinct episodic events. Second, generating a diverse and coherent visual summary for an optimal story-telling. Third, assessing the relevance of each visual unit for a personalized user experience. It has also presented three large-scale datasets for egocentric video analysis and understanding. This Chapter summarizes this thesis' contributions, and discusses further research opportunities.

8.1 Summary of Contributions

The potential of consumer and egocentric video cameras is limited by our capacity to process and organize their recordings. This thesis has identified and addressed the following major challenges related to this: how to separate and classify the different events involved in the video collection; how to generate a diverse, informative and visually pleasant summary; and how to make the output personalized for each individual user. Additionally, this thesis has contributed novel datasets for FPV video analysis in order to enlarge the limited number of public egocentric video datasets.

Large-scale datasets for video analysis and understanding This thesis has contributed two novel FPV video datasets to the community, which are presented in Chapter 3. CSumm is a High Time Resolution egocentric video dataset recorded by three users over a time-span of six months, adding to 65 recording sessions and a total of 40 hours. The videos are of unconstrained nature and depict a large selection of activities, being a wide representation of the wearer experiences during the said time. On the other hand, R3 is a Low Time Resolution lifelogging dataset recorded by 57 users during $1,723$ full days. The large-scale nature of R3 makes it optimal to train unsupervised models Additionally, Chapter 3 presents insights into the interests and video highlight preferences of over $15,000$ anonymous users from the dataset PHD^2. Such analysis validates our hypothesis that different users will have very different preferences and that personalization is needed for a better summarization output.

Episodic event modeling and segmentation Chapter 4 introduced Contextual Event Segmentation, a novel unsupervised event segmentation method that uses the sequential nature of a photo-stream to infer the presence of event boundaries. CES is analogous to our event segmentation perceptual system. Given a sequence of frames, *i.e.* the stimuli, the Visual Context Predictor generates an event model representation, which is the cue to changes in the current event. The Visual Context Predictor is trained with over one million images from R3. With such extense dataset, it is able to model human activities given sequences of visual features. In a series of experiments, it is shown that the event representation is a strong indicator of event changes. CES is a fully unsupervised pipeline that can be adjusted to different levels of event granularity, *i.e.* timescales. Chapter 5 further demonstrates that CES is also useful to obtain diverse summaries from retrieved video data. We conjecture that the event representation can also be useful for storytelling tasks and tracking of daily activities.

User-driven interestingness scoring for personalized summaries Chapter 6 presented Personalized Highlight Detection, an approach for personalized highlight detection in videos. PHD is a global ranking model personalized via its inputs: the target video, and the GIFs the user previously generated. This allows the model to make accurate user-specific predictions, as a user's GIF history allows for a fine-grained understanding of their interests and thus provides a strong signal for personalization. Moreover, as it is trained for all users jointly, it can be a high-capacity model, even if few examples per user are available. In a series of experiments, it is shown that the user history provides a useful signal for future selections. Incorporating that information into the highlight detection model significantly improves the performance: PHD outperforms generic highlight detection by 8% in Recall@5. PHD is competitive to the state-of-the-art for new users, *i.e.* empty history, and outperforms generic highlight detection given just a single user-specific training example, thus confirming the benefit of the proposed architecture.

Active video summarization to enrich the set of user's preferences Chapter 7 described Active Video Summarization, an approach to customize a video summary by interacting with the user. AVS is an interactive approach to video exploration that does not require extense knowledge of the user's preferences, nor that the user be familiar with the full content of the video. To gather the user's preferences, AVS iteratively asks specific questions about segments of the video until the user is satisfied with the proposed summary. AVS uses a Conditional Random Field to model the video, as introduced in Chapter 5. The same probabilistic approach

is used to infer the best interaction with the user. In a series of experiments, it is demonstrated that AVS strikes a balance between usability and quality of the summary. The results show that the summaries generated with AVS exploit better the user's preferences than the state-of-the-art video summarization algorithms. AVS significantly reduces the time spent by the users to generate their preferred summary.

8.2 Research Opportunities

As with all new research challenges, there are areas that can be improved or explored in more depth. Some of these aspects are the following:

Homogenization of the ground truth for highlight detection PHD has been evaluated on ground truths obtained before, and independently from, the summarization process. As such, we can only know that that segment was a highlight, but it is possible that it was not the only one. In the case of repetitive visual content such as wearable video, or choreographed music videos, several segments may share the same semantic information and interestingness, even if the user only selected one of them. As a result, an accurate prediction could be deemed incorrect if the segment selected by the user was at another time-step. To prevent the evaluation pipeline from detecting such false negatives, the ground truth can be extended to include all video segments with similarity to the user selection above a high threshold.

Limitation of Active Video Summarization for long videos We observe that longer videos translate into longer graphs for the CRF, which take longer to converge. A possible fix for this overload issue could be to first segment the video into events, and only consider the most important or aesthetic segments for each event.

Exploitation of the stored user-data The summary could be unobtrusively personalized to the wearer's life in different ways. As suggested by the results from Chapter 6, the user's previously generated summaries can be used to learn his or her preferences. Previously stored memories and their relation to the target video could thus be used to fine-tune the interestingness score of each video segment in AVS. This thesis has demonstrated the suitability of one way to aggregate the user profile. However, further history aggregations could be explored for a better customized prediction. One alternative would be to embed both the history items and the

target video into a shared space, and aggregate them separately using convolutions instead of comparing the averaged history to the target video.

On the other hand, previous video summaries could also be analyzed to constrain the current summary, and to learn to better interact with the user leveraging AVS. For an augmented summarization experience, patterns in past memories could be explored to detect similar or relevant events in the newly uploaded video.

