Query-based Video Summarization Using Machine Learning and Coordinated Representations

Pedro Sandino Atencio Ortiz

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de las Computación y de la Decisión
Medellin, Colombia
2020

Generación de Resúmenes de Videos Basada en Consultas Utilizando Aprendizaje de Máquina y Representaciones Coordinadas

Pedro Sandino Atencio Ortiz

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de las Computación y de la Decisión
Medellin, Colombia
2020

Query-based Video Summarization Using Machine Learning and Coordinated Representations

Pedro Sandino Atencio Ortiz

A dissertation submitted in partial fulfillment of the requirements for the degree of:
Engineering PhD - Systems and Informatics

Director:
Ph.D. John Branch Bedoya

Co-director:
Ph.D. Germán Sánchez Torres

Co-director:
Ph.D. Claudio Delrieux

Research Groups:
GIDIA - Universidad Nacional de Colombia
Automática, Electrónica y Ciencias Computacionales - Instituto Tecnológico Metropolitano

Universidad Nacional de Colombia
Facultad de Minas, Departamento de Ciencias de la Computación de la Decisión
Medellín, Colombia
2020

Dedication

To my son Martín, my wife Catalina, my father Martín and my mother Malena.

Acknowledgments

This thesis would not have been possible without the participation of numerous professors, colleagues, and friends who decided to support me with their thematic and methodological knowledge, and their constant contributions in multiple discussions, both in rigorous academic spaces and in not so strict spaces when the opportunity was given. On the other hand, not only the thematic and methodological support was important to me, but the constant emotional support that is required to persist and resist the emotional effort and charge that a doctoral study requires.

First of all, I would like to thank my directors: Professor John Branch, for always fostering the constant discussion that keeps the flame of research alive among colleagues and friends that make up our research group, and for helping me always find solutions to situations in which only experience is able to clear the dense fog that covers the eyes and allows to see the light again; Professor German Sanchez, who has guided me along the path of research and analytical thinking very early in my undergraduate studies, and who always helps me to put my feet on the ground to take firm steps; and Professor Claudio Delrieux who, after having been my teacher for a long time during my undergraduate studies, the very evolution of life allowed me to meet him again and have the opportunity to make him participate in my doctoral studies, and in which I have found constant support at the academic level, and also gave me the opportunity to do my doctoral internship in his laboratory.

I also want to thank the constant support I received from my family: son, wife, father, mother, and sister, for always being present and taking the time to listen to me in their free spaces and allow me to share my problems, joys, failures, and successes during all this time.

Finally, I want to thank all the institutions that have made this doctoral work possible: Universidad Nacional de Colombia for receiving me from my master's studies, Colciencias for giving me the opportunity to be a beneficiary of the Call 727 of 2015 through which to find financial support for my studies, to the Instituto Tecnológico Metropolitano for allowing me to be part of its teaching group and to support me over time, administrative procedures and various resources logistics to complete my doctoral studies, and to the Universidad Nacional del Sur for receiving me at his alma mater via Professor Claudio Delrieux for my doctoral internship.

Abstract

Video constitutes the primary substrate of information of humanity, consider the video data uploaded daily on platforms as YouTube: 300 hours of video per minute. Video analysis is currently one of the most active areas in computer science and industry, which includes fields such as video classification, video retrieval and video summarization (VSUMM).

VSUMM is a hot research field due to its importance in allowing human users to simplify the information processing required to see and analyze sets of videos, for example, reducing the number of hours of recorded videos to be analyzed by a security personnel. On the other hand, many video analysis tasks and systems requires to reduce the computational load using segmentation schemes, compression algorithms, and video summarization techniques.

Many approaches have been studied to solve VSUMM. However, it is not a single solution problem due to its subjective and interpretative nature, in the sense that important parts to be preserved from the input video requires a subjective estimation of an importance score. This score can be related to how interesting are some video segments, how close they represent the complete video, and how segments are related to the task a human user is performing in a given situation. For example, a movie trailer is, in part, a VSUMM task but related to preserving promising and interesting parts from the movie but not to be able to reconstruct the movie content from them, i.e., movie trailers contains interesting scenes but not representative ones. On the contrary, in a surveillance situation, a summary from the closed-circuit cameras needs to be representative and interesting, and in some situations related with some objects of interest, for example, if it is needed to find a person or a car.

As written natural language is the main human-machine communication interface, recently some works have made advances in allowing to include textual queries in the VSUMM process which allows to guide the summarization process, in the sense that video segments related with the query are considered important.

In this thesis, we present a computational framework to perform video summarization over an input video, which allows the user to input free-form sentences and keywords queries to guide the process by considering user intention or task intention, but also considering general objectives such as representativeness and interestingness. Our framework relies on the use of pre-trained deep visual and linguistic models, although we trained our visual-linguistic coordination model. We expect this model will be of interest in cases where VSUMM tasks requires a high degree of specification of user/task intentions with minimal training stages and rapid deployment.

Keywords: Query-based video summarization, video-text deep coordination models, video analysis framework.

Resumen

El video constituye el sustrato primario de información de la humanidad, por ejemplo, considere los datos de video subidos diariamente en plataformas como YouTube: 300 horas de video por minuto. El análisis de video es actualmente una de las áreas más activas en la informática y la industria, que incluye campos como la clasificación, recuperación y generación de resúmenes de video (VSUMM).

VSUMM es un campo de investigación de alto dinamismo debido a su importancia al permitir que los usuarios humanos simplifiquen el procesamiento de la información requerido para ver y analizar conjuntos de videos, por ejemplo, reduciendo la cantidad de horas de videos grabados para ser analizados por un personal de seguridad. Por otro lado, muchas tareas y sistemas de análisis de video requieren reducir la carga computacional utilizando esquemas de segmentación, algoritmos de compresión y técnicas de VSUMM.

Se han estudiado muchos enfoques para abordar VSUMM. Sin embargo, no es un problema de solución única debido a su naturaleza subjetiva e interpretativa, en el sentido de que las partes importantes que se deben preservar del video de entrada, requieren una estimación de una puntuación de importancia. Esta puntuación puede estar relacionada con lo interesantes que son algunos segmentos de video, lo cerca que representan el video completo y con cómo los segmentos están relacionados con la tarea que un usuario humano está realizando en una situación determinada. Por ejemplo, un avance de película es, en parte, una tarea de VSUMM, pero está relacionada con la preservación de partes prometedoras e interesantes de la película, pero no con la posibilidad de reconstruir el contenido de la película a partir de ellas, es decir, los avances de películas contienen escenas interesantes pero no representativas. Por el contrario, en una situación de vigilancia, un resumen de las cámaras de circuito cerrado debe ser representativo e interesante, y en algunas situaciones relacionado con algunos objetos de interés, por ejemplo, si se necesita para encontrar una persona o un automóvil.

