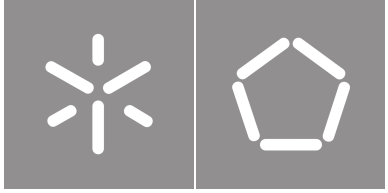
Universidade do Minho
Escola de Engenharia

Diogo Barroso Dias

Optimized video retrieval for interior vehicle monitoring

July, 2023



Universidade do Minho
Escola de Engenharia

Diogo Barroso Dias

Optimized video retrieval for interior vehicle monitoring

Master Thesis
Master in Masters in Informatics Engineering

Work developed under the supervision of:
João Miguel Lobo Fernandes
André Leite Ferreira
Sascha Lange

July, 2023

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.7.0) [18].

Acknowledgements

I would like to use this section to leave my gratitude to all the people that aided me in developing this dissertation.

To my university instructors André Ferreira and João M. Fernandes for the opportunity of enrolling in this academic internship and all the help they provided in writing this dissertation.

To my instructor at Bosch, Sascha Lange, for introducing me to an amazing work environment, helping me achieve the goals that were set out and improving my skills and abilities.

To Bosch Car Multimedia for accepting my internship, enabling me to peer into a new fascinating world, full of wondrous people.

To my mother, who throughout all that has happened in our life, kept pushing me forward into the right direction, serving as a anchor in my life.

To Helena Pinto, as she guided me through many of the difficult obstacles of preparing this dissertation.

To all my friends who never gave up on me and always believed in my capabilities, Pedro and José Parpot, Carlos, José Pedro, Pedro Mendes, Rui, and much more.

Obrigado, nunca serei capaz de retribuir a completo os momentos de felicidade que vivi por vossa causa.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

_____, _____

(Place) (Date)

_____

(Diogo Barroso Dias)

Abstract

Optimized video retrieval for interior vehicle monitoring

With the rapid growth in the amount of video data, an increasing need for efficient video retrieval systems has become an important problem in the multimedia management topic. Despite having a long past, the increase in file size of video collections, caused mostly by the increase of video resolution and quantity of videos, originated a big push for applying Machine Learning on the video retrieval subject. In today's world, when dealing with *Big Data*, it's unfeasible to still rely on video metadata and manually annotated videos to provide an accurate video retrieval engine, seeing as the sheer quantity of videos overwhelms an inept search and browse system, unable to provide the video the user wants. Therefore, by relying on machine algorithms to accurately mass tag the video collection we achieve great improvements. The process of allocating the video information to the video retrieval framework is severely less time-consuming and the viewer has at his disposal more precise and semantically accurate filters. This in turn, drastically reduces the quantity of redundant videos that are pulled from the user's queries. Another way to also ease the time it takes to analyze an immense quantity of videos, is by summarizing the content that is present on them. Condensing dozens of hours, pulled from one or more video streams, into a more accessible source of information that displays the most relevant data, is considerably a more efficient viewing experience for the user as it unburdens him of the task of surveying a grotesque amount of media content. The main focus of this thesis is to implement a video summarization method for recapping footage from the interior of a vehicle, that will be integrated on a video retrieval platform that is also being developed in parallel.

Keywords: *Video Retrieval, Video Summarization, Histogram, Disparity Minimization, Greedy Algorithm, Machine learning*

Resumo

Recolha de vídeo otimizada para monitorização do interior de veículos

Com o rápido crescimento na quantidade de videos, uma necessidade crescente de sistemas eficientes de recuperação de vídeo tornou-se num problema importante na gestão de multimédia. Apesar de ter um longo passado, o aumento do tamanho de ficheiro das coleções de vídeos, causado principalmente pelo aumento da resolução e quantidade de vídeos, originou um grande impulso na aplicação de aprendizagem automática na área de recuperação de vídeos. No mundo de hoje, ao lidar com *Big Data*, é inexequível ainda depender de metadados de vídeo e de vídeos anotados manualmente para fornecer um sistema de recuperação de vídeo preciso, visto que a grande quantidade de vídeos sobrecarrega um sistema de pesquisa e navegação inepto, incapaz de provisionar o vídeo que o utilizador deseja. Por conseguinte, ao basearmo-nos em algoritmos de máquina para etiquetar com precisão a coleção de vídeos, conseguimos grandes melhorias. O processo de atribuição da informação do vídeo à estrutura de recuperação de vídeo é muito menos moroso e o espetador tem à sua disposição filtros mais precisos e semanticamente exactos. Isto, por sua vez, reduz drasticamente a quantidade de vídeos redundantes que são retirados das consultas do utilizador. Outra forma de também diminuir o tempo de análise de uma imensa quantidade de vídeos, é resumir o conteúdo que está presente neles. Condensar dezenas de horas, extraídas de um ou mais fluxos de vídeo, em uma fonte de informação mais acessível que exibe os dados mais relevantes, é uma experiência de visualização consideravelmente mais eficiente para o usuário, pois alivia-o da tarefa de pesquisar uma quantidade imensa de conteúdo de média. O foco principal desta tese é implementar um método de sumarização de vídeo para recapitular gravações do interior de um veículo, que será integrado numa plataforma de recuperação de vídeo que também está a ser desenvolvida em paralelo.

Palavras-chave: *Recuperação de Vídeo, Sumarização de Vídeo, Histograma, Minimização de Disparidades, Algorítmo Ganancioso, Aprendizagem Automática*

Contents

List of Figures

List of Tables

Acronyms

CNN Convolutional Neural Networks 28, 33

DSNet Detect-to-Summarize Network 28, 30, 31, 33

FN False Negative 27

FP False Positive 27

RNN Recurrent Neural Networks 28, 33

TN True Negative 27

TP True Positive 27

1 Introduction

1.1 Context

In today's world it's almost unthinkable for any company that's looking to thrive in this technology driven society to not be familiar with the term "Big Data". Presumed to be coined by a group of computer science and statistics experts figured by John Mashley [7], Big Data was pronounced as "an emerging technology", of such importance that it would revolutionize the decision making of companies. Defined as very large sets of data that are produced by people using the internet, and that can only be stored, understood, and used with the help of special tools and methods, Big Data became a prevalent subject of study of corporations, scholars and organizations who have an important tool for unlocking the potential of Big Data in Artificial Intelligence. Described as "the capability of a device to perform functions that are normally associated with human intelligence, such as reasoning, learning and self-improvement" [32], Artificial Intelligence is a powerful application best harnessed when dealing with exponentially time-consuming processes too impractical to be fulfilled by humans. Therefore, its only expected for Big Data and AI to have a synergistic relationship. From this association several activities grouped around the capture, storage and consumption of big data can be complemented by the introduction of AI, being video retrieval of major focus. The increasing need for a video retrieval system capable of dealing with large amounts of video while still providing an accurate search and powerful browsing system led to the use of machine learning algorithms in this subject, where several techniques can be applied to aid the user in retrieving the information he deems most relevant. However, it still has a lot of room to improve, despite the positive developments it has already achieved. Machine learning experts are always trying to improve their models, trying to squeeze the best performance possible, maintaining an accurate but still efficient model. Ergo, a necessity surges in provisioning a framework able to deal with an extensive database, where the expert can evaluate the authenticity of its models with the aid of powerful, semantic focused techniques that aim to clarify the content presented in the data by presenting it in a simple to understand state, instead of the machine learning developer trying to extrapolate information from toneless, raw data, devoid of any human meaning. One project that's actively being explored at Bosch Car Multimedia SA, is a video retriever platform that aims to present to the viewer, summarized information from a dataset of video collections, created from

the monitoring of the interior of different vehicles, in several situations. The main goal is to use features extracted from the video stream, with the purpose of better allocating the video in accurate categories based on its content, deliver an efficient process of querying on databases and a faster, easier and more accurate way to filter data based on the desired features. The aforementioned advantages of video retrieval are significantly important on the topic of deep learning algorithms, as the most fundamental prerequisite in creating a model with good performance, is to use suitable, precise and well-adjusted datasets for the model's training and validation. In this thesis, the task of video summarization is going to be explored, in regard to a vehicle monitoring project, and then, its subsequently integration into a video retriever platform being developed in parallel by Bosch Car Multimedia SA.

1.2 Objectives and expected results

The main goal of this dissertation is to implement a video summarization method for summarizing the content of several hours of footage of a interior of a car, displaying the most relevant events during the in-vehicle monitoring. The first objective is to study the different state-of-the-art approaches in video/image retrieval, that is, to investigate the technical approaches for processing videos and extracting features for video retrieval, comparing the methodology, advantages and disadvantages the different approaches have. Then, a video summarization technique able to answer the problem presented is devised.

Having deliberated upon the approach we will use, we intend to devise a video summarization technique that is able to answer the problems laid out by Bosch. However, having constraints in regards to computational power available and the prerequisite of processing the user's request in a short time, we need to implement a lightweight and fast approach, optimized to deliver an accurate but efficient application.

Finally, it is expected that the solution stands as a reliable video summarization method, with the purpose of integrating a video retrieval framework being developed.

1.3 Document structure

The remaining material in this document is divided into the following major sections:

- Chapter 2 describes the state of art reviewed in the area of video summarization. In this chapter, a definition of video retrieval will firstly be presented, follow by the different types of Queries used. A research of video segmentation and video summarization are also presented in this chapter. As for the final part, a review of some studies in the area of Video Summarization is given.

- Chapter 3 comprises two primary segments that detail the product of this dissertation. The first segment provides an elaborate account of the procedures involved in executing the suggested solution. The second section within this chapter is dedicated to analyzing the outcome obtained from

implementing our method, with the aim of studying the performance when compared against a state of the art method, and if it reached the goals proposed.

- Chapter 4 is the final chapter of the dissertation, made up of Conclusion and Future Work, that gives an overview of the project, giving a short, succinct explanation of how our approach works and its results. After, we elaborate on possible improvements that can be implemented in our solution.

2

State of the Art

2.1 Video Retrieval

Video Retrieval is a framework for provisioning a collection of videos through the functionality of a search and browse system. The main objective follows along these lines:

1. A text query is given by the user
2. The video retrieval system returns a group of videos as candidates sorted by the relevance to the query
3. The user selects the video with the most significance to him

The main components of the video retrieval process are video segmentation, low-level feature and high-level concept extraction, indexing by using low level features and concepts, and the browsing and retrieval system.

Despite the various achievements in this branch of information retrieval, the challenge of bridging the semantic gap is still a prominent one. Even though the computational power present in today's machine has remarkably risen, the act of converting high level queries posed by humans to a language machines can understand is still a major problem, even though low level features can be easily measured and computed. According to Lew *et. al* [14] the essence of a semantic query is understanding the meaning behind the query. This can involve understanding both the intellectual and emotional sides of the human, not merely the distilled logical portion of the query but also the personal preferences and emotional subtones of the query and the preferential form of the results. Another challenge video retrieval seeks to answer is to provide a well structured and fast framework when dealing with a vast collection of videos.

Mohamed et al. [19] specify the steps taken in the video retrieval process, as show in Figure 2.1: firstly, the video is segmented, decomposed into several segments, shots, frames...; secondly, low level features are extracted from the video stream; afterwards, concepts and semantics are pulled from the video; both of these types of features, low and high-level are used in the indexing procedure, tagging the video with the correct label; finally, the last process takes as input, a query from the user, who requests a

video with a specific feature filter, which then the video retrieval system, returns the video or collection of videos that satisfy the user's query, sorted by ranked similarity of the user query.