Aesthetics and enjoyable moments Since the objective of the summarization is obtaining an informative and visually pleasant video or storyboard, it is necessary to detect and select enjoyable scenes, to cut in the right transition moments, and to stabilize the output videos. Such constraints can be set in Active Video Summarization by modifying the unary weights, *i.e.* the segment's importance, according to aesthetics or other cues like sentiment analysis. The pairwise weights for transitioning segments could also be modified so that event boundaries would be encouraged, or otherwise.

The generated summaries still keep the inherent shakiness of egocentric videos, and are usually dizzying to watch. Stabilizing algorithms such as [65] should be applied to solve this problem.

Use of other multimodal cues Looking at Chapter 2, we can observe that speech is seldom considered when analyzing FPV video, even if it is widely used in TPV summarization techniques. It could be because of the low audio quality of some devices, or because many times wearable video has no audio track. Whatever the reason is, the additional use of speech and other multimodal cues is very likely to improve summarization when available. The wearer's emotion and attention (captured with other sensors) can also be useful cues for importance estimation.

List of Publications

1. **A. García del Molino**, J.-H. Lim, and A.-H. Tan, "Predicting visual context for unsupervised event segmentation in continuous photo-streams," in *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pp. 10–17, ACM, 2018.

2. **A. García del Molino** and M. Gygli, "PHD-GIFs: Personalized highlight detection for automatic GIF creation," in *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pp. 600–608, ACM, 2018.

3. **A. García del Molino**, X. Boix, J.-H. Lim, and A.-H. Tan, "Active Video Summarization: Customized summaries via on-line interaction with the user.," in *AAAI Conference on Artificial Intelligence*, pp. 4046–4052, 2017.

4. **A. García del Molino**, C. Tan, J.-H. Lim and A.-H. Tan, "Summarization of egocentric videos: A comprehensive survey," in *IEEE Transactions on Human-Machine Systems*, vol. 47 (1), pp. 65–76, IEEE, 2017.

5. **A. García del Molino**, M. Bappaditya, J. Lin, J.-H. Lim, S. Vigneshwaran, and C. Vijay, "VC-I2R at ImageCLEF2017: Ensemble of deep learned features for lifelog video summarization," in *CLEF working notes*, CEUR, 2017.

6. J. Lin, **A. García del Molino**, Q. Xu, F. Fang, S. Vigneshwaran, and J.-H. Lim, "VC-I2R at the NTCIR-13 lifelog semantic access task," in *Proceedings of NTCIR-13*, 2017.

7. **A. García del Molino**, "First Person View video summarization subject to the user needs," in *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, (New York, NY, USA), pp. 1440–1444, ACM, 2016. (Doctoral Symposium)

8. **A. García del Molino**, Q. Xu, and J.-H. Lim, "Describing lifelogs with convolutional neural networks: A comparative study," in *Proceedings of the 1st Workshop on Lifelogging Tools and Applications*,pp. 39–44, ACM, 2016.

9. **A. García del Molino**, B. Mandal, L. Li, and L. J. Hwee, "Organizing and retrieving episodic memories from first person view," in *International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, IEEE, 2015.

Appendix

Detailed Experiments

A.1 Contextual Event Segmentation

Various ways of predicting the event representation are explored, as well as fine-tuning the boundary prediction with supervised learning. The performance of the different variations in EDUB-Seg (*c.f.* Chapter 3) is reported in Table A.1.

Predicting the event representation vs predicting the actual frame One could think that predicting a future frame $\hat{\mathbf{x}}_\mathbf{t}$ and comparing it to the actual future frame $\mathbf{x}_\mathbf{t}$ should be better than comparing the event representations. We tested this hypothesis, in which $pred(t) = abs(mse(\mathbf{x_t}, \hat{\mathbf{x}}_\mathbf{ft}) - mse(\mathbf{x_t}, \hat{\mathbf{x}}_\mathbf{pt}))$, where $\hat{\mathbf{x}}_\mathbf{ft}$ is predicted from $\mathbf{x}_{\mathbf{k}|\mathbf{0}\leq\mathbf{k}<\mathbf{t}}$ and $\hat{\mathbf{x}}_\mathbf{pt}$ from $\mathbf{x}_{\mathbf{k}|\mathbf{T}\geq\mathbf{k}>\mathbf{t}}$. The intuition behind this formulation is that a local outlier will be badly predicted both from the future and the past, whereas an event change will provide a good prediction only in one direction. This theory proves not precise in practice. The generative model embeds noise into the frame descriptor, and, as expected, generates samples closer to the previous (seen) frame than the (unseen) target. As such, using such a noisy signal is detrimental to the final objective. The performance of that method is reported as CES-error in Table A.1.

Informativeness of the event representation To validate the encoding efficiency of VCP and hence the informativeness of the event representation, we have tested CES using two alternative sequence encodings: first, an average of the previous $N = 10$ frames (or subsequent in the case of the past prediction); second, a PCA time-dimensionality reduction on the aforesaid set. These two variants are reported in Table A.1 as CES-mean and CES-PCA, respectively.

We observe that the event representation predicted by VCP results much more informative than any of the other contextual encodings. While the averaged encoding obtains a predictive performance similar to the output of our decoder (*c.f.* Table A.2), the encoding transformation of VCP is superior as a contextual visual feature. Moreover, unlike PCA, which takes the inputs as a set, VCP takes the inputs as a sequence and is able to learn a more informative context descriptor.