Dado que el lenguaje natural escrito es la principal interfaz de comunicación hombre-máquina, recientemente algunos trabajos han avanzado en permitir incluir consultas textuales en el proceso VSUMM lo que permite orientar el proceso de resumen, en el sentido de que los segmentos de video relacionados con la consulta se consideran importantes.

En esta tesis, presentamos un marco computacional para realizar un resumen de video sobre un video de entrada, que permite al usuario ingresar oraciones de forma libre y consultas de palabras clave para guiar el proceso considerando la intención del mismo o la intención de la tarea, pero también considerando objetivos generales como representatividad e interés. Nuestro marco se basa en el uso de modelos visuales y lingüísticos profundos pre-entrenados,

aunque también entrenamos un modelo propio de coordinación visual-lingüística. Esperamos que este marco computacional sea de interés en los casos en que las tareas de VSUMM requieran un alto grado de especificación de las intenciones del usuario o tarea, con pocas etapas de entrenamiento y despliegue rápido.

Keywords: Generación de resúmenes de video basada en consulta, modelos de coordinación de video a texto, análisis de video.

Content

1. Introduction

1.1. Context

Video summarization (VSUMM) aims to reduce the length of an input video V, while preserving frames, segments or scenes that may be valuable, either visually when the purpose is to retain visual information actually present in frames, or semantically because they represent information about the meaning or the story present in V. VSUMM models are becoming critical, given the current deluge of video creation and streaming in almost all aspects of our culture.

Given the diversity of sources and purposes associated to VSUMM, there is a wide diversity of methods, all of which share some common ideas. In general, a video segmentation stage extracts segments or similar frame sequences which compose the complete video, i.e., given an input video V a segmentation method returns a list of segments $S = \{s_0, s_1, s_2, ..., s_n\}$ where a segment s_i is a pair (a, b) of time stamps or video-frames. A common yet limited approach consists of performing a uniform sampling over V, taking segments of user-defined length and spacing. More complex approaches use dynamic video segmentation, where sequential frames closed in feature space are taken as segments [1].

From a computational perspective, VSUMM consists on solving the knapsack problem, i.e., maximize a given score (e.g., value score), subject to a summary length constraint, usually, 15 % of video length $|V|$ [2]. The VSUMM knapsack problem is shown in equation 1-1, where z_i is a binary array with value 1 for segments to be maintained in summary and 0 otherwise, $R(s_i)$ is a function of predicted value for extracted video segment s_i and $\alpha \in (0, 1)$ is an arbitrary summary threshold.

$$\begin{aligned} &\max_{z} \sum_{i=1}^{m} z_i R(s_i) \\ &s.t. \sum_{i=1}^{m} z_i |s_i| \leq \alpha |V| \end{aligned} \tag{1-1}$$

In terms of its output, VSUMM can produce three kinds of results (see Fig. **1-1**) as follows:

- **Skim**: An output video $V' \subset V$ composed of the most valuable segments from V is returned.

- **Dynamic Fast-Forward**: An output video V' with a variable playback speed is returned, where speed is the regular one for the most valuable segments and is accelerated on the contrary.
- **Storyboard**: A sequence of key-frames $K \in V$ is returned.

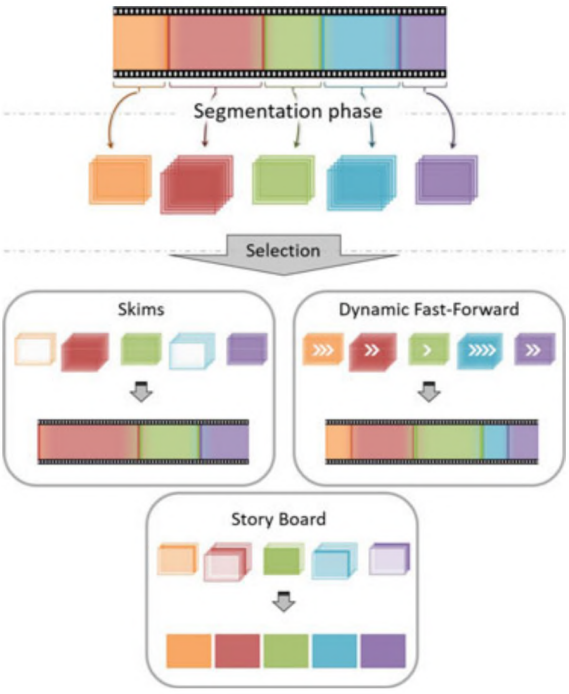

Figure 1-1.: A general framework for VSUMM. Taken from [3].

As we later explore in Section 1.2.1, VSUMM is a multiobjective optimization task in the sense that the value-score function R (see Equation 1-1) depends on multiple considerations which can be extracted or not directly from V. For example, the value-score can be related to what it can be considered important, interesting, or representative. In the literature, many approaches have been proposed in order to solve VSUMM, considering one or many of these objectives.

A new research trend arose recently in the VSUMM community, which is to include user purposes in the process. Recent works (see for instance [4, 5]) have proposed the injection of query-words, which allows specifying the result of the VSUMM task according to the task or user objectives. The VSUMM community has named this approach query-based video summarization (Q-VSUMM).

General challenges

Summarizing a given video implies estimating which segments or frames are considered most valuable for the user, concerning the task they is performing, while preserving the completeness of the video story. The number of complex and non-deterministic cognitive processes that need to be in consideration to achieve the latter can be considerable: Visual recognition, scene understanding, detection of spatio-temporal interactions and relations between objects, object-concept abstractions, attention, and semantics [6], among others, each of which is an open problem by itself.

A direct approach to estimate the value of a video segment consists of measuring the variance of visual-only features. In this way it is possible to suppose that consecutive segments with low visual variance may be shortened or upright discarded from the final summary, i.e., only segments with high visual variance can be considered unusual, and for that reason valuable. Nonetheless, this approach can be naive in the sense that value must consider more elements, for example, categorical variance and attention, among others.