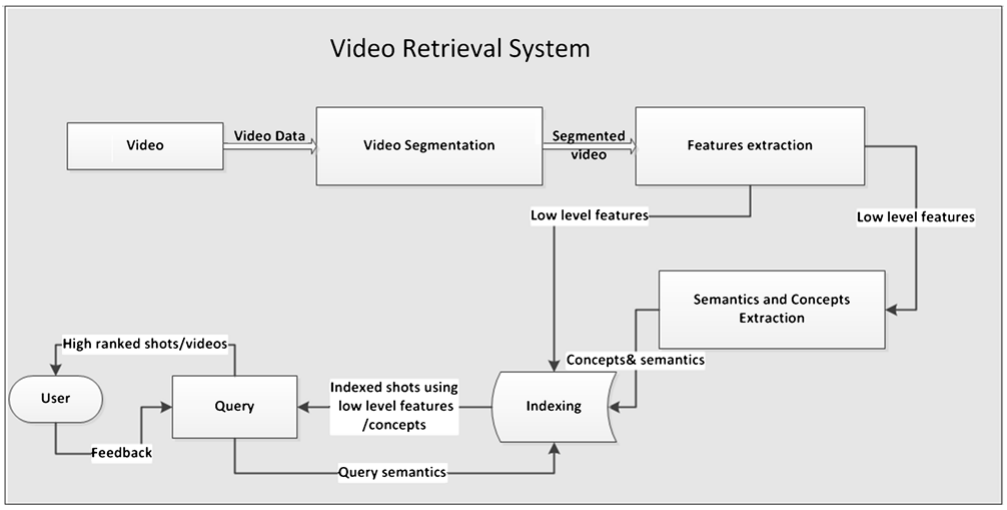


Figure 1: Basic Video Retrieval framework from [19].

2.1.1 Query

The goal of video retrieval is to retrieve the most relevant video(s) given a user's request, therefore making the query an important element of video retrieval as they answer the user's need for a specific video document, whether it is the entire video, a scene, a shot or a keyframe. One of the major benefits video queries have in regard to classic text document queries, is the variety of inquiry methods the search engine provides to the user as a consequence to the several types of information that can be extracted from a video. Snoek and Worring [26] define the following types of query:

- **Query by Keyword:** Being the most simple and popular method, text retrieval is a staple in most video retrieval systems due to the ease of implementation and being the most straightforward method for the user to operate. In this approach several pieces of information are extracted from the video itself, technical attributes (such as file dimensions, file type), bibliographic attributes (such as title, description, tags), structural attributes (such as sections, video chapters)... Defined as metadata they generate keywords from its content providing the user an accurate retrieval method based on keyword matching [book refs], the act of corresponding user inserted keywords with the keywords extracted from the video. One of the major downsizes of this query type in video retrieval is their incapacity in obtaining data from the visual content of the video being therefore unable to provide the user query-able keywords from perhaps the most important piece of information.

- **Query by Example:** In contrast to text retrieval methods, query by example follows a visual-only procedure to match the user query, being the main difference with the previous method the fact that the query doesn't rely on keywords extracted from the video metadata. Instead, the user has the

disposal to use images or videos as a depiction of the content that's being searched for, and through the extraction of low-level features from the sampled images/videos, the system queries its own collection of videos for one or more video documents that display the higher level of similarity with the user query, subsequently returning either the exact video or the more similar ones. While this is a very effective method for retrieving almost identical videos, a user typically searches for kindred semantic content, not just videos that have similar visual descriptors like color, texture, shape, etc. For this problem, B. Wang et al. [30] proposed "an intelligent semantic video retrieval approach that transforms visual images into conceptual keywords by effective semantic image annotation". But the query by example paradigm isn't only restricted to the sampling of general images/videos.

- **Query by Concept:** Instead of relying on the system capacity to match the user keyword input with a video, a list of concepts is devised with the intent of covering all possible subjects that could be present in the video collection. Thus, the user can query the database by selecting one or more concepts from the aforementioned developed index, obtaining a list with all the videos that pass the concept criteria. Though this procedure might be more reliable than querying by keyword or by example, there's a risk in overwhelming the user since an enormous quantity of concepts can be difficult to memorize or to sift through.

Though these types of queries are still relevant, additional techniques have been developed that supplement the existing ones to achieve a more precise search.

Hu *et al.* [10] devised a Query by Sketch method that permits the user to search for his desired video by sketching the motion trajectories expected in the activities he is querying for. The sketch features are extracted and then matched with the paths extracted from the videos.

When dealing with Query by Concept, Otto, Springstein and Veith [22] present a novel algorithm for visual concept detection, similarity search, face detection, face recognition and face clustering, all combined in a multimedia tool for video inspection and retrieval, supplemented with different procedures for querying. With a total of 58 concepts, they divide them into different categories: "who", "what, "where, "when" and "how" with the purpose of covering the most global aspects of a video content, overall developing a more effortless search system for the user to understand. Moreover, a query paradigm for searching a person is also implemented with the purpose of retrieving a specific person.

2.2 Video Segmentation

Video Segmentation is defined as the process of decomposition of the video into units of frames that follow a specific homogeneous criteria. They can be divided into shots, stories, scenes, subshots and keyframes. A shot matches an continuous video clip. A story corresponds to a group of several scenes that record an ongoing series of events. A scene is defined as an event, that is composed of a number of neighbouring shots that are semantically related. A shot stands for a sequence of temporally adjacent

frames. A keyframe is a single frame representation of a shot. M. Wang goes into detail of the video units in [29], illustrated in Figure 2:

- **Shot**: uninterrupted clip recorded by a single camera. Often forms the building block of video content.
- **Scene**: collection of semantically related and temporally adjacent shots, depicting and conveying a high-level concept. A scene usually comprises a series of consecutive shots that are recorded in the same location.
- **Story**: clip that captures a continuous action or a series of events and it may be composed of several scenes and shots.
- **Subshot**: segment within a shot that corresponds to a unique camera motion. A shot can be divided into one or more consecutive subshots according to the movement of the camera.
- **Keyframe**: frame which best represents the content of a shot or a subshot. According to variation of the content, one or more keyframes can be extracted from each shot or subshot. Keyframes can be used as the entries of the video data for manipulations, such as indexing and browsing. Therefore, video content structuring may involve the following five techniques: shot detection, scene grouping, story identification, subshot segmentation, and keyframe extraction.

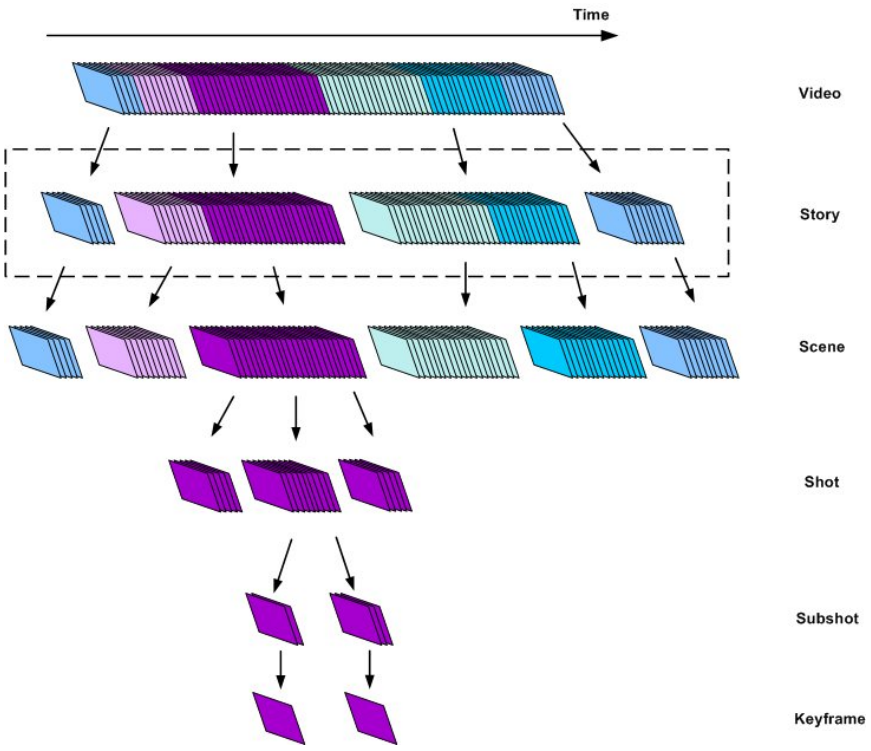


Figure 2: Hierarchical decomposition and representation of video content adapted from [29].

2.3 Video Summarization

Video summary can be defined as a sequence of still or moving pictures (with or without audio) presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original video is preserved [25].

The video summarization process has three steps [12]:

- Examine video information with the purpose of discovering the most relevant structure or methods within the visual, audio and textual components (audio and textual if they're present);
- Select the keyframes which represent the content of the video;
- Produce a synthesis/summary by organizing the frames/shots/scenes selected from the original video.

According to Troung and Venkatesh [28], there are two basic forms of video abstract: *Static Video Summary* or *Keyframes*, also called *representative frames, R-frames, still-image abstracts* or *static storyboard*. Consists of a collection of salient images extracted from the underlying video source. *Dynamic Video Skimming*, also called a *moving-image abstract, moving storyboard* or *summary sequence*. Consists of a collection of video segments extracted from the original video. These segments are joined by either a cut or a gradual effect (e.g., fade, dissolve, wipe). It is itself a video clip, but of significantly shorter duration (for example, a movie trailer).

Kini *et al.* [12] made the present Table 1, comparing the two forms of video summarization:

Keyframe based summarization	**Video Skimming**
Not restricted by synchronization issues	Able to include audio and motion elements that potentially enhance both the expressiveness and the amount of information conveyed by the summary
More flexibility in terms of organization for browsing and navigation purposes	Contains motion information
Able to grasp the video content more quickly	High User viewing experience
Reduces computation complexity for various video analysis and retrieval applications	Time restricted and synchronization is necessary
Only video frames considered	Video, audio and text data are considered
Helps in video indexing and searching	Helps in video indexing and searching but performance is inferior than keyframe based summarization

Table 1: Comparison between keyframe based summarization and video skimming adapted from [12].

Money and Agius [20] discuss two other types of methods that can summarize the video stream, as shown in Fig 3 :

- **Textual**, text descriptors are used to summarize the content of the video, normally by a combination of concept detection and the study of low level image features like color, edges, blobs, patterns, that can be picked up with the use of histograms or more advanced descriptors, like *HOG*, Histogram of oriented gradients, that operates by counting occurrences of gradient orientations in localized portions of an image, normally used for object detection because of the invariance to change in light deformations [6]; and *SIFT* (Scale Invariant Feature Transform) [21], a feature detection algorithm with the purpose of locating the keypoints, local features, of an image/frame that are rotation and scale invariant. It is distinguishable from *HOG* features through the fact that it is impervious to size or orientation change.