Pruning of the candidate boundaries using supervised learning For high recall results, false candidate boundaries can be discarded using cluster analysis between the frames at each side of the boundary. Having annotated data to train a pruning model can improve the performance of the segmentation algorithm in terms of precision, having minimal impact on the recall. This hypothesis is tested on EDUB-Seg by training an SVM model to detect false positives on a held-out validation set (EDUB-SegDesc, *c.f.* Chapter 3 for details on the datasets). The SVM evaluates the boundary likeliness from cluster consistency indicators. In particular, two clusters are defined at opposite sides of the candidate boundary, containing the 15 frames that precede or follow it. The indicators used are the correlation between the two clusters, the compactness of each of them and their union, and the BetaCV and Normalized Cut scores [142].

As can be observed in Table A.1, such model improves the average precision of CES by 15% (absolute gain of 10%), while recall only decreases by 7.8% (absolute loss of 6%). The benefit of using a supervised SVM pruning is much significant for segmentation algorithms of lower precision, such as k-means, even if coming at a higher recall cost.

Table A.1 Detailed experiments. Comparison of CES using the event representation from VCP as opposed to using other feature predictions or aggregations; performance of the SVM pruning; and accuracy of the manual annotations against the selected ground truth. (Evaluated on EDUB-Seg).

	averaged F1	averaged Prec.	averaged Rec.
CES-error	0.42	0.45	0.49
CES-mean	0.52	0.56	0.56
CES-PCA	0.66	0.67	0.69
CES (with VCP)	0.69	0.66	**0.77**
k-means w/ SVM	0.67	0.70	0.67
CES w/ SVM	0.71	0.75	0.71

Table A.2 Performance of the auto-encoder's prediction at test time (mean *mse* amplified $\cdot 10^2$, with $N = 1$, $M = T - 1$ and $T = \textbf{len}\,[\mathbf{x}]$) for different training configurations of VCP (on R3 dataset).

trained with N / M :	10 / 10			1 / 40		1/100	1/1	10/1
# neurons :	256	512	1024	512	1024	1024	mean*	
mse future pred.:	1.058	1.030	**1.024**	1.03	1.029	1.028	1.58	1.054
mse past pred.:	1.059	1.029	**1.024**	1.03	1.029	1.028		

*The predicted feature corresponds to the average of the previous N frames, *i.e.* $\hat{\mathbf{x}}(t) = \sum_{n=1}^{N} \mathbf{x}(t-n)/N$.

A.2 Personalized Highlight Detection

Averaging the user history is not the only way in which it can be aggregated into the highlight prediction model. Indeed, various ways to include the user history are explored, as well as network architectures and fusion of different inputs. Figure A.1 shows these different configurations, while their performance is given in Table A.3.

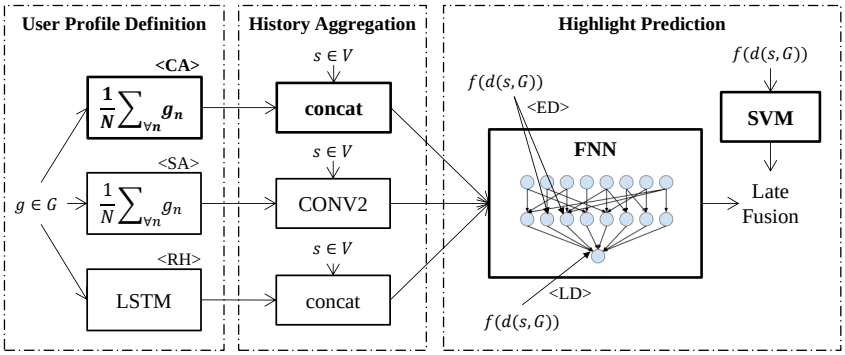


Figure A.1 Model architectures tested. This figure shows schematically the proposed model (**bold**) and alternative ways to encode the history and fuse predictions.

Learning an aggregation *vs* averaging? Our proposed model aggregates the history via averaging (**PHD-CA**, *c.f.* Section 6.2). Alternatively, Recurrent Neural Networks are often successfully used to encode visual sequences [27, 115]. Thus, we also explored a model that uses an LSTM to learn to aggregate the history (**PHD-RH**). The history is then concatenated to the segment representation and passed through 2 fully-connected layers. As can be seen from Table A.3, having a predefined aggregation performs better than learning it. We attribute this to the challenge of learning a sequence embedding from limited data and conclude that an average aggregation provides an effective representation of the users' history.

Convolutional combination or concatenation? In Section 6.2 we propose to concatenate the average history to the segment representation **s**. Since they both use the same C3D representation, however, it is also possible to first aggregate each dimension of the two vectors with 1D convolutions, before passing them through fully connected layers (**PHD-SA**). We compared these two approaches and found the concatenation to give superior performance. The convolutional aggregation uses the structure of the data to reduce the number of network parameters and therefore has roughly half the parameters of the concatenation model. Convo-

Table A.3 Detailed experiments. We analyze different ways to represent and aggregate the history, as well as ways to use the distances to the history to improve the prediction.

Model	mAP	nMSD	R@5
PHD-SA	15.73%	42.80%	**28.65%**
PHD-RH	15.74%	42.75%	27.45%
PHD-CA	**16.58%**	**41.01%**	28.18%
PHD-CA-ED (1st layer)	16.14%	41.26%	29.20%
PHD-CA-LD (last layer)	16.20%	41.07%	29.78%
Video2GIF (ours) + SVM-D	16.39%	40.90%	28.70%
PHD-CA + SVM-D	**16.68%**	**40.26%**	**30.71%**

lutional aggregation, however, requires the network to aggregate the history into the segment information per dimension, using the same weights. Thus it is limited in its modeling capacity, compared to a network using concatenated features as inputs.

Adding distances, with early or late fusion? As we discussed, our assumption is that the similarity of a segment to the previously chosen GIFs is informative when predicting the score of a segment. Thus, we tested models that use the distance to the history elements as an additional input.