Visual variance can also be modeled through time series analysis, either using classic methods as conditional random fields (CRF) or deep learning approaches using recurrent neural networks as long-short term memory networks (LSTM), just to mention the most popular approaches in time series analysis. This kind of analyses constitutes a challenge due to the impact in the result that has the kind of feature used, the architecture selected, and the high computational cost that implies.

Recent approaches use object recognition [7, 5] which allows to estimate the value of a segment by considering objects with an score of importance for some task. An open challenge in this topic consists on the limited number of categories that can be classified by a recognition model (1000 for an image-net model) which constitutes a barrier to determine whether the recognized object is related with the associated task. For example, in categories *boat*, *sea* and *fishing rod* appears in a visual scene, implicitly we can assume that the latter is related with the concepts *fishing* and *fish*. The latter is also related to a significant challenge for the Q-VSUMM task, where it is required a shared latent space for visual and linguistic features in order to perform textual queries in the video.

The increased use of linguistic information to solve semantic tasks in the analysis of images and videos has led to the creation of various databases of videos with written annotations in natural language made by humans [6, 7, 9, 8], which has enhanced the development of video analysis applications with high-level semantics. Nonetheless, annotations in natural language can contain actions and hierarchical relationships hardly recognizable by a traditional classification model. For example, the annotation: "*cheerleaders are getting thrown in the air*"(see Fig. **1-2**) requires the detection of people in the scene, poses, spatial configurations, and temporal behaviors, and some grasp of metaphorical language, among others.

Annotator 1: a man wearing a black suit
Annotator 2: a man speaks to audiences indoors

Annotator 1: cheerleaders are getting thrown in the air
Annotator 2: cheerleading group does throwing moves on a stage

Figure 1-2.: An example from TRECVid video-to-text dataset [8].

Then, it is of current interest to develop strategies that can take advantage from this textual annotations and enrich the estimation of the value of a video segment to be preserved on the final summary.

Progress in the field

The state-of-the-art has shows advances in the development of VSUMM models based on machine learning [1, 5, 10] and various deep learning architectures such as deep convolutional networks (DCN) [11], retrospective encoders [12], reinforcement learning [13], end-to-end memory networks [14] and long-short term memory networks [15], which can learn automatically intrinsic relations between input video, its human-made summary, and associated metadata.

Other domains such as visual psychology [16, 17, 18], neuroscience [19], and computational linguistics [20, 21, 6] have expanded the frontiers in image and video understanding, particularly, in the inclusion of linguistic knowledge in different tasks of automatic visual analysis, reinforced in turn by recent researches that support the power of human thought-language relation.

Specific challenges

As far as we know from the reviewed literature, some challenges remain open, as follows:

- Time series analysis of relationships between visual concepts to estimate the value-score of segments or frames of the input video [22]. Visual action recognition [23, 24, 25] and

visual question answering [26] are interesting frameworks to work on it.

- Modern approaches to VSUMM, which incorporates human linguistic knowledge, and use general context corpuses such as Wikipedia [27]. It is crucial to evaluate the impact of the various linguistic corpus with different contexts (e.g., medical, news, security, engineering, among others) in the quality of the VSUMM task.
- The exploration of various deep learning architectures capable of representing the hierarchical and structural nature of the human language syntax in order to obtain conceptual relations of higher-order in the video. According to Zhang et. al. [15]: "*In particular, it would be very productive to explore new sequential models that can enhance LSTM's capacity in modeling video data, by learning to encode semantic understanding of video contents and using them to guide summarization and other tasks in visual analytics*".
- To include in the VSUMM task, new elements from visual psychology and widely used in image analysis such as emotions, context, attention, and importance, among others.
- The exploration of different linguistic representations or embeddings [28, **?**, 29] to extract semantic relationships from human-made annotations in the VSUMM task.
- The ability to generate video summaries dependant of the user intention or context [5], in other words, the ability to personalize VSUMM [3].

1.2. Related work

1.2.1. VSUMM as a multi-optimization problem

The major challenge of VSUMM task consists of defining the criteria for constructing the value-score function in order to select relevant segments to be preserved in the final summary. Initial works in VSUMM considered a global score function generally related to what could be considered "important" in the video. Nonetheless, gradually the VSUMM community began to consider the task as a multi-objective optimization [3] in the sense that users do not only consider what it is important, but what is general and what is interesting, among other criteria. Recent works consider the following objectives for VSUMM:

- **Interestingness**: Commonly accepted as the local importance of a video segment [2].
- **Diversity**: This is related to a low redundancy in the video summary [30].
- **Representativeness**: How much the summary represents the input video. The latter can be expressed in terms of the distance in some conceptual sense between the

summary and the input video [3].

- **Uniformity**: Generally accepted as temporal coherence [3].
- **Importance**: Related to how much the summary contains relevant objects, actions, or relations [31].

Generally, interestingness is approached as a regression or ranking task that depends on the features extracted from the video segments. Traditional approaches focus on the use of exclusively visual characteristics [32, 33], such as SIFT [34], HOG [35], optical flow [36], difference of images [37] or combinations of characteristics, specially designed for some instances of the application (hand-crafted features). For example, the importance of sport video segments is related to the events and actions defined by the particular sport rule, as well as the appearance of certain scenes or objects of interest[38]. A radically different approach is required to generate summaries of films, for example, using actor recognition or subtitle analysis [39].

The increasingly use of video capture devices, such as sports cameras or mobile phones, generated a new category of videos called by the scientific community as egocentric videos [22], composed mainly of free content, without any edition and with high relationship with the interest of the person who records the video. This category of videos led to addressing the problem of VSUMM from a more general perspective and the use of multi-modal data.

Works like Lee et al. [7] propose the use of features such as fixation of the gaze, the frequency of appearance of an object and interaction with it, in order to receive data from the user and the environment in which the video is developing, which help to estimate the importance of a visual scene. Some approaches, such as the one proposed by Sun et al. [40] assume the importance of the segments of the video, depending on the topic contained in it. In that work, the authors use a Ranking-SVM [41] to learn the importance of a video segment, based on the weighted annotations made by groups of 5 users. The videos were categorized by topic according to the search terms used on YouTube to obtain the videos. Attention has been used as a relevance criterion in [42]. The authors propose a computational model of attention that takes into account information from different domains, particularly visual (faces, salience, and behavior of the camera) and the auditory case (salience, speech, and music). Subsequently, the user attention curves are computed from the extracted models and the summary videos made by users.