- **Graphical**, which can supplement existing video summaries or substitute them, by presenting an additional level of detail that traditionally isn't achievable via other methods, through the display of a graphical overview of video content extracted summary. R. Lienhart *et al.* [16] proposed the graphical interface *MoCA*, Movie Content Analysis System, that lays out a two-dimensional colour coded block map of the video highlighting different events, such as dialogues, explosions, text appearances, and so on, that can be shown at Fig 4. The graph shows the temporal distribution of the detected video and audio events of a movie as well as those which have been chosen during the clips selection process to be part of the trailer. Each box represents two seconds (2828 in total). Time passes from left to right and top to bottom.

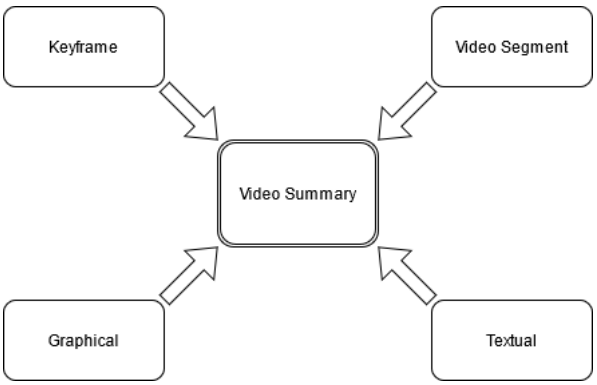


Figure 3: Visual cues used for Video Summaries

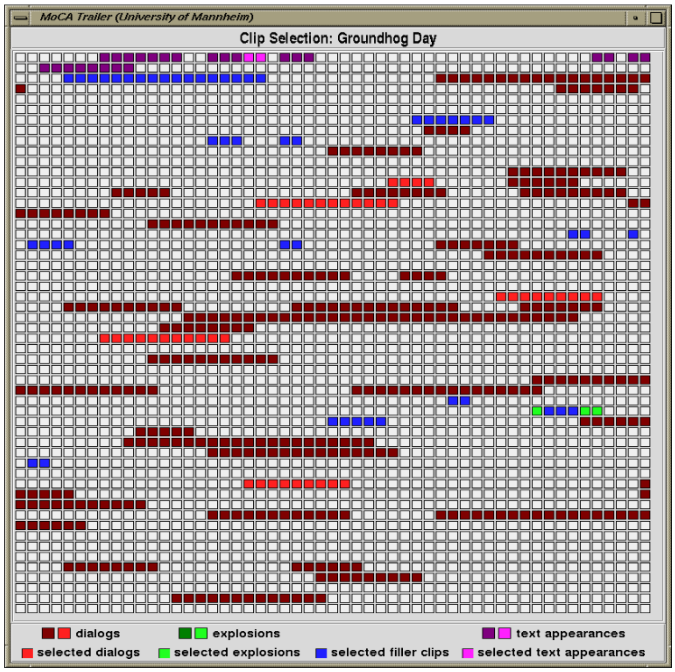


Figure 4: MoCA Interface.

Video summarization techniques can be breakdown into two types, depending on the number of input videos: *Single-Video Summarization*, where a summary is extracted from a single video [insert], and *Multi-Video Summarization*, a summary is extracted from a collection of videos [reference]. On the other hand, taking into account they type of information that is being summarized, A. Money and H. Agius [20] describe three methods for summarizing a video:

- **Internal:** A summary is created by analyzing information taken directly from the input video stream.
- **External:** Information not obtained directly from the video stream is analyzed (e.g. audio, emotion, video capture time);
- **Hybrid:** Procedure where both internal and external information are combined with the purpose of being analyzed;

All of these three techniques follow a specific type of study, *domain specific* or *non-domain* specific analysis. Domain specific refers to a methodology that focuses on the content of the video being summarized (e.g. news, sport, music, movie, surveillance). The objective is to reduce the vagueness of the video summary by applying prior knowledge of the domain during the analysis procedure, that can help reduce the ambiguity traditionally associated with extrapolating meaning from low level features. Therefore, internal video summarization benefits greatly from domain-specific analysis, although it can also improve the other two methods of summary. On the other hand, non-domain specific analysis answers the summarization problem with a more generic approach, offering a solution for any type of content domain.

Continuing on this subject, internal, external and hybrid summaries can be classified according to the type of content included on the summary, in accordance with the criteria used for their analysis. Not only that, they can also be distinguished by their degree of interactivity and personalization.

Shown at Figure 5, we can distinguish between internal and external video summarization, the different types of content in video summaries, and the summary type.

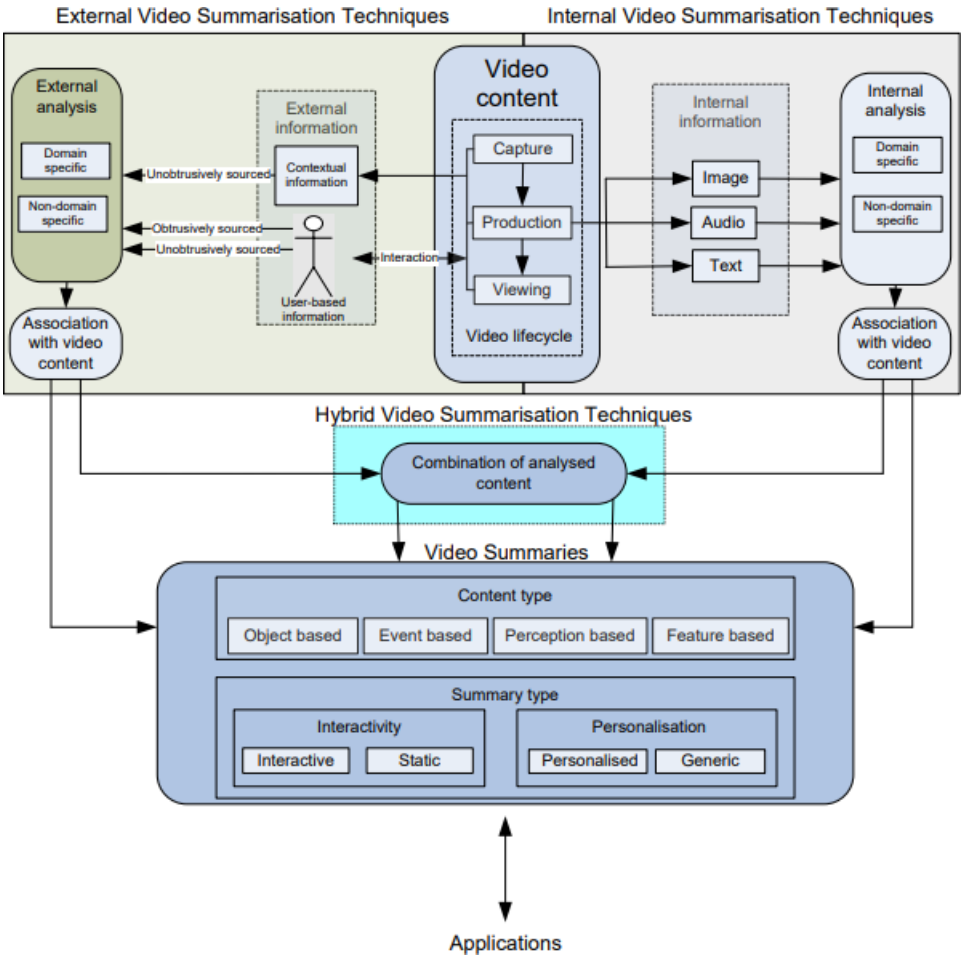


Figure 5: Conceptual framework for video summarization from [20]

.

2.3.1 Keyframe Based Summarization

U. Gawande et al. [8] define keyframe as a representative frame of a video which includes the whole of the video collection. Used for indexing, classification, evaluation, and retrieval of video, keyframe is an important component of computer vision algorithm, they summarize the visual content of a video segment. When extracting keyframes of a video, machine learning approaches base their selection process on selecting frame features and then using them as a basis for measuring inter-frame similarity. Similar frames are normally grouped by two principles, temporal proximity or label similarity, pertaining to some kind of distance metric. From this group, a single frame is selected as the feature frame, on the grounds

that it was proven to be the frame that contained the most information about the video segment it was pulled from.

The keyframe selection procedure, show at Figure 6 can normally be defined as the following: first, a target video is retrieved; secondly, frames are extracted from the video stream depending on two possible techniques, a simple frame sampling where an temporal interval is given and frames pertaining to that interval are extracted, or they are pulled from previously detected shots; afterwards, features are extracted from the frames and computed by one or more feature extraction algorithms; with this new source of information extrapolated from the selected frames, a clustering algorithm is run, grouping similar frames, and then, finally, selecting the frame that best represents the group it was clustered into.

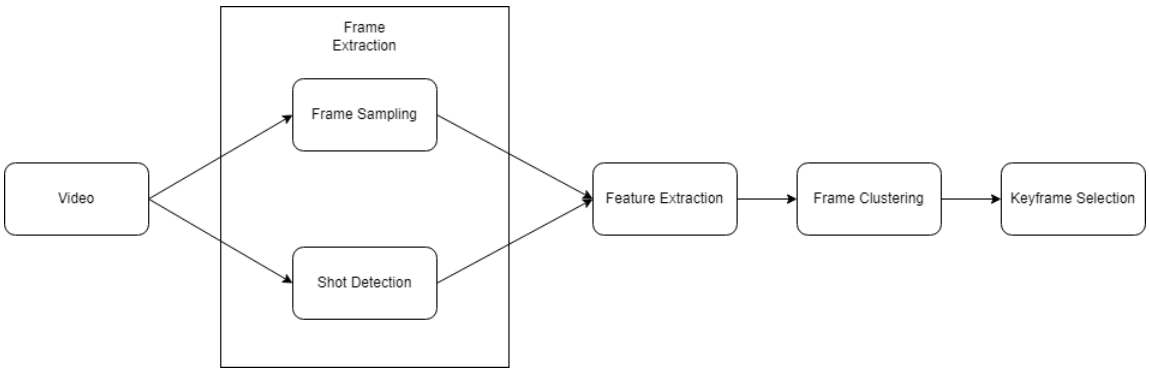


Figure 6: Keyframe Selection

2.3.2 Keyframe Extraction Methods

The norm for extracting keyframes from shots is defined by the use of measure, distance metrics or frame sampling at a fixed rate. Jadon and Jasim [11] talk about four keyframe extraction techniques:

- **Uniform Sampling**, a fairly common method [23], where every *n*th frame is pulled from the video, *n* being defined by the length of the video. A common baseline for selecting the duration of the summarized video is between the range of 5% to 15% of the original video.

- **Image histogram** [31], a graphical representation of the image distribution, containing valuable information about the image/frame, that counts and divides the number of pixels according to their brightness, value between the range of 0 and 256. The procedure to keyframe extraction using histograms is normally divided into the tasks of extracting histograms from all the frames, and then, running a comparison algorithm that determines whether or not, a pair of frames have significant dissimilarities between them. In consecutive frames, where the percentage of dissimilarity is high, it can be assumed there's significant difference between the two frames, therefore concluding each of the can be selected as candidate keyframes since they both hold different visual components that may be of interest.

- **Scale Invariant Feature Transform (SIFT)** [21], as mentioned before, a feature detection algorithm used for several computer vision applications, can be used for keyframe extraction. In SIFT, important locations are first defined using a scale space of smoothed and resized images, afterwards, Gaussian functions are applied on the images with the purpose of finding the minimum and maximum responses. Non maxima suppression is performed and computative matches are discarded to ensure a collection of highly interesting and distinct collection of keypoints. Then, a HOG is performed, dividing the image into patches to discover the dominant orientation of the localized keypoints, which are extracted as local features.