Since using distance features leads to a different representation compared to the feature activations of C3D, it is unclear how to best merge the two different modalities. We tried early fusion (concatenation of the two inputs, **PHD-CA-ED**), late fusion before the prediction layer in one single model (**PHD-CA-LD**) and late fusion with training two separate models (**PHD-CA + SVM-D**), as shown in Figure A.1. We find that late fusion performs superior to early fusion, and that combining two different models outperforms merging on the last layer of the neural network. The superiority of late fusion is to be expected, as neural networks often struggle to combine information from different modalities [15, 111]. Adding the distances in the neural network even slightly decreases mAP, while Recall@5 improves. While this inconsistency is somewhat surprising, Recall@5 is arguably more important, as it evaluates the accuracy of the top-ranked elements, which is what matters for finding highlight in videos, while mAP considers the complete ranking. When using a separate model for the distances and fusing their predictions, we obtain a consistent improvement in all metrics.

We also tried adding personalization to a generic highlight detection model by combining its predictions with the predictions of the distance SVM (**Video2GIF (ours) + SVM-D** in Table A.3). This leads to a significant improvement over the generic model. While it doesn't perform quite as well as our full model, this approach provides a simple way to personalize existing highlight detection, in order to improve their performance.

Implementation details for the detailed experiments For the aggregation of $s \in V$ and $g \in \mathscr{G}$, a size of either four or ten neurons is considered for the 1-D convolution in **PHD-SA**, flattened with a single neuron convolution before the FNN layers. For the **PHD-RH** model, we tested using 1000 or 512 neurons in the hidden layer of the LSTM.

Different architectures for the FNN predictive network were considered, with between one and three layers having up to 512 neurons each, where each following layer would have the same number or fewer neurons. The best performing architectures are a single hidden layer of 256 neurons for **PHD-SA** and a single hidden layer of 512 neurons for **PHD-RH**.

Impact of the late fusion weight The late fusion weight ω from eq. 6.4 has an impact on the final result. To show that **PHD-CA + SVM-D** is consistently better than **Video2GIF (ours) + SVM-D** Figure A.2 shows the performance of both models as a function of the weight ω. We observe that the ideal weight would be around 4.25. Since the weight for our experiments was learned from the validation set, the performance reported in this thesis is not the best possible one.

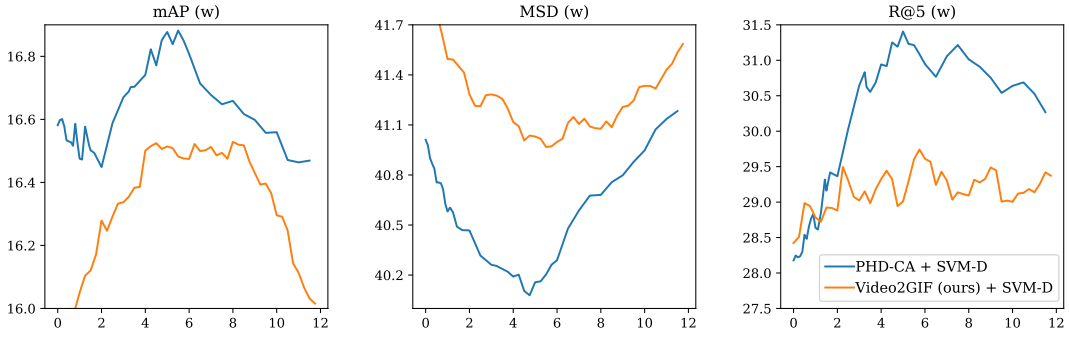


Figure A.2 Impact of the late fusion weight. Performance of **PHD-CA + SVM-D** and **Video2GIF (ours) + SVM-D** as a function of the late fusion weight. PHD is consistently better than adding the SVM-D model to the baseline.

References

[1] L. Agnihotri, J. Kender, N. Dimitrova, and J. Zimmerman, "Framework for personalized multimedia summarization," in *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*. ACM, 2005.

[2] K. Aizawa, K. Ishijima, and M. Shiina, "Summarizing wearable video," in *International Conference on Image Processing*, vol. 3. IEEE, 2001, Conference Proceedings, pp. 398–401.

[3] K. Aizawa, D. Tancharoen, S. Kawasaki, and T. Yamasaki, "Efficient retrieval of life log based on context and content," in *Proceedings of the the 1st ACM workshop on Continuous Archival and Retrieval of Personal Experiences*. ACM, 2004, Conference Proceedings, pp. 22–31.

[4] N. Babaguchi, K. Ohara, and T. Ogura, "Learning personal preference from viewer's operations for browsing and its application to baseball video retrieval and summarization," *IEEE transactions on multimedia*, 2007.

[5] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[6] A. Betancourt, P. Morerio, C. S. Regazzoni, and M. Rauterberg, "The evolution of first person vision methods: A survey," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 5, pp. 744–760, 2015.

[7] V. Bettadapura, D. Castro, and I. Essa, "Discovering picturesque highlights from egocentric vacation videos," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[8] B. L. Bhatnagar, S. Singh, C. Arora, C. Jawahar, and K. CVIT, "Unsupervised learning of deep feature representation for clustering egocentric actions." in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1447–1453. [Online]. Available: https://doi.org/10.24963/ijcai.2017/200

[9] M. Bolanos, M. Dimiccoli, and P. Radeva, "Toward storytelling from visual lifelogging: An overview," *IEEE Transactions in Human-Machine Systems*, vol. 47, pp. 77–90, 2017.

[10] M. Bolanos, M. Garolera, and P. Radeva, "Video segmentation of life-logging videos," in *International Conference on Articulated Motion and Deformable Objects*. Springer, 2014, pp. 1–9.