Representativeness and uniformity

Representativeness in VSUMM implies considering global metrics of mutual information, in some cases performing a clustering process to determine the groups of events that make up the story contained in the video or an optimization process that maximizes a metric of mutual information between the summary and the video. In the method proposed in [22], the authors

consider that the segments in the video summary should lead from one to another. For this, the authors use camera-level features and visual features such as HOG. They were using a random-walk search strategy to construct the possible sequences of segments that constitute the final summary. Kim et al. [43] formulate the problem as the subset selection in a graph of web images and video frames. Given this graph, the authors optimize an anisotropic diffusion objective to select a set of densely connected but diverse nodes, which leads to balanced summaries in interestingness and representativeness. Gygli et al. [10] estimate the representativeness by using the k-medoids clustering algorithm to select the best k segments that represent the video, using deep features trained in image-net [44]. Uniformity is related to the segmentation stage. Recent works [10, 45, 5] argue that using a uniform segmentation scheme generates better results than clustering-based approaches, in terms to maintain the temporal coherence of the video and the common sense of the story.

1.2.2. Summary personalization

Main VSUMM methods aim to generate general summaries, that is, independent of the content of the video, the method is capable of obtaining the segments that best represent the entire video. Nonetheless, certain aspects of the video, such as the time of appearance of certain elements, the camera speed, and proximity to specific objects, are related to the interests of the person making the recording. On the other hand, it is desirable to be able to generate different summaries according to user intentions (see Fig. **1-3**). The customization of the VSUMM process is of great interest and is a topic of considerable activity within the community [46, 47, 5, 4].

Some approaches try to obtain personalization through the learning of patterns found by analyzing summaries of videos classified at the user level, that is, for each video, there are n summaries discriminated by users [48]. This approach has the limitation that it does not take into account the initial interest of the individual at the time of performing the VSUMM process, so the learning focuses on finding visual patterns of interest for each individual in this task.

Other approaches propose the capture or extraction of information that allows customizing videos according to the behavior of an observer. The relationship between certain brain waves (α waves) and visual characteristics was studied by Ng et al. [50]. In this work, the authors manage to capture alpha waves of a group of people while watching a video, and with this information, group the visual characteristics of the video into two sets or classes. Subsequently, using a support vector machine (SVM), the authors classify a segment of a video as important or not. A similar approach is the one proposed by Xu et al. [51], in which they use follow-up of the gaze of a set of human observers to determine which elements or events are important in a visual scene. This type of approach finds its main limitation in

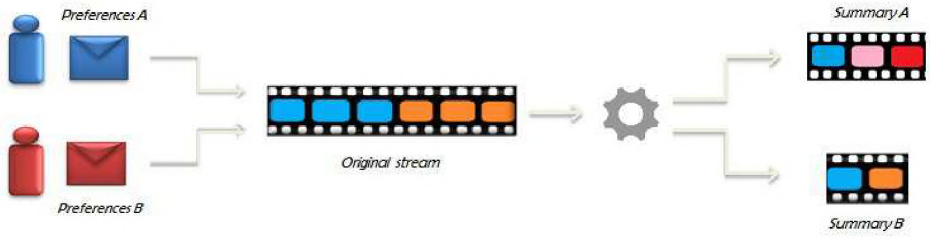

Figure 1-3.: VSUMM personalization scheme example. Users with different interests or needs, want to generate resumes of different videos from the same video. Taken from [49]

.

the difficulty for data acquisition, for which external devices connected to observer users are required, as well as for the generalization of training that would require repeating the experiment on a considerable set of users.

A modern tendency to customize video summaries is to capture the specific intention of the person making the summary. The most natural approach to achieve the above is to inject a query vector into the VSUMM process that reflects the interest of the user who wishes to generate the summary. This vector can be either a set of categories or a plain natural language expression. Xiong [46] propose an approach to video consultation through natural language consultations (see Fig. **1-4**), using a combination of elements detected in the scene (actors, events, locations, objects) and their temporary modeling using hidden Markov models (HMM). The main limitation of this work is that the detection of these elements is by exact coincidence, which implies that it can only be applied in conditions in which the environment in which the video is developed (Disneyland for this work) is known. On the other hand, the processing of the textual query that the user injects is based on the detection of keywords that generate exact coincidence, leaving aside the semantic relations that can happen with the elements of the scene.

A first approach to the use of textual queries and semantic description of them is the one proposed by Shargi [4]. In this work, the authors propose a general scheme for VSUMM in which the segments are selected according to the relevance regarding the input queries (keywords) and their importance for the video context, developing for this purpose a probabilistic model which they call SH-DPP (Sequential and Hierarchical Determinantal Point Process). An important contribution of this work lies in the use of the SentiBank database [52] to build the lexicon of concepts that will later be used to relate the keywords that the user enters, with the segments of video in which they happen. The main limitation of this work is that the authors limit the query to the selection of two names (flowers, cars, etc.) from a previously defined list since the solution lacks the ability to operate with more complex

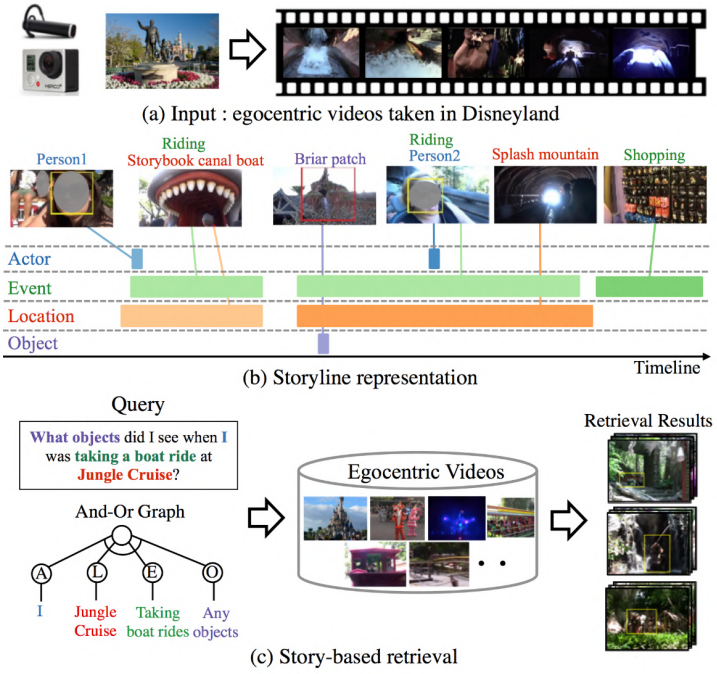

Figure 1-4.: Scheme of the method proposed in [46]. (a) Input video. (b) Story-based representation with 4 elements (actors, events, locations and objects). (c) Linguistic input represented as an and-or graph for a video retrieval task. Taken from [46].

signs.