- **VSUMM**, proposed by S. Avila *et al.* [3], this technique uses the K-means method to cluster features extracted from each frame, to calculate the similarity among video frames. In this approach, the video isn't split into shots. Instead, frame clusters are obtained by video frame analysis, independently of the shot the frame belongs to.

2.3.3 Benchmark

Truong and Venkatesh[28] describe three methodologies for evaluating video summaries. Result description [33] is perhaps the most frequently used form of evaluation, probably because it doesn't compare results with other techniques. Objective Metrics [17], commonly, the fidelity function originated from the computation of the original frame sequence and the automatic summary. Unlike result description, objective metrics are used to compare automatic summaries created by different video summarization methods. Thirdly, User Studies [13]. Instead of relying on computational power to evaluate computer made summaries, in User studies, independent users rate the quality of automatic video summaries. Although it may be the best technique for judging video summaries, it's the most hard to set up, due to the complexity of organizing the studies.

One popular method for evaluating machine made video summaries, was proposed by S. Avila [3], denominated Comparison of User Summaries (CUS). As the name implies, it compares automatic summaries with man made summaries, which serves as ground truth, as shown on Figure 7, which compares user generated summaries, to VSUMM and the papers approach, GVSUM [4].

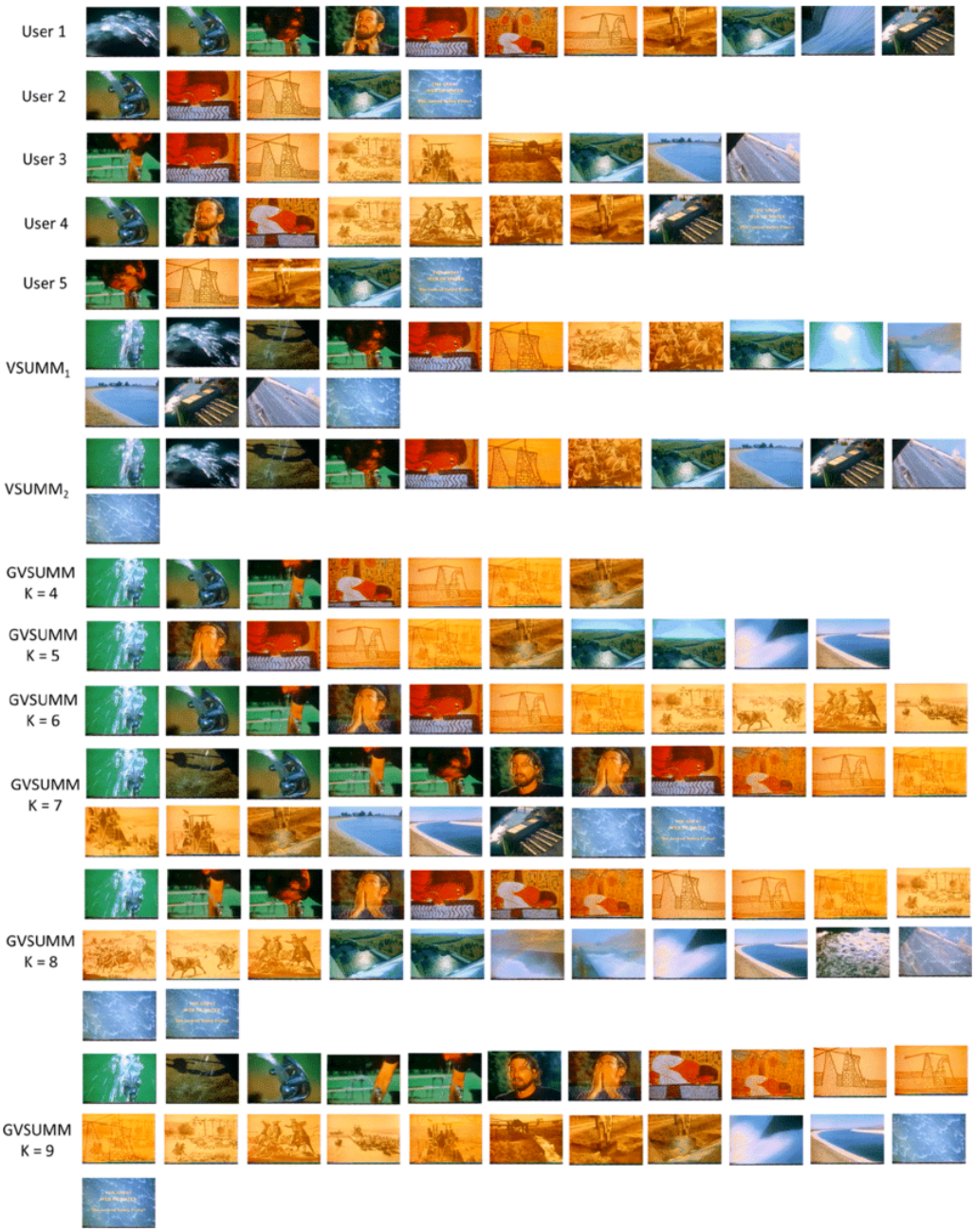


Figure 7: Comparison of several video summaries adapted from [4]

Concluding, evaluation of video summaries is always gonna be a challenging and subjective task, due to each user personal bias, the type of content being summarized and the length desired.

2.4 Related Literature

Video Summarization is a popular field due to its crucial role in the video analytic problem, having an abundance of research papers submitted on the topic. This section presents an in-depth review of some

studies where researchers pushed the boundaries on state of the art video summarization, but also novel ideas that introduced new frameworks, techniques more optimized for out of the ordinary problems.

S. Zhang, et al. proposed CAVS, Context-Aware Surveillance Video Summarization, in [34], a framework built to capture the most important video segments through data about individual local motion regions, including the interactions that affect these motion regions. Adopting a methodology of sparse coding with generalized sparse group lasso that learns a dictionary of video features and spatiotemporal feature correlation graphs, they are able to prove CAVS effectiveness in finding new events or different event correlations in surveillance videos and a small amount of other online videos.

M. Otani, et al. presented a text-based video summarization method that summarized the video content taking into attention a textual input, in [24], with the objective of improving quality of life of bloggers . Since the video editing process takes time and require a basis of knowledge to correctly edit a video, the authors studied the possibility of using video summarization to automate this process. They proposed a framework that took into consideration the author's intentions, with the purpose of achieving a video summary that reflected those intentions, where supporting text served as a prior to the video summarization. This techniques uses high-level features, in particular, objects and words, employing a content-similarity measure between the two, so then it can select the video segments where the objects described in the text were present in the visual content. Therefore different summaries can be produced from the same input of video(s), depending on the input text.

G. Liang et al. [15] developed a dual-path attentive network to settle the parallelization difficulty that similar recurrent structure models have due to it's step-by-step characteristic. The proposed model incorporates a temporal spatial encoder, score-aware encoder and decoder, with the purpose of capturing temporal and spatial information, in the first case, and to integrate appearance features with previously predicted importance scores, in the latter case. This approach, when comparing with other RNN methods, has the ability to be trained in full parallel leading to a significant performance improvement and the capacity to scale with larger databases.

3

Approach for In-vehicle Monitorization Video Summarization

3.1 Problem Definition

The proposed thesis was created with the purpose of developing a solution for a problem machine learning developers at Bosch have been encountering. A project being worked on involves the monitorization of the vehicle's interior so that the footage can be reviewed by machine learning developers to construct models that infer on several different scenes utilizing specific footage to achieve relevant results. Therefore, Bosch hired several actors and set up specific conditions to reenact from everyday scenes to more exceptional ones that occur inside the interior of the vehicle, with a higher-than-average focus on acting out different violent events. A problem arises, though, related to the several hundred hours of footage that are labelled incorrectly or are not labelled at all. It just is not feasible to watch such a high quantity of videos trying to cherry-pick the data that is more appropriate for the model in development. As such, this dissertation aims to help machine learning developers find specific data without wasting time finding videos when that time could be spent instead on improving machine learning models.

This project's dataset consists of a set of activities performed by actors inside a shared vehicle service, classified as 'Violent' or 'Non-Violent' activities. 'Violent activities' are considered harmful and dangerous to persons, involving physical violence and sexual assault. 'Non-Violent' activities represent ordinary activities that can occur during a ride without causing harm or disturbance to others.

Some efforts have been made to label the videos according to their content, however, the time needed to annotate them was seen as too costly, as sometimes the duration of a video can break the one-hour mark, plus when paired with the hundreds of recordings that need to be tagged, made the task of labelling the dataset too arduous to be achieved. While it may be the most accurate method, it simply is not feasible, as noted by the fact that only a tiny portion of the dataset is labelled. Furthermore, even with the possibility of working with a fully labelled dataset, an ongoing effort is needed to tag videos that continuously keep being added to the dataset. Consequently, the method in development needs to display the most noteworthy events of a video as quickly as possible while also being future-proof, in account of

an obligatory requirement to provide the user with the capability to display such information on the fly. Suppose the user wants to check the content of the video without actually seeing the full-length recording. In that case, the method should promptly display out of ordinary segments to the user instead of waiting for a person to label the video manually.

A good summary should pursue three crucial goals:

- **Interestingness:** Unique events ought to have the utmost importance, as they hold the most appeal among every other video segment, making it a priority to be included in the summary. For example, a moment of violence inside the vehicle would be essential in constructing a good summary.
- **Diversity:** It should maintain the diversity of the original video while removing redundancy. Even though a video might possess a particular unrivalled noteworthy sequence, the summary would not do justice to the video if it disregarded every other unique sequence, however mundane it might be.
- **Representativeness:** Amongst candidate video segments or frames to be appended to the summary, the most representative ones should be chosen.

Videos can be summarized into many different representations: keyframes, skims, storyboards, timelapses or video synopses. In this thesis, skims were ultimately selected after evaluating what representation holds the most potential for showing the video's content efficiently and quickly.

Keyframes are representative frames extracted from the video, and though it may be the fastest method for summarizing, it definitely pays for it. They completely remove any information regarding motion, actions and transitions, which are important for understanding the content of the video, obviously lacking any type of temporal context and possibly missing out on essential information that occurs in non keyframe frames. They also lack coherence and have low storytelling capabilities, they fail to capture the logical sequence of events of the video making it difficult to understand the overall story of the recording.

Skims, or **dynamic video summaries**, output a condensed version of the original video, retaining motion information, which creates a more pleasant viewing experience. They're perfect for longer videos or videos with extended perios of inactivity or repetitive content. By preserving motion, actions and transitions that may exist in the recording, it captures its temporal context and narrative.

Storyboard, or **static video summary**, represents a video sequence in a static imagery form, one or more selected representative frames from the original video, or a synthesized image generated from the selected keyframes. A set of keyframes are extracted from shots of the original video, arranged or blended in a two-dimensional space [1]. While this method might present the content of the video in an easy-to-see manner, it starts facing problems when there is a high quantity of scenes that need to be present in the summary, as it only really has two ways to do this: one, create a storyboard filled with dozens of keyframes that in turn creates an unpleasant experience for the user, it is not practical in comparison to other video summarization types of representation, to spend a great deal of time and focus to gain a better understanding of the original content from several hundreds of pictures; two, instead of an abnormal

amount of keyframes being used, the most relevant ones can be chosen as to provide a more captivating experience, however, with this a lot of temporal and motion information is omitted from the output, making it impossible for a person to interpolate events of the video with such a low number of images.