[11] M. Bolanos, R. Mestre, E. Talavera, X. Giró-i Nieto, and P. Radeva, "Visual summary of egocentric photostreams by representative keyframes," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. IEEE, 2015, pp. 1–6.

[12] M. Bolanos, R. Mestre, E. Talavera, X. Giró-i Nieto, and P. Radeva, "Visual summary of egocentric photostreams by representative keyframes," in *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[13] M. Bolaños, Á. Peris, F. Casacuberta, S. Soler, and P. Radeva, "Egocentric video description based on temporally-linked sequences," *Journal of Visual Communication and Image Representation*, vol. 50, pp. 205–216, 2018.

[14] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 9, pp. 1124–1137, 2004.

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[16] V. Chandrasekhar, W. Min, X. Li, C. Tan, B. Mandal, L. Li, and J. H. Lim, "Efficient retrieval from large-scale egocentric visual data using a sparse graph representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, Conference Proceedings, pp. 527–534.

[17] W.-S. Chu, Y. Song, and A. Jaimes, "Video co-summarization: Video summarization by visual co-occurrence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, Conference Proceedings, pp. 3584–3592.

[18] C. Chunseong Park, B. Kim, and G. Kim, "Attend to you: Personalized image captioning with context sequence memory networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 895–903.

[19] C. T. Dang and H. Radha, "Heterogeneity image patch index and its application to consumer video summarization," *Image Processing, IEEE Transactions on*, vol. 23, no. 6, pp. 2704–2718, 2014.

[20] D.-T. Dang-Nguyen, L. Piras, M. Riegler, G. Boato, L. Zhou, and C. Gurrin, "Overview of ImageCLEFlifelog 2017: Lifelog Retrieval and Summarization," in *CLEF2017 Working Notes*, ser. CEUR Workshop Proceedings. Dublin, Ireland: CEUR-WS.org <http://ceur-ws.org>, September 11-14 2017.

[21] F. De la Torre, J. Hodgins, A. Bargteil, X. Martin, J. Macey, A. Collado, and P. Beltran, "Guide to the carnegie mellon university multimodal activity (CMU-MMAC) database," *Robotics Institute*, p. 135, 2008.

[22] A. G. del Molino and M. Gygli, "PHD-GIFs: Personalized highlight detection for automatic GIF creation," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. New York, NY, USA: ACM, 2018, pp. 600–608. [Online]. Available: http://doi.acm.org/10.1145/3240508.3240599

[23] A. G. del Molino, B. Mandal, L. Li, and J. H. Lim, "Organizing and retrieving episodic memories from first person view," in *International Conference on Multimedia and Expo Workshops*. IEEE, 2015, Conference Paper, pp. 1–6.

[24] A. G. del Molino, M. Bappaditya, J. Lin, J.-H. Lim, S. Vigneshwaran, and C. Vijay, "VC-I2R at ImageCLEF2017: Ensemble of deep learned features for lifelog video summarization," in *CLEF working notes, CEUR*, 2017.

[25] A. G. del Molino, X. Boix, J.-H. Lim, and A.-H. Tan, "Active Video Summarization: Customized summaries via on-line interaction with the user." in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 4046–4052.

[26] A. G. del Molino, C. Tan, J.-H. Lim, and A.-H. Tan, "Summarization of egocentric videos: A comprehensive survey," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 65–76, 2017.

[27] A. G. del Molino, J.-H. Lim, and A.-H. Tan, "Predicting visual context for unsupervised event segmentation in continuous photo-streams," in *Proceedings of the 26th ACM International Conference on Multimedia*, ser. MM '18. New York, NY, USA: ACM, 2018, pp. 10–17. [Online]. Available: http://doi.acm.org/10.1145/3240508.3240624

[28] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*. Springer, 2009, pp. 1–583.

[29] M. Dimiccoli, M. Bolaños, E. Talavera, M. Aghaei, S. G. Nikolov, and P. Radeva, "Sr-clustering: Semantic regularized clustering for egocentric photo streams segmentation," *Computer Vision and Image Understanding*, vol. 155, pp. 55–69, 2017.

[30] M. Dogariu and B. Ionescu, "A Textual filtering of hog-based hierarchical clustering of lifelog data," *CLEF working notes (September 11-14 2017)*, 2017.

[31] A. R. Doherty and A. F. Smeaton, "Automatically segmenting lifelog data into events," in *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. IEEE, 2008, pp. 20–23.

[32] A. R. Doherty, D. Byrne, A. F. Smeaton, G. J. Jones, and M. Hughes, "Investigating keyframe selection methods in the novel domain of passively captured visual lifelogs," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*. ACM, 2008, Conference Proceedings, pp. 259–268.

[33] A. R. Doherty, C. Ó Conaire, M. Blighe, A. F. Smeaton, and N. E. O'Connor, "Combining image descriptors to effectively retrieve events from visual lifelogs," in *Proceedings of the 1st ACM international conference on Multimedia information retrieval*. ACM, 2008, Conference Proceedings, pp. 10–17.

[34] A. R. Doherty, K. Pauly-Takacs, N. Caprani, C. Gurrin, C. J. Moulin, N. E. O'Connor, and A. F. Smeaton, "Experiences of aiding autobiographical memory using the SenseCam," *Human–Computer Interaction*, vol. 27, no. 1-2, pp. 151–174, 2012.

[35] A. R. Doherty, S. E. Hodges, A. C. King, A. F. Smeaton, E. Berry, C. J. Moulin, S. Lindley, P. Kelly, and C. Foster, "Wearable cameras in health," *American journal of preventive medicine*, vol. 44, no. 3, pp. 320–323, 2013.