As far as we know, the first semantic approach to the use of queries in a VSUMM task was proposed by Oosterhuis et al. [5]. For this, the authors make use of linguistic representations, particularly word2vec [28], trained on a corpus obtained from Wikipedia, to represent the words of the query that the user enters. Due to the semantic nature of word2vec, it is possible to obtain indirect relationships between the words of the query and the visual entities that are detected by a classifier in the video frames. For example, a video summary for the query *turkey* could contain scenes related to Christmas or cooking, even if the turkey animal does not appear as a visual entity in the video. This method has the limitations that queries rely completely on the quality of the categorical classifier (see Fig. **1-5**).

A similar approach is proposed by Varini and her colleagues in [49]. In this case, the authors use a VSUMM methodology composed of two stages. In the first stage, they characterize the attention of a group of people who capture a video in a cultural context, using a manual categorization of the video into behavioral states: *paying attention, changing the focus of attention*, among others, and states of movement: *running, standing, walking*, and so on. Later they use a HMM to predict an attention value according to the visual movement information. The authors use this modeling of attention to identify groups of relevant pictures and

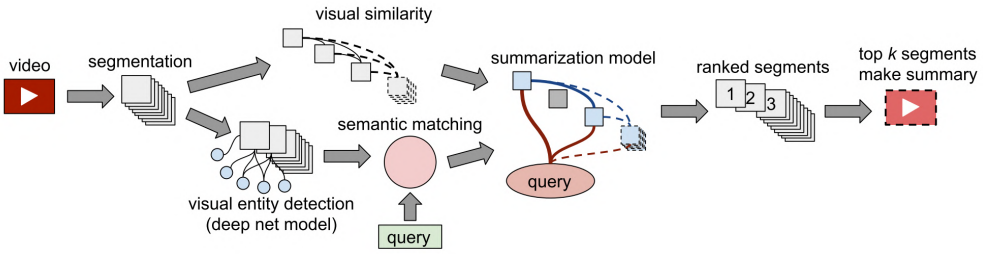

Figure 1-5.: Graphical abstract of method proposed by Oosterhuis et al. Taken from [5]

.

eliminate irrelevant (visual diversity). In the second stage, the authors propose a semantic classification that consists of generating a dataset of related visual concepts from the words in the user-made query and the relationships found in the DBPedia semantic database. Subsequently, this dataset is used to train a BOW classifier of the segments or tables related to the query words. Finally, each segment is weighted according to a linear relationship between word bag classification (BOW) and visual diversity. The main limitation of this work is the need to train on each occasion a classifier according to the query made by the user, which makes it impossible to generalize the VSUMM method and, therefore, its evaluation with respect to other methods. On the other hand, the model of visual diversity evaluated is dependent on the particular interest of the group of users who recorded the videos, so it cannot be considered as a general model.

An extension of the previous work, proposed by the same authors in [53] consists in modifying certain elements of the previously proposed scheme, particularly: 1) the use of word2vec for the representation of the elements that make up the query entered by the user, trained on the basis of DBPedia data, 2) the extraction of the visual diversity from the video by means of visual flow characteristic (Farneback algorithm) and the classification by a 3D-CNN, 3) the visual characterization of the video using the VGG-16 network (Simonyan Zisserman, 2014) pre-trained, 4) the generation of a semantic vector space using the product vectors of the VGG-16 network and the word2vec representation of the query and 5) the use of spatial information obtained by GPS tracking to limit the space of possibilities. It is important to highlight the proposal of a semantic vector space, which allows unifying visual and linguistic information. In the same way, as for the previous work, the authors limit the application of the proposed method to cultural tourism tours.

According to the reviewed works regarding video customization, we found that current approaches try to include linguistic knowledge in the VSUMM process, using textual queries with the aim of capturing the user interest and modifying the video summary according to those interests. Among the most remarkable advantages of using this approach are: 1) the ability to take advantage of the linguistic knowledge of the metadata associated with the video summary, in order to obtain high-level semantic relations, 2) the ability to discern the

context of application and therefore the video processing scheme, prior to its processing, 3) the opportunity to generate different summaries of the same video from different linguistic combinations in the input query, which constitute the interest or context of the observer, and 4) the ability to generate summaries with dynamic context if the query changes over time as the video is processed. Due to the novelty of this approach, it is still possible to make contributions aimed at generating summaries with a higher level of personalization and semantics.

1.2.3. Segmentation and Video Summarization

Many video analysis tasks such as *video retrieval* [32, 37, 54], *video classification* [55] and *video summarization* [56], among others, require an automatic shot (or take) detection to extract segments or sequences of related frames. Data augmentation by data-set unification is another task that requires a shot extraction method. Grauman et al. [15, 12] use the Kernel Temporal Segmentation (*KTS*) method proposed by Potapov et al. [57], to unify videos from different datasets: SumMe [2] and TVSum [31] for training/testing, and OVP [58] and Youtube [58], for data augmentation. Basic approaches for shot extraction consists on doing time-uniform sampling, for example, by considering 1-frame per second [5]. Also it is possible to perform more sophisticated sampling schemata using sliding windows and shifting [59].

Commonly, video shot detection methods are based on local differences between consecutive frames relying on visual features and clustering [30], for example, using RGB and HSV color descriptors as in [60], optical flow [49, 53] or bag of words [61]. Potapov et al. [57] proposed a kernel segmentation method based on the statistical framework *change point detection* which considers features differences between all pairs of video frames, allowing to detect not only shots related with dramatic changes in video but also non-abrupt boundaries between two consecutive frames with different semantic content. Similarly, Lee et al. [7] proposed a segmentation method based on pairwise distance matrix between all frames, and hierarchical clustering with minimum inter-frame distance. Nonetheless, visual features per-se are not sufficient to measure the content of a video frame. It is possible but not necessary that different video frames which have a high similarity in terms of visual features, have a high similarity in terms of categorical features, i.e., that the theme of those video frames are similar. Due to this, it is necessary to explore video representations that allow to consider a more semantic approach that considers a variety of elements such as *objects*, *people*, *actions*, etc.