Time-lapses are normally used to capture natural phenomena and to provide artistic cinematic effects, and in the case of video summarization, it is more popular with surveillance videos, as time-lapses are an effective tool for visualizing motions and processes that evolve too slowly to be perceived in real-time [5]. Therefore it is not well suited for the domain of the provided Bosch dataset, as the focus is to infer upon violent and everyday human actions, which are short motions by default and can be recognized instantly. One crucial factor is also the person's place in the video, which in normal circumstances occupies a specific position in the video, the car seat, rarely getting out of bonds of this box, which confronts standard time-lapse display methods. Having a cluttered output video defeats the purpose of video summarization, displaying in a clear way events of the video.

Video synopses are unique in the way that it is not only composed of frames from the original video. Normally, a algorithm is used for detection and tracking of moving objects, which are then displayed in short clip. This short clip displays all objects and events that were detected at different times, simultaneously, meaning that it stitches multiple frames together. Whilst this method of summarization may be of great use for closed circuit television footage, where there isn't really a risk of events overlapping each other, if we apply it to this thesis domain, the interior of the car, we quickly find ourselves with a video too cluttered to be able to make sense of. It fails on domains where actions, objects and persons take up a large portion of the screen.

3.2 Dataset

In order to test and evaluate the different summarization algorithms, a pool of videos were selected and stored, comprised of several different types of scene reenactment, with the purpose to create a lightweight but accurate dataset able to represent the diversity of the recordings Bosch possesses. In the beginning phase of the project:

- All chosen videos had the duration between one and five minutes.
- They were recorded in monochrome.
- No sound was present.

As later explained, in the initial testing of the algorithms, we were able to see positive results , reinforcing the idea that submodular summarization models can be applied to the Bosch Dataset.

As the work continued and progress was made in relation to the summarization algorithm a new dataset was introduced to help design a better solution for the problem proposed in the thesis. As a main focus, the new dataset is composed with a larger number of vehicle interior violence recordings that incorporate different lengths, color palettes and type of events.

3.3 Work Process

On the video summarization pipeline which is the process from which the video goes in its raw state to its summarized version ready to be inferred upon by the viewer, there exists multiple steps the video goes through:

- **Preprocessing** - To extract image features from the video frames and reduce the computational cost of subsequent stages, which is done by resizing all frames to an uniform size.
- **Feature Extraction** - Before beginning to extrapolate between the keyframes, first it is needed to extract features from every frame in the video, in order to be possible to make a comparison between them.
- **Analysis**:
  - **Scoring** - After extracting the image features, the scoring procedure begins. A value is assigned to each frame depending on its level of similarity or interest, meaning, how much that specific frame is likely to be of great importance for the viewer.
  - **Disparity Minimization** - Having a myriad of possible summaries, we employ a method for reducing them to a uniform distribution, ensuring we have a good balance between achieving a final summary with the highest possible value and obeying through the time constrainst imposed.
  - **Evaluation** - After the scores are assigned, a submodular algorithm is introduced in order to select a diverse shot subset that best encapsulates the video's highlights and adheres to the time constraint introduced by the user, which is the desired summarized video length.
- **Output** - The final summary video is then created by concatenating the chosen shots.

3.3.1 Preprocessing

Frames are downsized to a predefined resolution in order to facilitate the histogram similarity comparison and reduce the computational cost of extracting features. An array of snippets is also created, by dividing the raw video in shots of 3 seconds, effectively distinguishing each temporal segment of the video, providing the system with unique shots to differentiate between.

3.3.2 Feature Extraction

Main purpose of this phase is to calculate histograms for all frames, and then determine the mean of histograms for each snippet. First, having opened the video file using *OpenCV's* video capture function and looping through each segment that with their frames resized, we start calculating the histogram of said resized frames. Depending on whether the segment holds one or more frames, the histogram is

calculated for both but with certain differences. When the segment only consists of one frame, the color histogram of the frame is calculated and is appended to an array, *snippetHist*. If the segment consists of multiple frames, the function loops through each frame in the segment and extracts them to a temporary array. It then calculates the average color histogram of all frames in the segment and adds it to the end of *snippetHist*. It also adds a cost value to a separate array, *costList*, based on the length of the segment.

The two functions we created for the purpose of calculating histograms of the video frames use *OpenCV's* histogram function *calcHist()*, that calculates the frequency distribution of pixel intensities of the frame. One function takes a list of frames and loops through each one in the segment, calculating their histogram using *cv2.calcHist()* and then normalizing them using *cv2.normalize*, both functions being part of *OpenCv's* class. After looping through all frames it calculates the average of histograms, and normalizes said histogram again. The function colorHist() takes a frame/image and calculates it's color histogram using the function *cv2.calcHist()* and then using *cv2.normalize()* to normalize the histogram. The reason why we normalize histograms is to ensure that they aren't biased by variations in the overall intensity or color distribution of the frames. Frames with higher overall pixel intensity dominate histograms representation, which brings about the potential of alienating the summarization algorithm to favor those frames. The same can be said about color channels, where is also the possibility of a single channel overpowering the others, since they may have a different distribution of pixel values. However, if we normalize the histograms, meaning, we developed a standardized representation, we significantly remove potential biases and guarantee our summarization technique isn't dangerously influenced by global variations in the intensity or color distribution of the video's frames.

3.3.3 Analysis Phase

The analysis phase of our approach is critical in extrapolating important information and identifying interesting frames from the input video. We use various techniques and algorithms to analyze the content of the video, measure the similarity between frames and determine the significance, level of interestingness of each frame in the context of the original video. We begin by employing the *computeKernel()* function, that plays a fundamental role in capturing the relations frames have between each other and establishing a measure of similarity. After, we also evaluate the gains of adding or removing frames to the final summary, by employing the disparity minimization technique to identify diverse and interesting frames, and applying a submodular algorithm, a greedy optimization algorithm, to establish a level of priority on segments based on their importance, to ultimately be decided upon whether or not they should be included in the summary.

3.3.3.1 Similarity scoring

The *computeKernel* function takes two optional parameters: gamma and compare method. It computes the similarity between a set of histograms using different comparison methods. The similarity score between two histograms is raised to the power of gamma before it is stored in a similarity matrix.

Firstly, the function initializes an empty similarity matrix with the purpose to store similarity scores between each pair of histograms in a set. Then, it iterates over each histogram and computes it's similarity score with all other histograms using *OpenCV's cv2.compareHist()* function, which in essence, takes two histograms and a comparison method as input, and outputs a numerical score representing the degree of similarity between the histograms. In this case, we're using the Correlation comparison metric, where a higher score translates to more similarity, and a score of 1 means that the histograms are identical. After, the maximum similarity score between all histograms is calculated, and elevated to the power of gamma, the result is stored in the similarity matrix at the corresponding position. Gamma is a parameter used to adjust the shape of similarity scores between pairs of histograms. By changing the value of gamma, the function can emphasize or de-emphasize the importance of the similarity score, which allows a certain degree of flexibility in similarity score calculations.

After assigning scores to each segment, the function *summarizeBudget*, as its name implies, serves as the primary driver for the overall video summarization process, employing the summarization method, Disparity Minimization, heavily inspired by the Knapsack optimization problem. Derived from a possible real life predicament of filling a "knapsack"with unique items, each with their own specific weight and value, some calculations must be made in order to maximize the total value of objects in the knapsack without exceeding the knapsack's weight capacity. Simplistically, we can describe the problem as follows:

- **Items:** We have N distinct items, each with their own weight and value.
- **Knapsack:** We have a knapsack that can carry a maximum weight.
- **Goal:** We aim to maximize the total value of objects we can place in the knapsack without surpassing the imposed weight limit.

While there is more than one problem variation to this puzzle due to the fact that one unique item might actually not be that unique and have several identical copies of it, in our project we specifically tackle the most known variant, the 0-1 knapsack problem, where each object type only has one copy, meaning each item can either be taken or not, 0-1 decision.

In relation to the video summarization subject, the knapsack problem comes into play when selecting video frames/segments to include in the summary where we can make use of the knapsack logic to solve the issue of selecting the most interesting segments of the video to include in the summary. By considering each set of frames as an item with "weight"(the cost of including the frame segment in the summary) and "value"(level of interestingness), we can apply the 0-1 knapsack problem logic to maximize the total value in the video summary without exceeding the budget, meaning selecting video segments significant enough to include in the summary while abiding by the time constraint. The "cost"of the segment is important because while there may be a highly interesting scene to include in the summary, it could consume a large part of the summary's capacity, leaving little room for other video segments, therefore decreasing the overall value that can be obtained.

3.3.3.2 Disparity Minimization

The mentioned summarization method, Disparity Minimization, has the main goal of producing a summary that captures the broadest possible range of content from the original video, thus ensuring diversity in the summary. However, it does so by seeking the most different set of frames, not taking into account the possibility that while two groups of frames might be closely similar, what's being expressed in them may hold very different meanings.

The logic behind Disparity Minimization is to select a subset of frames that minimizes the redundancy of information, with the underlying principle that including similar segments wouldn't add any new worthwhile information. Although there isn't any popular method for the knapsack problem that makes use of the disparity minimization, we can employ it when dealing with several possible solutions. As the name implies, by minimizing the disparities among them, we can achieve a more uniform distribution of solutions, ensuring a better balance between the conflicting objectives of maximizing value and adjusting to the weight limitation.

So, by representing a video as a set of frames, visual features can be distilled from them. We then define a similarity measure that computes the similarity between pair of frames, forming the logic behind our Disparity Minimization function. However, this only gives us a set of possible solutions which goes against the purpose of finding one definite solution. Therefore we can take advantage of optimization algorithms, most specifically the Greedy Optimization algorithm, used to optimize the Disparity Minimization function under the budget limitation, which iteratively selects the set of frames that provides the maximum gain in diversity when added to the current summary. We can define a greedy algorithm as an algorithm that aims to solve a problem by selecting the most appropriate option based on the current situation without regarding whether or not the current best result may not bring the overall optimal result. It may fail to find the globally optimal solution though, because it doesn't consider all the data, it is incapable of being aware of future choices it could make. It always chooses the most optimal solution at each step, even if it fails to find the solution that holds the maximum value.

We can thus define how both methods work together:

- The greedy algorithm calculates the diversity gain for each set of frames by using the disparity minimization function to calculate the value it would add to the summary if it was to be added to the summary set. Then, it adds the set of frames with the highest gain to the summary set.
- Diversity gains are recalculated for the remaining frames to yet again repeat the process of choosing the frames that offer the maximum diversity gain, each time subtracting the cost it takes to include the 'scene' in the summary from the available budget.
- The loop continues until we have exhausted the budget.

As such, we can summarize Disparity Minimization with the purpose to define what constitutes as a valid summary and the greedy algorithm with the role to achieve the most valuable summary under a time

constraint.

Returning to the *summaryBudget* function, we can now describe the logic behind the summarization process:

- **Initialization**: The function takes as input the similarity matrix that holds the similarity value computed for the set of histograms, a list that holds the cost for each set of frames, budget (total number of frames allowed in the summary) and an instance of the Disparity Minimization class that holds the aforementioned method.