[36] A. Fathi, X. Ren, and J. M. Rehg, "Learning to recognize objects in egocentric activities," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On*. IEEE, 2011, Conference Proceedings, pp. 3281–3288.

[37] A. Fathi, J. K. Hodgins, and J. M. Rehg, "Social interactions: A first-person perspective," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, Conference Proceedings, pp. 1226–1233.

[38] A. Fathi, Y. Li, and J. M. Rehg, "Learning to recognize daily actions using gaze," *Computer Vision–ECCV*, pp. 314–327, 2012.

[39] A. Furnari, S. Battiato, and G. M. Farinella, "Personal-location-based temporal segmentation of egocentric videos for lifelogging applications," *Journal of Visual Communication and Image Representation*, vol. 52, pp. 1–12, 2018.

[40] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[41] J. Gemmell, L. Williams, K. Wood, R. Lueder, and G. Bell, "Passive capture and ensuing issues for a personal lifetime store," in *Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences*. ACM, 2004, Conference Proceedings, pp. 48–55.

[42] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems*, 2014.

[43] Y. Graham, G. Awad, and A. Smeaton, "Evaluation of automatic video captioning using direct assessment," *PloS one*, vol. 13, no. 9, p. e0202789, 2018.

[44] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, D.-T. Dang-Nguyen, R. Gupta, and R. Albatal, "Overview of ntcir-13 lifelog-2 task," in *Proceedings of the 13th NTCIR Conference*, 2017.

[45] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, R. Gupta, R. Albatal, D. Nguyen, and D. Tien, "Overview of NTCIR-13 lifelog-2 task." NTCIR, 2017.

[46] M. Gygli, "Ridiculously fast shot boundary detection with fully convolutional neural networks," in *2018 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2018, pp. 1–4.

[47] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Computer Vision–ECCV*. Springer, 2014, pp. 505–520.

[48] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, Conference Proceedings, pp. 3090–3098.

[49] M. Gygli, Y. Song, and L. Cao, "Video2gif: Automatic generation of animated gifs from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1001–1009.

[50] T. Halperin, Y. Poleg, C. Arora, and S. Peleg, "Egosampling: Wide view hyperlapse from egocentric videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 1248–1259, 2018.

[51] B. Han, J. Hamm, and J. Sim, "Personalized video summarization with human in the loop," in *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*. IEEE, 2011, pp. 51–57.

[52] M. Harvey, M. Langheinrich, and G. Ward, "Remembering through lifelogging: A survey of human memory augmentation," *Pervasive and Mobile Computing*, vol. 27, pp. 14–26, 2016.

[53] G. Healy, C. Gurrin, and A. F. Smeaton, "Lifelogging and EEG: utilising neural signals for sorting lifelog image data," in *Quantified Self Europe Conference*, 2014.

[54] H.-I. Ho, W.-C. Chiu, and Y.-C. Frank Wang, "Summarizing first-person videos from third persons' points of view," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 70–85.

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

[56] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015.

[57] P. Isola, J. Xiao, A. Torralba, and A. Oliva, "What makes an image memorable?" in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, Conference Proceedings, pp. 145–152.

[58] A. Jaimes, T. Echigo, M. Teraguchi, and F. Satoh, "Learning personalized video highlights from detailed mpeg-7 metadata," in *Image Processing. 2002. Proceedings. 2002 International Conference on*. IEEE, 2002.

[59] D.-J. Jeong, H. J. Yoo, and N. I. Cho, "A static video summarization method based on the sparse coding of features and representativeness of frames," *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, pp. 1–14, 2016.

[60] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *arXiv preprint arXiv:1708.09545*, 2017.

[61] W. Jiang, C. Cotton, and A. C. Loui, "Automatic consumer video summarization by audio and visual analysis," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[62] Y.-G. Jiang, B. Xu, and X. Xue, "Predicting emotions in user-generated videos." in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 73–79.

[63] T. Joachims, "Optimizing search engines using clickthrough data," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2002.

[64] N. Joshi, W. Kienzle, M. Toelle, M. Uyttendaele, and M. F. Cohen, "Real-time hyperlapse creation via optimal frame selection," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 63, 2015.

[65] A. Karpenko, D. Jacobs, J. Baek, and M. Levoy, "Digital video stabilization and rolling shutter correction using gyroscopes," *CSTR*, vol. 1, p. 2, 2011.

[66] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[67] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, Conference Proceedings, pp. 3241–3248.

[68] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 972–981.

[69] V. Kolmogorov and M. J. Wainwright, "On optimality properties of tree-reweighted message-passing," in *Uncertainty in Artificial Intelligence*. Morgan-Kaufmann, 2005.

[70] J. Kopf, M. F. Cohen, and R. Szeliski, "First-person hyper-lapse videos," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 78, 2014.

[71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[72] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[73] C.-P. Lee and C.-J. Lin, "Large-scale linear ranksvm," *Neural computation*, 2014.

[74] H. Lee, A. F. Smeaton, N. E. O'Connor, G. Jones, M. Blighe, D. Byrne, A. Doherty, and C. Gurrin, "Constructing a sensecam visual diary as a media process," *Multimedia Systems*, vol. 14, pp. 341–349, 2008.

[75] Y. J. Lee and K. Grauman, "Predicting important objects for egocentric video summarization," *International Journal of Computer Vision*, pp. 1–18, 2015.

[76] Y. J. Lee, J. Ghosh, and K. Grauman, "Discovering important people and objects for egocentric video summarization." in *Computer Vision and Pattern Recognition*, vol. 2, 2012, Conference Proceedings, p. 6.