Lu and Grauman proposed in [30] the use of semantic information in order to extract shots from an egocentric video. In this work, authors classify the activity of the wearer of a head-mounted camera in three categories: *static*, *moving the head*, *in transit*, then group together

consecutive frames with same category.

1.2.4. Deep learning and deep features

Recent advances in deep convolutional neural networks (D-CNN) in conjunction with the creation of datasets of massive images [44, 62], has led to the creation of powerful representations of images and video such as VGG-16 [63] and 3D convolutive networks (3D-CNN) [24]. This concept is called representation learning by Goodfellow et al. in [64] and states that unlike the traditional feature engineering approach that can take decades of work by the scientific community, deep features can be built automatically in minutes, hours or months depending on of the complexity of the problem. Modern neural network architectures have shown high performance in many applications like object detection [65, 66], VSUMM [10] and generation of linguistic descriptions of visual scenes [67, 68].

In computational linguistics and natural language processing, these network architectures have also been used. A powerful application for linguistic analysis consists in obtaining vector representations of words [20, **?**] and sentences [29] known as embeddings, which have the power to preserve the semantic relationships that exist in a linguistic corpus. An analysis and discussion on the use of traditional features, for example, color histograms, GIST, HOG, and dense SIFT, versus deep features in the context of VSUMM, is presented by Zhang et al. [69], in which they demonstrate that the latter has advantages in both precision and performance in the representation of the video.

Word embeddings

Vector representations or word embeddings constitute the main technique in modern natural language processing (NLP) related tasks to represent words using linguistic corpus. This representations allow words to be compared in such a manner that semantic relations can be obtained from using algebraic operations [20]. For example, the distributed representations of word *high* with respect to word *tall* is expected to be closer than from word *small* and this can be expressed as:

$$E_{high} - E_{tall} < E_{high} - E_{small},$$

where E_i is the distributed representation or activation of word i over embbedding matrix E. Another important demonstration of the power of this representations is the capability to perform analogy tasks. For example the analogy *"king is to man as queen is to woman"* can be expressed as:

$$E_{king} + E_{man} \sim E_{queen} + E_{woman}$$

then

$$E_{king} + E_{man} - E_{queen} \sim E_{woman}.$$

Two main word embeddings commonly used in NLP literature are skip-grams known as word2vec [20] and GloVe [**?**], both with similar performance to capture semantic relations and public pretrained embedding matrices, trained using large corpus as Wikipedia. The main difference between GloVe and word2vec consists in that word2vec solves the representation problem as a prediction and GloVe as a co-ocurrence count matrix. The statistical nature of GloVe allows this model to be trained easily parallelized to train over more data, i.e., a large corpus. Using distributed representations for categorical information of input video-frames V_i allows the method to be sensitive to detect consecutive video-frames (V_i, V_{i+1}), with similar information although detected words for both frames could be different, e.g., labels *ball* and *player* in consecutive frames are similar as $E_{ball} - E_{player}$ is small.

1.2.5. Visual-Linguistic representation

Due to the multi-modal nature (linguistic, visual, and temporal) of the data required to perform VSUM based on queries, it is necessary to find representations that allow unifying in some way such data, i.e., representations capable of dealing with information from different sources in a common space. These representations are called joint/coordinated representations. An extensive review in the use of deep learning models for joint and coordinated multi-modal representations is presented by Baltrusaitis and colleagues in [70] (see Fig. **1-6**). They state that joint representations are mainly used for information fusion where data from various modalities are accessible at any moment, different from coordinated models where the intention is to transfer aspects from one modality to another or to enrich each modality with its counterpart. In the latter, only one modality or a subset of the original modalities is accessible in execution time.

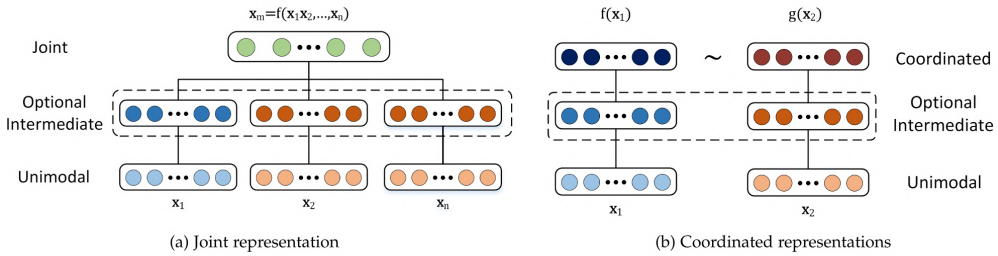

Figure 1-6.: Joint vs coordinated architectures for deep multi-modal representations. Taken from [70].

Many approaches have been addressed in the literature with the purpose of unifying linguistic and visual data. Farhadi et al. [71] propose an intermediate space between the space of images and the space of linguistic sentences, called meaning space, in which each element has different projections in the image and the linguistic space. On the other hand, this space is symmetrical, so given an image, the closest sentence to it can be found and vice versa. The authors represent this space as a triplet $(object, action, scene)$, for example, $(ship, sail, ocean)$, at which images with visual elements or linguistic sentences with syntactic elements close to the triplet, will be projections on the visual and linguistic spaces (see Fig. **1-7**). The main limitation of this approach is that the representation of triplets could be understood as a generic description of a visual or linguistic scene, and therefore can lose interesting relationships that do not happen in that space. On the other hand, the specificity of this representation required the construction of the dataset manually, thus making it difficult to expand this space in future works.

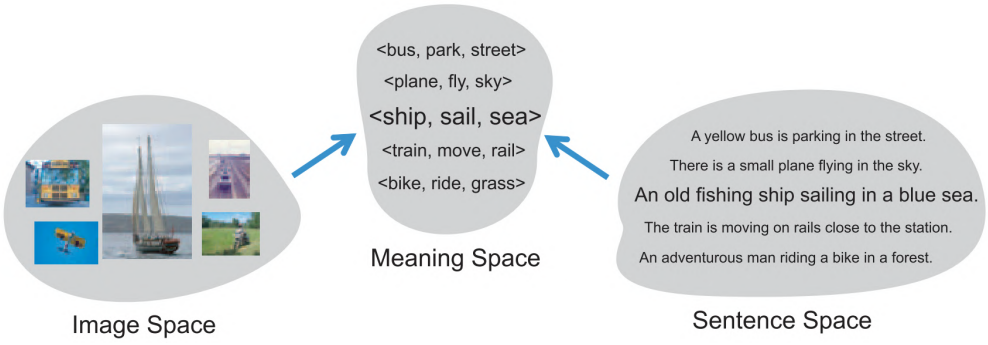

Figure 1-7.: Meaning space example as presented by Farhadi. Taken from [71]