- **Preprocessing**: An empty list is initialized with the purpose to hold the set of frames that are going to be selected for the final summary and a priority queue to manage the frames yet to be selected in the greedy algorithm loop. Each set of frames has a priority value to be determined by a benefit-cost calculation, the gain of adding the set of frames to the summary divided by the cost of said frames. Higher priority values means the set of frames has a higher benefit-to-cost ratio, which in essence translates to it adding more new and non-redundant information at a lower cost than other low priority valued frames.

- **Priority Selection Loop**: In this next step of the function, we enter a loop that continuously selects the set of frames with the highest gain from the top of the queue.

- **Gain Recalculation**: For each set of frames selected we have to recalculate its gain in accordance to the set of frames already selected for the summary, since the inclusion of new frames may change the benefit of adding this specific frame to the final summary set. If the recalculated gain is smaller than the gain value of the set of frames currently sitting at the top of the queue, the set reenters the queue with the new recalculated gain, and the loop restarts from step 2, where it selects the set of frames with the higher priority value and gain. In the case where the recalculated gain is not smaller, then the algorithms proceeds to the next step.

- **Frame set selection**: If the cost of including the afore selected set of frames does not exceed the budget, then the set is added to the summary and its gain is added to the total gain, which represents the cumulative diversity achieved by all frames selected to be included in the summary, a value we aim to maximize.

- **Precomputed Marginal Gains Update**: We also update the precomputed marginal gains for all set of frames that are still in the queue each time a set is added to the summary. We call this marginal because it accounts only for the gain relative to the current state of the summary. For each set of frames being added to the summary set, the marginal gain of the remaining frames may change, there may be frames in the queue that share similar information with the frames that were just added in the summary, which means that this specific set of frame is now less "interesting", it offers reduced diversity (gains) if it were to be added to the summary. Therefore we need to

recalculate the marginal gains for the frames in the queue each time a frame gets added to the summary, taking into account their similarity with the frames already in the summary. In order to reduce time spent on this process, we precompute and store these gains. When we arrive at the phase of considering a set of frames for inclusion in the summary, the algorithm already has these precomputed values available for quickly determining the frames marginal gain removing the need to calculate it from scratch.

- **Loop Continuation**: The loop continues until the budget has been completely exhausted or there are no more frames left in the queue. The function then returns the list of frames that constitute the summary.

Thus, having developed a list of video scenes the algorithm deemed worthy enough to be included in a video summary, we proceed with the final stage of the summarization process, to stitch the chosen frames in a temporally coherent video, where each scene in the summary is in the same order they were in the original video, in relation to time.

In order to evaluate the accuracy and effectiveness of our proposed method, it's important to put it through a scoring and evaluation process, that not only provides us information about the behaviour of our video summarization algorithm with the dataset Bosch provided, but also, by working with state of the art scoring and evaluation frameworks, we can compare our results with said state of the art methods.

We can therefore define three important purposes:

- **Validation of Methodology**: If we compare our machine-generated summaries with human-generated summaries, we can objectively assess how effective our algorithm is at capturing key information and important scenes from videos. Defining an evaluation process ensures that our method achieves its goal of condensing lengthy videos into short ones, that provides us with concise and informative summaries of the main events of the original video.

- **Performance Benchmark**: In order to properly assess our video summarization technique, we employ standardized metrics such as precision, recall, accuracy and F1 score that excel in evaluation video summarization algorithms. Thus, we can quantitatively measure the similarity between between machine-generated summaries and human-generated summaries, giving us a certain level of confidence in our own method, that can be fairly compared with other popular summarization techniques since we employ state of the art scoring and evaluation procedures.

- **Identifying Strengths and Weaknesses**: By putting our method through a scoring an evaluation process, we can ascertain about the strengths and weaknesses of it, giving us valuable insight that we can use to compare with human-generated summaries. We can therefore identify where our model underperforms in relation to an human-generated summary, which gives us guidance for potential improvements.

Following through these objectives, a segment-level scoring method is employed, we divide the video into three second shots with the purpose of improving the quality and efficiency of our video summarization method, since it facilitates identifying key events and gives us a more deeper level of comparison with a user based summary. This is all thanks to multiple specific reasons: **Granularity**, by segmenting the video into smaller three seconds shows, we can get a more granular analysis of video content, which allows us to capture finer details and events within the video, resulting in a more informative summary; **Event Identification**, it's easier to identify key events when we're studying distinct segments instead of just the whole raw video. It allows for a more representative summary, due to the fact that it's more capable of recognizing specific actions and transitions when it's scrutinizing a small chunk of video instead of being overloaded with details that do not pertain any valuable bits of information to the current timeframe being inspected; **Faster Summary Generation**, shorter segments are less computationally expensive to process and analyze compared to the entire video, being by far the most efficient method of summarization, specially when dealing with long videos or large datasets; **Diversity**, when dividing the video into segments, the summary can cover a broader range of content from the video, which helps algorithm avoid any bias towards specific sections of the video, ensuring more diversity is present. The video summarization algorithm then extracts visual information for each shot, calculating an histogram that represents the average graphical representation of the distribution of pixels in the 3 second shot frames. After, we compute the similarity score between the histograms of all shots, to then finally assess a priority value for each shot that determines whether or not a shot enters the summary. In the final summary, we can therefore correctly identify what timeframes from the original video were selected to be in the summary, which is key to be able to accurately compare with the shots a person deemed interesting enough to be included in his human-generated summary, which likewise, also assigns values to three second shots. Studying each segment from the video, the person assigns a score from 1 to 5 to the segment, 1 corresponding to a low level of interestingness and 5 to the highest level of interestingness, that dictates how probable it is to be included in the summary. Afterwards, we use the knapsack algorithm to compute a formula that decides between score levels and budget restraints, which segments are deemed valuable enough to be included in the human-generated summary. Finally, with two complete video summaries of the original video, we transcribe the segments that resulted from diving the video in three second shots, into a binary array, where through ones and zeros, illustrates which segments from the original video are present, or not, in the summary. Hence, we can now effectively compare both summaries, enabling us to analyze how efficient our method is at summarizing videos of the interior of a car, which scenes our algorithm correctly aligns with an human created summary, and which segments our method fails to correctly identify the level of interest. In summary, both types of summary are represented as binary sequences, where each element in the sequence corresponds to a segment of the video, and hold either a value of one or zero, one if the segment is included in the summary, zero if it is excluded from it.

3.3.4 Dataset groundtruth labeling

While there are prominent datasets used in the video summarization state of art, such as *TVSum* [27], focusing on videos from television broadcasts and *SumMe* [9], consisting of videos collected from Youtube covering domains such as music, sports and cooking, we only use the dataset provided by Bosch because of its uniqueness, it is its own domain, the interior of a vehicle, and with its specific characteristics. Employing our method in the previously mentioned datasets would not yield us any significant information that could be applied in improving the summarization of our own dataset. Consequently, evading those popular datasets, means we will be lacking in any ground truth made by experts in the field, as our dataset doesn't come with any type of meta information that could be useful to us. Therefore, we proceeded to score the Bosch dataset of our own accord, making use of an annotation tool that expedited the creation of the ground truth significantly, **MuViLab**, *MUltiple VIdeos LABelling tool*, created by Alessandro Masullo and Liam Dalgarno [2]. As its description reads, we can slip a video into short clips of 3 seconds, which are shown simultaneously in loop on screen, and easily label those segments according to our own scoring criteria. By using *MuViLab* we can considerably improve the efficiency and speed of scoring video segments compared to manual labeling, as it streamlines the annotation process, reducing the repetitiveness of the task, ultimately saving time and effort.

So, in conjunction with our scoring guideline, where we assign scores to video segments from one, deeming the specific scene as one that does not pertain any valuable information, to five, where we consider it to be of great importance to the user, and the use of a third party tool, *MuViLab*, that enables us to scrutinize our scores by visually comparing multiple shot sequences and their assigned values in a single window, we created a dataset consisting of a total of thirty nine videos and their respective ground truths, ranging from normal interactions inside of the vehicle, such as talking, pointing, reading, dancing, to more violent scenes covering intense discussions, fights, weapon usage, robberies and other actions with sexual connotation.

Having produced a viable dataset to work upon, we proceed with the next phase of the project, to evaluate the performance of our algorithm in summarizing our unique dataset domain. We implemented a method that compares both summaries, employing several metrics to measure how accurate our machine generated summary is, Precision, Recall and F1-score, this last one being an combination of the precision and recall metrics, and widely used on evaluation frameworks of video summarization methods, due to its ability to provide a balanced assessment of its performance. When analyzing the summary we consider how well its capability of including relevant content is, through the use of the recall metric, and how precise it is in excluding irrelevant content, through the precision metric, which culminates into ultimately relying on the F1-score as it owns its strength to its ability of accurately evaluating the trade-off between including important video scenes and avoiding impertinent information in a summary. Furthermore, the F1-score is also particularly useful when dealing with imbalanced datasets, since it accounts for situations where there is a significant difference in quantity of relevant and non-relevant video segments.

We can breakdown the function responsible for comparing machine-generated summaries with human-generated summaries as follows: First, the function takes as input two sets of summaries, user and machine generated summaries), the number of videos and their respective identifiers. After, to evaluate the summary, we iterate over each segment of the summary in a loop, updating with each iteration the count of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). While inside the loop, we calculate the accuracy, precision and recall metrics based on the specific TP, FP, TN and FN counts of the video being evaluated at the moment, to get a closer look at how well our algorithm responds to the unique characteristics of the video. We then compute the F1-score, using the precision and recall scores, considering the harmonic mean of the two. After the loop ends, we run the same metric calculations, but this time using the total global count of the four result classifications, to get a better look at the performance of our summarization method on the dataset. Finally, we calculate the mean of F1-scores for each video instead of using the total count of true positives, true negatives, false positives and false negatives, for these three benefits:

- **Handling Imbalanced Videos**: Videos in the dataset may vary in terms of duration's, what type of content they have and the level of importance they may hold to the viewer. Some videos can have a low number of highly relevant segments, where for the most duration of the video nothing noteworthy really happens, while others may have a more equally distributed number of important segments. Therefore, a global F1 score may not adequately capture the performance of our method across the different video event types of our dataset.

- **Weighted Assessment**: Computing the mean of the F1-scores allows for a weighted assessment of the performance of each video in the algorithm. Videos with longer duration's or higher metric disparities can greatly influence the overall evaluation of our method, overshadowing other videos. Alternatively, if we consider a mean of F1-scores, we give an equal weight in the assessment of our algorithm.

- **Comparative Analysis**: Whether its comparing the capabilities of different video summarization methods or evaluating the performance on multiple datasets, the use of a mean F1-score yields a standardized metric for comparison, that facilitates identifying datasets or methods that consistently perform well, by enabling a fair assessment of each.

For visual aid, we generate a graph, as seen in figure 8, to visualize and better compare the machine and user generated summaries, in which we create subplots for rows and columns, populating them with the video information. Each subplot represents the binary summary information of a video in a graphic way, in which values of ones represent segments that constitute the summary and zeroes represent segments that aren't part of it. In the *x* axis we display the order of the 3 second segments that constitute the length of the video. Some videos have a short duration and therefore have a lower number of segments to compare with, and likewise, some videos are high in length, meaning they get separated into more segments. This

greatly impacts the performance of our method in terms of accuracy, since it is easier to get an higher F1-score when the video is only 90 seconds and not 22 minutes, it only needs to label 30 segments for the first case, but for the longer video, it needs to predict the level of importance of around 440 segments. Furthermore, for a comparative analysis, we display machine and human generated summaries on two sub rows, where the first sub rows holds machine generated summaries and the bottom sub row human generated ones, and we also display the identifier of the video on the title of each subplot to be able to distinguish them.