[77] Y. Li and B. Merialdo, "Multi-video summarization based on video-mmr," in *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. IEEE, 2010, pp. 1–4.

[78] J. Lin, A. G. del Molino, Q. Xu, F. Fang, S. Vigneshwaran, and J.-H. Lim, "VC-I2R at the NTCIR-13 lifelog semantic access task," in *Proceedings of NTCIR-13, Tokyo, Japan*, 2017.

[79] W.-H. Lin and A. Hauptmann, "Structuring continuous video recordings of everyday life using time-constrained clustering," in *Multimedia Content Analysis, Management, and Retrieval 2006*, vol. 6073. International Society for Optics and Photonics, 2006, p. 60730D.

[80] Y.-L. Lin, V. Morariu, and W. Hsu, "Summarizing while recording: Context-based highlight detection for egocentric videos," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 51–59.

[81] Y. Liu, J. Fu, T. Mei, and C. W. Chen, "Storytelling of photo stream with bidirectional multi-thread recurrent neural network," *arXiv preprint arXiv:1606.00625*, 2016.

[82] Y. Liu, J. Fu, T. Mei, and C. W. Chen, "Let your photos talk: Generating narrative paragraph for photo stream via bidirectional attention recurrent neural networks," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[83] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *Computer Vision and Pattern Recognition*. IEEE, 2013, Conference Proceedings, pp. 2714–2721.

[84] Y.-F. Ma, X.-S. Hua, L. Lu, and H.-J. Zhan, "A generic framework of user attention model and its application in video summarization," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 907–919, 2005.

[85] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, 2017, pp. 1–10.

[86] S. Marvaniya, M. Damoder, V. Gopalakrishnan, K. N. Iyer, and K. Soni, "Real-time video summarization on mobile," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 176–180.

[87] K. Masumitsu and T. Echigo, "Video summarization using reinforcement learning in eigenspace," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2. IEEE, 2000, pp. 267–270.

[88] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[89] A. G. Money and H. Agius, "Video summarisation: A conceptual framework and survey of the state of the art," *Journal of Visual Communication and Image Representation*, vol. 19, no. 2, pp. 121–143, 2008.

[90] A. G. Money and H. Agius, "Analysing user physiological responses for affective video summarisation," *Displays*, vol. 30, no. 2, pp. 59–70, 2009.

[91] H. W. Ng, Y. Sawahata, and K. Aizawa, "Summarization of wearable videos using support vector machine," in *International Conference on Multimedia and Expo*, vol. 1. IEEE, 2002, Conference Proceedings, pp. 325–328.

[92] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang, "Video summarization and scene detection by graph modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 2, pp. 296–305, 2005.

[93] M. Okamoto and K. Yanai, "Summarization of egocentric moving videos for generating walking route guidance," *Image and Video Technology*, pp. 431–442, 2014.

[94] R. Panda and A. K. Roy-Chowdhury, "Collaborative summarization of topic-related videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[95] R. Panda, S. K. Kuanar, and A. S. Chowdhury, "Scalable video summarization using skeleton graph and random walk," in *International Conference on Pattern Recognition*. IEEE, 2014, Conference Proceedings, pp. 3481–3486.

[96] R. Panda, A. Das, Z. Wu, J. Ernst, and A. K. Roy-Chowdhury, "Weakly supervised summarization of web videos," in *ICCV*, 2017.

[97] W.-T. Peng, W.-T. Chu, C.-H. Chang, C.-N. Chou, W.-J. Huang, W.-Y. Chang, and Y.-P. Hung, "Editing by viewing: automatic home video summarization by viewing behavior analysis," *Multimedia, IEEE Transactions on*, vol. 13, no. 3, pp. 539–550, 2011.

[98] H. Pirsiavash and D. Ramanan, "Detecting activities of daily living in first-person camera views," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, Conference Proceedings, pp. 2847–2854.

[99] B. A. Plummer, M. Brown, and S. Lazebnik, "Enhancing video summarization via vision-language embedding," in *Computer Vision and Pattern Recognition*, vol. 2, 2017.

[100] Y. Poleg, C. Arora, and S. Peleg, "Temporal segmentation of egocentric videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, Conference Proceedings, pp. 2537–2544.

[101] Y. Poleg, T. Halperin, C. Arora, and S. Peleg, "Egosampling: Fast-forward and stereo for egocentric videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, Conference Proceedings, pp. 4768–4776.

[102] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, "Compact CNN for indexing egocentric videos," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–9.

[103] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," *Computer Vision–ECCV*, pp. 540–555, 2014.

[104] J. Ren, X. Shen, Z. Lin, R. Mech, and D. J. Foran, "Personalized image aesthetics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[105] G. Roig, X. Boix, R. De Nijs, S. Ramos, K. Kuhnlenz, and L. Van Gool, "Active map inference in crfs for efficient semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[106] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[107] Y. Sawahata and K. Aizawa, "Wearable imaging system for summarizing personal experiences," in *International Conference on Multimedia and Expo*. IEEE, 2003, Conference Proceedings, p. 45.

[108] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," in *European Conference on Computer Vision*. Springer, 2016, pp. 3–19.

[109] A. Sharghi, J. S. Laurel, and B. Gong, "Query-focused video summarization: Dataset, evaluation, and a memory network based approach," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 2127–2136.

[110] M. M. Silva, W. L. S. Ramos, J. P. K. Ferreira, M. F. M. Campos, and E. R. Nascimento, "Towards semantic fast-forward and stabilized egocentric videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 557–571.

[111] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems*, 2014.

[112] A. Singla, S. Tschiatschek, and A. Krause, "Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[113] M. Soleymani, "The quest for visual interest," in *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM, 2015, pp. 919–922.