One of the first approaches towards the construction of a joint representation with semantic value is the one proposed by Socher et al. [72]. The authors propose a computational model to perform zero-shot learning through which it is possible to classify instances of previously unobserved categories. For this, the model takes information from a linguistic corpus to conclude the category. First, the model maps the images in a semantic space constructed through the use of visual characteristics and linguistic embeddings and the co-training of these characteristics using a shared-cost function based on the distance lost function L2 (see equation 1-2), in which v_k refers to the visual characteristics of an image k and s_k to the linguistic embedding of a k statement. Once the semantic space has been trained, if the model detects that an input image does not belong to a known visual category (outlier), then it proceeds to assign its category according to the most probable category in the semantic space constructed (see Fig. 8). The main concept behind this computational approach is that it is possible to project phenomena that only happens in a domain A to a domain B, through a shared-cost function $L(A, B)$.

$$L = \sum_{k=1}^{N} (\|v_k - s_k\|_2^2) \tag{1-2}$$

Frome et al. [21] proposed a classification model that uses a measure of similarity between joint representations of images and linguistic labels. For this purpose, they use word embeddings, particularly word2vec, for the representation of linguistic labels and VGG-16 for the representation of images. The authors propose a new loss function called rank loss, which allows better results than the L2 standard used by Socher et al. [72]. Defining the shared loss function according to the problem or application is a crucial point to obtain adequate results.

The construction of joint representations in the case of the video requires processing sequences of frames, in which actions or interactions between the elements on the scene. Lin et al. [54] use a linguistic approach to obtain videos from textual queries made by users. In this case, the authors represent modalities as a semantic graph in which names, verbs, adjectives, and adverbs are identified from the words of the textual query made by the user. Subsequently, by analyzing certain visual elements in the video such as movement and appearance of the objects, a representation of the visual information is obtained, which is finally used to evaluate the correspondence between the user's query and the objects detected in the video. This type of approach has a major limitation in its low power to capture the semantic information for the entire video, due to the location of the characteristics used to describe it.

A computational model to relate the visual content of a video with the content of a book is proposed by Zhu et al. [73], where the authors use joint representations based on skip-thought [29] and deep features extracted from the GoogLeNet architecture [74], for the description of video-frames. An important work on joint representations of video and text is done by Xu et al. [75], in which the authors propose a general and unified framework (see Fig. **1-8**) to create joint representations of videos and linguistic models. This framework validates the use of neural network architectures for this purpose, which outperform widely used methods such as SVM, CRF, and canonical correlation analysis (CCA). The authors mention that, through the representation obtained by this framework, it is possible to 1) generate descriptions in natural language from a video, 2) select videos related to descriptions in natural language, and 3) select descriptions in natural language from a video. Unlike the models of multi-modal representations previously explored, the authors propose adding the temporal analysis of the video through the use of a temporary pyramid pooling (temporal-pyramid pooling) inspired by the work proposed by Wang [23].

Otani et al. [11] proposes an extension to Xu's work [76] by including a recurring network architecture (RNN) in the representation of the texts in natural language and use images

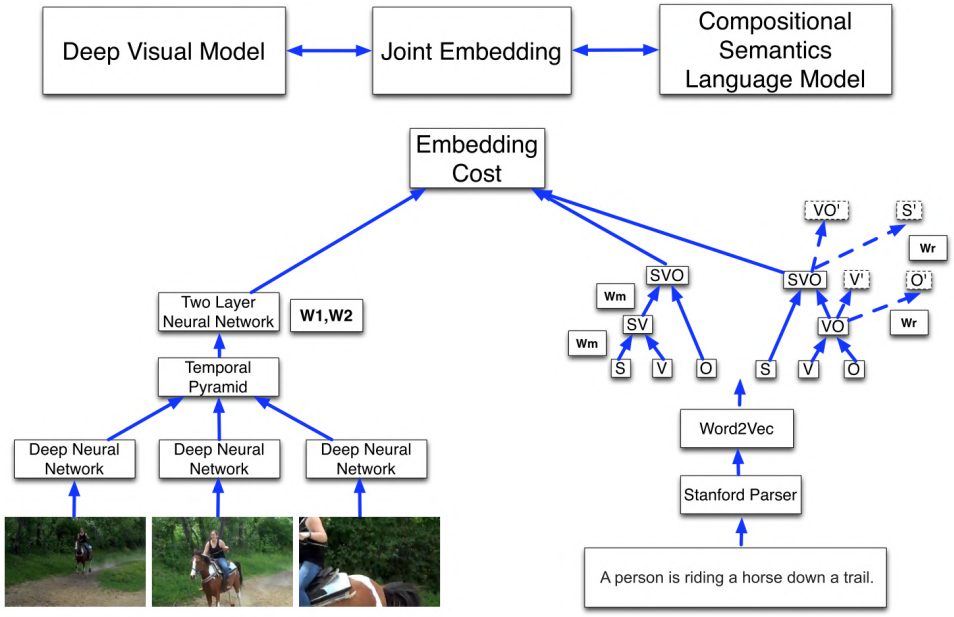

Figure 1-8.: General framework for obtaining multi-modal representations between video and text in natural language proposed by Xu and colleagues. Taken from[75].

related to the video, extracted from queries on the internet using the texts in natural language. Using the previous framework, Otani and colleagues [59] build a VSUMM model which uses a semantic space constructed and joint representations.

The most salient aspects that may be concluded from the review of the literature are described above:

- The concept of representation learning from neural network architectures has advantages in terms of performance, scalability, reuse, and further expansion to other domains.
- The trend in the use of multi-modal representations between visual, temporal, and linguistic information appears to be promising. From this perspective it is possible to project semantic information extracted from the analysis of texts in natural language, on the visual and temporal characteristics of a video.
- In various applications, it is desirable to customize the summary of a video. One of the most natural way for this is to enter a linguistic query that reflects the user's interest (Q-VSUMM).
- Although some Q-VSUMM works have been proposed recently, the novelty of the topic allows the exploration of new models for this purpose.

1.3. The goals of this thesis

General objective

In this research we propose a computational framework for the automatic query-based video summarization task, using a semantic space build by coordinated representations that integrate data from different modalities associated with the video, such as visual features, human-made annotations, and user-made queries.