3.3.5 Results

In this section, we present the results of our video summarization method on our own curated dataset, the interior of a vehicle. We compare the performance of our method with a popular state of the art method, Detect-to-Summarize Network (DSNet), a deep learning-based approach for video summarization, utilizing a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to analyze visual and temporal information in a video. As mentioned before, the evaluation metric used is the F1 score, which measures the balance between precision and recall.

Our video summarization method achieved a F1 score of approximately 29% on our dataset of the interior of the car while the state of the art method DSNet got a score of 32%, which compared to results of performing video summarization on state of the art datasets that break the 45% score, can be considered disappointingly low, however it can be explained by the complexity of the dataset. The interior of a car can be a challenging environment for applying video summarization algorithms since its lacking in visual diversity, due to similar backgrounds and repetitive, static scenes, which makes it difficult for algorithms to identify unique and interesting shots, resulting in lower F1 scores.

When we study the above figure that displays a comparison between machine and human generated summaries on a video by video basis, we can see how our method performs. The length of the video is represented by the x axis, where each 3 second segment of the video corresponds to a integer in the subplot, and in the y axis, we have values of zero and one, where we graphically represent segments of the original video that didn't get chosen to be part of the summary, in the case of the value zero, and segments that were deemed interesting enough to be included in the summary, in the case of the value one.

Analyzing the subplots of the figure, we can distinguish some unique characteristics of our method. By inferring upon the several 'spikes' on the graphs we can conclude that its not really efficient at including important scenes of the video that have a longer duration than normal, being the videos 7 and 22 of the figure 8 a clear example. While the person that annotated the video determined that a long continuous scene existed in the beginning of the video and was important enough to be in the summary, the algorithm, although also deeming it interesting, instead choose to represent the scene with several short segments rather than one longer and more temporally coherent segment. This may be due to the fact that our

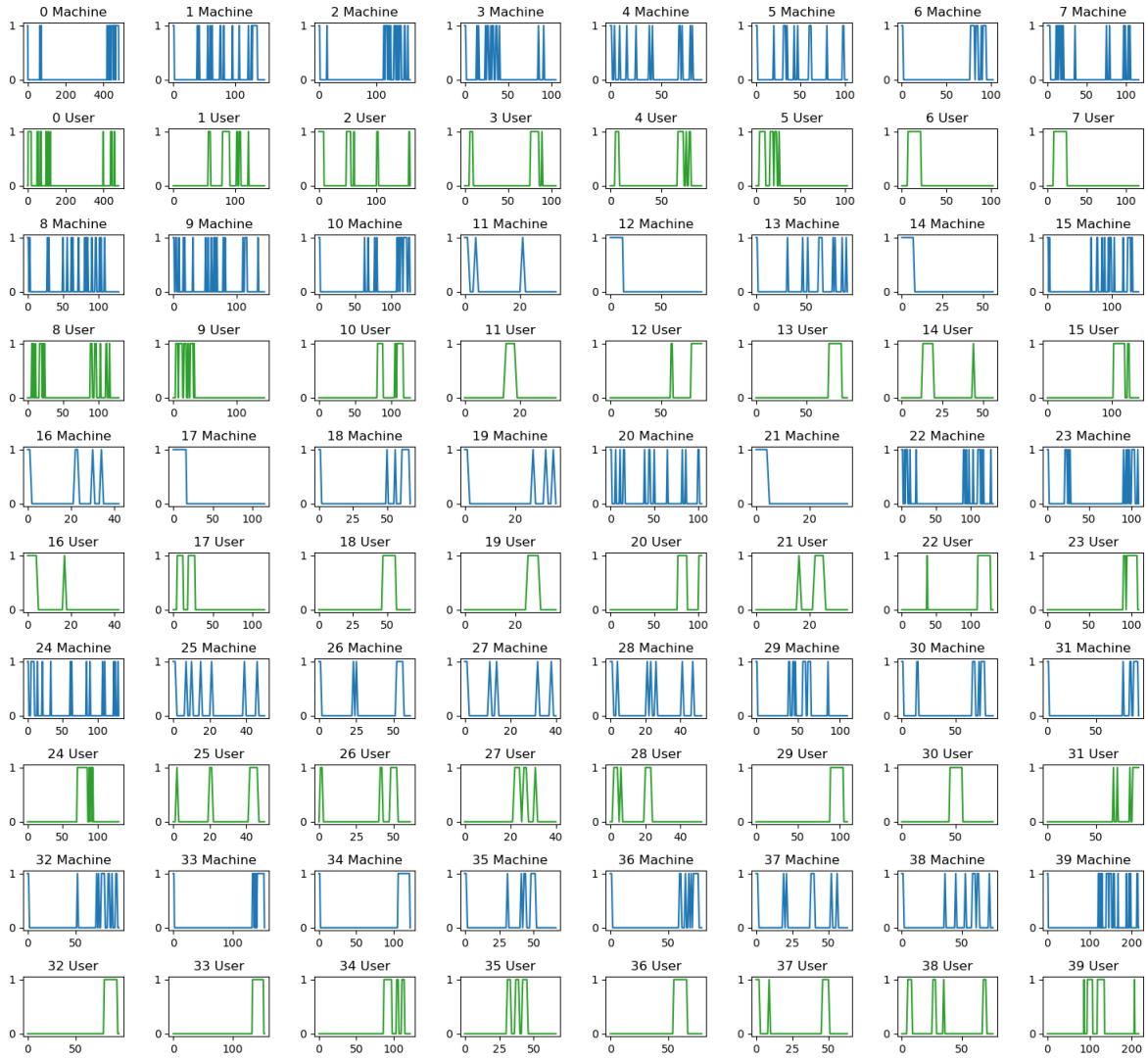


Figure 8: Graph displaying the best achieved results from our approach.

method assigns benefit-cost values to sequence of frames, where if temporally adjacent sets of frames are visually similar (known by calculating its histograms) it gives the frames that are in the middle of the sequence less value, effectively making them less probable to be in the summary. It prefers to pick a short sequence of a group of similar frames, so then it can spend the rest of the budget in more unique and diverse scenes. For example, in one of the videos a person slaps another, which obviously makes it important enough to be included in the summary, and our method doesn't fail to recognize it, however, it does fail to display the slapping scene in an uninterrupted video sequence, it cuts in the moment the slap connects to the victim, since the set of frames that are temporally located in that moment has high values of similarity to the frames that precede and follow it. Our method only considers its neighbours to be sufficiently interesting enough to be included in the summary, but this problem can be easily fixed by implementing an exception where we include all frames from a shot if they are very close to each other. However, in some cases, this motivation of the algorithm to relinquish visually similar groups of frames,

can be seen as positive in some videos as it gives more chance for other unique scenes in the video to be included in the summary. The client of this program, programmers choosing specific videos to feed their models, don't really need to see the whole portrayal of the action, mere seconds of the event are sufficient enough for him to infer upon whats happening and it leaves them with more scenes of the video to inspect upon.

One property of our method that gets misrepresented by the F1-score is its accuracy in correctly choosing violent scenes where a lot of movement is present, to be included in the summary. In all of the videos of our dataset, where actions of violence are present and have a lot of motion, our algorithm never failed to include them in the summary, it always correctly picked them for the summary. It's on the more mundane videos, where there is a lot of static, not much movement, lack of interaction between the passengers of the vehicle, that it gets exceptionally difficult for both our method and the DSNet to correctly identify scenes of interest, seen for example in videos 6 and 12 of the figure 8, normally resulting in the algorithm overestimating the level of interestingness the beginning of the video has.

As already mentioned before, its this type of domain, when the recording is situated solely on the interior of a car where for most instances not a lot of motion is present, normally resulting in a long continuous sequence of static, that presents a huge challenge for video summarization methods. It is specially daunting for our own algorithm since it relies so much on visual information, considering that the only method of feature analysis is the study of histograms, which is the biggest limitation imposed on our method. Histograms primarily capture pixel intensity distribution, and this may not effectively represent meaningful events in a video without motion. On scenarios where there is a lack of discriminative information, due to pixel intensity distribution remaining relatively constant over time, histograms may exhibit minimal variation over time, making it challenging to distinguish characteristics that allows us to identify and differentiate between significant content. Another limitation of histograms its their disregard for spatial relationships between pixels in the video, they only provide information on the distribution or frequency of pixel intensity in the frames, without taking into consideration the spatial arrangement of those pixels. Considering the nature behind the dataset domain, a predominant lack of motion in a significant number of videos, the arrangement of certain objects may play a crucial role in understanding the importance of a scene's content, which the histograms do not faithfully capture. Another downside of using histograms is their lack of temporal awareness, as they only focus on individual frames and don't capture changes that happen over time.

Despite the limitation imposed by the use of histograms, the inability to capture spatial relationships between pixels or objects in the video and disregard for important temporal relations, we managed to achieve a satisfying result, having a similar F1-score to a state of the art method. Considering how the motivation behind the proposed dissertation was helping machine learning engineers select videos that filled their criteria to feed them to their machine learning models, with a heavy focus on distinguishing violence scenes, we accomplished what was expect in the thesis and more. One of the discussed objectives was to give the user the ability to summarize a large quantity of

videos and review the video summary on the spot, without wasting the programmers time nor the computational resources available. This project was developed to be treated as an application accessible to multiple users at all times with fast results and low processing power to spend. We actually outperform the state of the art method DSNet on this objective, seeing as the use of histograms enables us an efficient analysis of large video datasets, unlike machine learning based algorithms, their computation complexity and training requirements makes them slower compared to histogram-based approaches, they are more computationally intensive and time-consuming. Histograms on the other hand significantly reduce computation complexity as they condense video data into a compact format, therefore making them faster to process due to the simplicity and speed of of histogram calculations, enabling a quick summarization and near real-time inspection of the produced video summary. They also do not require extensive training on labeled data, unlike machine learning algorithms, taking into account the fact that they rely on simple statistical calculations, not involving complex learning models or parameter tuning, making them easy to deploy and apply to new video datasets. Comparing the speed of our method with DSNet, we outperform the state of the art algorithm in more than 100% at summarizing our dataset of thirty nine videos, and this difference only grows bigger when we increase the number of videos. Furthermore, unlike DSNet that provides video summaries only after finishing the summarization process on all videos, our application immediately yields the summary of the video after processing it, eliminating long waiting times.

In summary, by using histograms, we developed an approach that is fast and efficient at summarizing videos, making it suitable for an application where near real-time processing and quick analysis is required, though we forgo any spatial or temporal relations that may exist in the content of the video. We could have tackled the problem using machine learning algorithms but at increased computational complexity and training time, both requisites that we can't fulfill as per the discussed allocation of processing power and time budget.

4

Conclusion and Future Work

This section is composed of two sections, the first one presents a conclusion of all the work done on this thesis along side some observations and thoughts. The second and last section, represents the future work that this thesis opens up with additional approaches that can be taken in order to improve the summarization algorithm.