[114] E. H. Spriggs, F. De La Torre, and M. Hebert, "Temporal segmentation and activity classification from first-person sensing," in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference On*. IEEE, 2009, pp. 17–24.

[115] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using LSTMs," in *International Conference on Machine Learning*, 2015, pp. 843–852.

[116] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," *Computer Vision–ECCV*, pp. 787–802, 2014.

[117] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[118] Y. Takahashi, N. Nitta, and N. Babaguchi, "User and device adaptation for sports video content," in *Multimedia and Expo*, 2007.

[119] E. Talavera, M. Dimiccoli, M. Bolanos, M. Aghaei, and P. Radeva, "R-clustering for egocentric video segmentation," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2015, pp. 327–336.

[120] E. Talavera, M. Dimiccoli, M. Bolaños, M. Aghaei, and P. Radeva, "R-clustering for egocentric video segmentation," *Pattern Recognition and Image Analysis*, pp. 327–336, 2015.

[121] C. Tan, H. Goh, V. Chandrasekhar, L. Li, and J.-H. Lim, "Understanding the nature of First-Person Videos: Characterization and classification using low-level features," in *Computer Vision and Pattern Recognition*. IEEE, 2014, Conference Proceedings, pp. 549–556.

[122] D. Tancharoen, T. Yamasaki, and K. Aizawa, "Practical experience recording and indexing of life log video," in *Proceedings of the 2nd ACM workshop on Continuous archival and retrieval of personal experiences*. ACM, 2005, Conference Proceedings, pp. 61–66.

[123] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[124] B. L. Tseng and J. R. Smith, "Hierarchical video summarization based on context clustering," in *ITCom 2003*. International Society for Optics and Photonics, 2003, pp. 14–25.

[125] P. Varini, G. Serra, and R. Cucchiara, "Personalized egocentric video summarization for cultural experience," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, Conference Proceedings, pp. 539–542.

[126] P. Varini, G. Serra, and R. Cucchiara, "Egocentric video summarization of cultural tour based on user preferences," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 931–934.

[127] A. B. Vasudevan, M. Gygli, A. Volokitin, and L. Van Gool, "Query-adaptive video summarization via quality-aware relevance estimation," in *Proceedings of the 2017 ACM on Multimedia Conference*, ser. MM '17. New York, NY, USA: ACM, 2017.

[128] X. Wang, Y.-G. Jiang, Z. Chai, Z. Gu, X. Du, and D. Wang, "Real-time summarization of user-generated videos based on semantic recognition," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 849–852.

[129] B. Xiong and K. Grauman, "Detecting snap points in egocentric video with a web photo prior," *Computer Vision–ECCV*, pp. 282–298, 2014.

[130] B. Xiong, G. Kim, and L. Sigal, "Storyline representation of egocentric videos with an applications to story-based search," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4525–4533.

[131] B. Xu, X. Wang, and Y. G. Jiang, "Fast summarization of user-generated videos using semantic, emotional and quality clues," *IEEE MultiMedia*, vol. PP, no. 99, pp. 1–1, 2016.

[132] J. Xu, L. Mukherjee, Y. Li, J. Warner, J. M. Rehg, and V. Singh, "Gaze-enabled egocentric video summarization via constrained submodular maximization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, Conference Proceedings, pp. 2235–2244.

[133] S. Yamamoto, T. Nishimura, Y. Akagi, Y. Takimoto, T. Inoue, and H. Toda, "Pbg at the ntcir-13 lifelog-2 lat, lsat, and lest tasks," *Proceedings of NTCIR-13, Tokyo, Japan*, 2017.

[134] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo, "Unsupervised extraction of video highlights via robust recurrent auto-encoders," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4633–4641.

[135] H. Yang, L. Chaisorn, Y. Zhao, S.-Y. Neo, and T.-S. Chua, "VideoQA: question answering on news video," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 632–641.

[136] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 982–990.

[137] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2282–2312, 2005.

[138] S. Yeung, A. Fathi, and L. Fei-Fei, "Videoset: Video summary evaluation through text," *arXiv preprint arXiv:1406.5824*, 2014.

[139] A. Yoshitaka and K. Sawada, "Personalized video summarization based on behavior of viewer," in *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. IEEE, 2012, pp. 661–667.

[140] L. Yu, M. Bansal, and T. Berg, "Hierarchically-attentive rnn for album summarization and storytelling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 966–971.

[141] J. M. Zacks, N. K. Speer, K. M. Swallow, and J. R. et al., "Event perception: a mind-brain perspective." *Psychological bulletin*, vol. 133, no. 2, p. 273, 2007.

[142] M. J. Zaki, W. Meira Jr, and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[143] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[144] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *European conference on computer vision*. Springer, Cham, 2016, pp. 766–782.

[145] B. Zhao and E. Xing, "Quasi real-time summarization for consumer videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, Conference Proceedings, pp. 2513–2520.

[146] B. Zhao, X. Li, and X. Lu, "Hierarchical recurrent neural network for video summarization," in *Proceedings of the 2017 ACM on Multimedia Conference*, ser. MM '17. New York, NY, USA: ACM, 2017, pp. 863–871. [Online]. Available: http://doi.acm.org/10.1145/3123266.3123328

[147] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems*, 2014, Conference Proceedings, pp. 487–495.

[148] L. Zhou, L. Piras, M. Riegler, G. Boato, D.-T. Dang-Nguyen, and C. Gurrin, "Organizer team at imageCLEFlifelog 2017: Baseline approaches for lifelog retrieval and summarization," in *Proc. CLEF 2017*, 2017.