4.1 Conclusion

Our main objective in this dissertation was to develop a functional approach to summarize recordings of the interior of vehicles. We aimed to provide a fast and easy to use solution that captured the most important scenes, enabling an efficient retrieval and analysis of relevant content that may be present in the video and is of interest to the user.

To achieve this, we employ a series of algorithms and techniques that work synergistically to extract key frames, ascertain about their value of interestingness and construct an informative summary. Our pipeline begins with the computation of histograms, capturing pixel intensity, color distributions within a set of frames, histograms that we take the mean of to obtain representative values of said set of frames, serving as the basis for the analysis.

We then assess the similarity between frames by comparing the mean of histograms of consecutive set of frames. This similarity metric permits us to identify segments within the video that exhibit similar content, visual patterns, that often indicate the presence of significant events.

For the purpose of selecting video scenes for the summary, we evaluate upon their value of interestingness and level of priority, by implementing disparity minimization, a technique that allows us to assign priority values to each segment based on their degree of similarity to other segments. As such, effectively appraising the uniqueness and importance of each segment, we ensure that the final summary captures the most significant, representative content from the original video.

Ending the selection process, we utilize a greedy algorithm to optimize it. This algorithm iteratively selects segments with the highest priority values, gradually building the summary by incorporating the

most significant scenes. This proves to be an effective way to quickly and correctly identify key segments that convey the most diverse events present in the video.

Moving on to the evaluation of the summarization outcome, we compared the performance of our method with a state of the art approach called DSNet. Our results showed that our algorithm achieved a summary with a F1 score of approximately 29%, while DSNet obtained a score of 32%. The specific nature of our dataset, focusing on the interior of the vehicle, presented a unique challenged that impacted the overall performance of the two methodologies, nevertheless, despite the low metric score, we managed to provide a quick and accessible solution for video summarization in this specific context, that didn't falter on correctly choosing scenes of violence for the final summary.

In conclusion, we developed a video summarization approach tailored for the dataset Bosch provided us with and by using histogram-based analysis, similarity assessment, disparity minimization and a greedy algorithm, we were able to assemble important events into summaries that successfully captured the essential content of the vehicle recordings. While improvements can be made to further enhance the summarization accuracy, we achieved our goal of providing a quick and user-friendly application for video retrieval and analysis.

4.2 Future Work

Although our video summarization solution has shown promising results, there are several areas on which we can improve. One potential avenue for improvement lies in the integration of deep learning techniques, widely used in state of the art approaches. Deep neural networks, such as CNN or RNN, can capture complex visual patterns and temporal dependencies within the video, therefore potentially improving the accuracy in detecting scenes and better understanding the displayed contextual information. However, despite the fact that using deep neural networks may have great potential, a lot of accommodation for its usage needs to be done considering the limited budget constraints presented in terms of computation power, as seen by the use of the DSNet approach.

Object detection and recognition algorithms can also serve as an additional route we can take to reveal more meaningful scenes of interest allowing for a more context-aware summarization process, deepening the level of key information and insight we can provide that its crucial for the user in choosing videos for their machine learning training datasets.

Finally, if we address how our evaluation is done, we can conclude that our work would have benefited more if more comprehensive and robust evaluation measures were employed. In future works, we should focus on enhancing the evaluation framework, by adding additional evaluation metrics, such as coverage, diversity and representativeness; gathering user feedback on the accuracy and quality of the machine and human generated summaries; improving upon in-depth analysis of specific events and how they are perceived by machine and humans, shedding light on the strengths and weaknesses of our approach.

Bibliography

[1] M. Ajmal et al. “Video Summarization: Techniques and Classification”. In: *Computer Vision and Graphics*. Ed. by L. Bolc et al. 2012, pp. 1–13 (cit. on p. 17).

[2] L. D. Alessandro Masullo. *MuViLab*. https://github.com/ale152/muvilab. 2018 (cit. on p. 26).

[3] S. E. F. de Avila et al. “VSUMM: An Approach for Automatic Video Summarization and Quantitative Evaluation”. In: *2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*. 2008, pp. 103–110. doi: 10.1109/SIBGRAPI.2008.31 (cit. on p. 13).

[4] M. Basavarajaiah and P. Sharma. “GVSUM: generic video summarization using deep visual features”. In: *Multimedia Tools and Applications* 80 (Apr. 2021). doi: 10.1007/s11042-020-10460-0 (cit. on pp. 13, 14).

[5] E. P. Bennett and L. McMillan. “Computational Time-Lapse Video”. In: SIGGRAPH ’07 (2007), 102–es. doi: 10.1145/1275808.1276505 (cit. on p. 18).

[6] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. doi: 10.1109/CVPR.2005.177 (cit. on p. 9).

[7] F. Diebold. “On the Origin(s) and Development of the Term ’Big Data’”. In: *SSRN Electronic Journal* (Sept. 2012). doi: 10.2139/ssrn.2152421 (cit. on p. 1).

[8] U. Gawande, K. Hajari, and Y. Golhar. “Deep Learning Approach to Key Frame Detection in Human Action Videos”. In: *Recent Trends in Computational Intelligence*. Ed. by A. Sadollah and T. S. Sinha. Rijeka: IntechOpen, 2020. Chap. 7. doi: 10.5772/intechopen.91188 (cit. on p. 11).

[9] M. Gygli et al. “Creating Summaries from User Videos”. In: *Computer Vision – ECCV 2014, European Conference on Computer Vision*. Ed. by D. Fleet et al. 2014, pp. 505–520 (cit. on p. 26).

[10] W. Hu et al. “Semantic-Based Surveillance Video Retrieval”. In: *IEEE Transactions on Image Processing* 16.4 (2007), pp. 1168–1181. doi: 10.1109/TIP.2006.891352 (cit. on p. 6).

[11] S. Jadon and M. Jasim. "Unsupervised video summarization framework using keyframe extraction and video skimming". In: *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*. 2020, pp. 140–145. doi: 10.1109/ICCCA49541.2020.9250764 (cit. on p. 12).

[12] M. Kini M. and K. Pai. "A Survey on Video Summarization Techniques". In: *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*. Vol. 1. 2019, pp. 1–5. doi: 10.1109/i-PACT44901.2019.8960003 (cit. on p. 8).

[13] Y. J. Lee, J. Ghosh, and K. Grauman. "Discovering important people and objects for egocentric video summarization". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 1346–1353. doi: 10.1109/CVPR.2012.6247820 (cit. on p. 13).

[14] M. S. Lew, N. Sebe, and J. P. Eakins. "Challenges of Image and Video Retrieval". In: *Lecture Notes in Computer Science*. 2002, pp. 1–6. doi: 10.1007/3-540-45479-9_1 (cit. on p. 4).

[15] G. Liang et al. "Video summarization with a dual-path attentive network". In: *Neurocomputing* 467 (2022), pp. 1–9. doi: https://doi.org/10.1016/j.neucom.2021.09.015 (cit. on p. 15).

[16] R. Lienhart, S. Pfeiffer, and W. Effelsberg. "Video Abstracting". In: *Communications of the ACM* 40 (Sept. 2000). doi: 10.1145/265563.265572 (cit. on p. 9).

[17] T.-Y. Liu et al. "Shot reconstruction degree: a novel criterion for key frame selection". In: *Pattern Recognit. Lett.* 25 (2004), pp. 1451–1457 (cit. on p. 13).

[18] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. url: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[19] S. Mohamed, A. Ghalwash, and A. Youssif. "Semantic-Based Video Retrieval Survey". In: *Iraqi Journal of Science* 59 (2018), pp. 739–753. doi: 10.24996/ijs.2018.59.2a.12 (cit. on pp. 4, 5).

[20] A. Money and H. Agius. "Video summarisation: A conceptual framework and survey of the state of the art". In: *Journal of Visual Communication and Image Representation* 19 (Feb. 2008), pp. 121–143. doi: 10.1016/j.jvcir.2007.04.002 (cit. on pp. 9–11).

[21] J.-M. Morel and G. Yu. "Is SIFT scale invariant?" In: *Inverse Problems and Imaging* 1 (Feb. 2011). doi: 10.3934/ipi.2011.5.115 (cit. on pp. 9, 13).

[22] M. Mühling et al. "Deep learning for content-based video retrieval in film and television production". In: *Multimedia Tools and Applications* 76 (Nov. 2017). doi: 10.1007/s11042-017-4962-9 (cit. on p. 6).

[23] M. S. Nixon and A. S. Aguado. “Chapter 4 - Low-level feature extraction (including edge detection)”. In: *Feature Extraction Image Processing for Computer Vision (Third Edition)*. 2012, pp. 137–216. doi: https://doi.org/10.1016/B978-0-12-396549-3.00004-5 (cit. on p. 12).

[24] M. Otani et al. “Video summarization using textual descriptions for authoring video blogs”. In: *Multimedia Tools and Applications* 76 (May 2017), pp. 1–19. doi: 10.1007/s11042-016-4061-3 (cit. on p. 15).

[25] S. F. S. Pfeiffer R. Lienhart and W. Effelsberg. *Abstracting Digital Movies Automatically* (cit. on p. 8).

[26] C. Snoek and M. Worring. “Concept-Based Video Retrieval”. In: *Foundations and Trends in Information Retrieval* 2 (Jan. 2009), pp. 215–322. doi: 10.1561/1500000014 (cit. on p. 5).

[27] Y. Song et al. “TVSum: Summarizing web videos using titles”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 5179–5187. doi: 10.1109/CVPR.2015.7299154 (cit. on p. 26).

[28] T. Truong and S. Venkatesh. “Video abstraction: A systematic review and classification”. In: *TOMCCAP* 3 (Jan. 2007) (cit. on pp. 8, 13).

[29] M. Wang and H. Zhang. “Video Content Structuring”. In: *Scholarpedia* 4.8 (2009). revision #91922, p. 9431. doi: 10.4249/scholarpedia.9431 (cit. on p. 7).

[30] B.-W. Wang et al. “Semantic Video Retrieval by Integrating Concept- and Content-Aware Mining”. In: *2011 International Conference on Technologies and Applications of Artificial Intelligence*. 2011, pp. 32–37. doi: 10.1109/TAAI.2011.14 (cit. on p. 6).

[31] Y. Wang, Q. Chen, and B. Zhang. “Image enhancement based on equal area dualistic sub-image histogram equalization method”. In: *IEEE Transactions on Consumer Electronics* 45.1 (1999), pp. 68–75. doi: 10.1109/30.754419 (cit. on p. 12).

[32] M. S. Willick. “Artificial Intelligence: Some Legal Approaches and Implications”. In: *AI Magazine* 4.2 (June 1983), p. 5. doi: 10.1609/aimag.v4i2.392 (cit. on p. 1).

[33] X.-D. Yu et al. “Multilevel video representation with application to keyframe extraction”. In: *10th International Multimedia Modelling Conference, 2004. Proceedings.* 2004, pp. 117–123. doi: 10.1109/MULMM.2004.1264975 (cit. on p. 13).

[34] S. Zhang, Y. Zhu, and A. K. Roy-Chowdhury. “Context-Aware Surveillance Video Summarization”. In: *IEEE Transactions on Image Processing* 25.11 (2016), pp. 5469–5478. doi: 10.1109/TIP.2016.2601493 (cit. on p. 15).