Specific objectives

- To elaborate a coordinated representation space, from video data and its human-made textual annotations.
- To develop a computational method to obtain representative video segments, using the coordinated representation space.
- To develop a computational method to obtain relevant segments, which allows textual user-made queries.
- To propose a computational framework to obtain interesting and representative segments.

1.4. Summary

In this chapter we have presented a literature review about general VSUMM and more recent branches as Q-VSUMM, and deep learning approaches, which integrates information from a diverse set of modalities and allow to generate a personalized video summary. From the latter, we exposed some general and specific open challenges, and some topics that will be the basis of this thesis, mainly the injection of queries in natural language on a VSUMM scheme using for that purpose a coordinated architecture to transfer properties from the textual to the visual domain and vice-versa.

1.5. Contributions and organization of this work

We proposed a computational VSUMM framework based on deep multi-modal coordination models which rely on modern concepts such as transfer-learning, representation learning and vector words/sentence representations that allows to simplify and unify various VSUMM

objectives: representativeness and uniformity, interestingness by diversity, and importance by query-based personalization for single-concepts and free-form textual inputs. This framework is feasible today due to the advances made by the community, which delivers and shares openly their computational implementations, allowing to collaborate and build knowledge collectively. At the moment of this writing, we have made a publication at the journal IET-Computer Vision (Q1) titled: *Video Summarization by Deep Visual and Categorical Diversity* [77], and currently finishing a second journal paper about the general framework which we have been proposed.

This dissertation is organized as follows: In Chapter 2 we present a general view of the proposed framework in terms of the information flow to generate a video-summary, we discuss how data pre-processing was made, and a briefly description of each stage is presented. In Chapter 3 a coordinated model architecture is experimentally defined a tested over a video-retrieval task, and a video vs textual-query similarity scheme is presented. Representative and uniform VSUMM by hierarchical segmentation and k-medoids is explored in Chapter 4. In Chapter 5 we propose a method to generate relevant video summaries by deep visual and categorical diversity, and also a scheme for concept-query similarity is proposed. Finally, an integrated framework is discussed in Chapter 6.

2. Framework for Query-based VSUMM by Using Visual and Linguistic Information

As previously explored in chapter 1, state of the art in VSUMM personalization, consists in allowing the injection of textual queries from the user, and adjust the final summary according to a hand-crafted measure of matching. Many approaches have been proposed for achieving that purpose. Nonetheless, as far as we explored in literature, there is not a general framework that attends to solve each VSUMM optimization objective and also allows the use of textual queries from users to guide the VSUMM process. Also, most methods use a diverse set of tools to process, describe, and optimize information inside the VSUMM process, which results in complex processing schemes.

Based on the above, in this chapter we present our proposed framework for VSUMM which allows to consider optimization objectives as *representativeness*, *uniformity*, *diversity* and *importance* or user intention by categorical and free-form textual queries. Our framework is simpler than methods in literature due to that it is mainly based on the use of pre-trained models, which simplifies feature engineering, training complexity, and information flow using a small set of models and methods.

In the next sections, we describe in general terms, each processing stage of the proposed framework and the unification process to integrate them; also, we present some general pre-processing steps for video and linguistic information.

2.1. Data-Preprocessing

Most video analysis tasks need to sample the input video according to criteria such as uniform sampling or sampling by shots detection, in order o improve computational performance.

In this work, uniform sampling to **1-frame per second** was used to pre-process the input video, as this is the general approach in the VSUMM literature. Although a shot detection method can be employed in this pre-processing stage, the input video can be arbitrary in

content and length. Also, the desired summary length can be any percentage of the video length, making it unfeasible to determine fixed criteria to guide a shots selection method.

Nonetheless, we use a hierarchical segmentation method in chapter 4 to determine how many representative shots have the input video, but initially, the video was uniformly sampled. Also, a linear resizing algorithm was employed in order to adjust video frames to the required by the visual pre-trained models.

2.2. Pretrained-Deep Models

Many ImageNet pre-trained classification models can be accessed publicly and used as image feature extractors, as explored previously in chapter 1. In table **2-1** it can be observed the most popular DCN architectures for image-net classification challenge. In chapter 5 a detailed evaluation and selection of DCN models using a VSUMM task will be presented, from which we decided to use **InceptionV3** [78] as the main pre-trained visual model of our framework.

For each video frame v_i from the sampling pre-processing, we computed the penultimate (fully connected) and last layer (softmax) activations, as deep visual and categorical features, respectively. To reduce computational time for later stages, we pre-computed deep features and softmax activations for all video datasets and store them.

Table 2-1.: Publicly available deep convolutional architectures for ImageNet [44] dataset. Top-N accuracy refers to the mean number of classes (1000 for image-net) correctly classified in the first N model activations.

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception [79]	88 MB	790	945	22,910,480	126
VGG16 [63]	528 MB	715	901	138,357,544	23
VGG19 [63]	549 MB	727	910	143,667,240	26
ResNet50 [80]	99 MB	759	929	25,636,712	168
InceptionV3 [78]	92 MB	788	944	23,851,784	159
InceptionResNetV2 [81]	215 MB	804	953	55,873,736	572
MobileNet	17 MB	665	871	4,253,864	88
DenseNet121 [82]	33 MB	745	918	8,062,504	121
DenseNet169 [82]	57 MB	759	928	14,307,880	169
DenseNet201 [82]	80 MB	770	933	20,242,984	201

In the case of textual data related to word queries and categorical information from softmax activations (see figure **2-3**), we used **GloVe** 100-dimensional model [27], pre-trained over the Wikipedia dataset. It is important to mention that when working with word-vectors embeddings, it is required first to validate that the word or category exists in the embedding dictionary and perform hyphenation of compound words.

For free-form texts or sentences, we employed skip-thoughts [29] and bi-directional skip-

thoughts models to represent video sentences from a video-to-text (VTT) dataset, and also to measure a degree of similarity between an input sentence and a sequence of frames. This model will be explored in chapter 3.

2.3. Visual-Linguistic Bridge Using Coordination Models

In chapter 3, we design and train a coordination model to construct a numeric space where video and text data can be compared numerically. The basic idea behind this coordination task is that a neural network model can be trained over a set of videos and its related sentences, to map video and text data to a numeric space where, semantically related $(video, text)$ pairs are close, and far on the contrary.

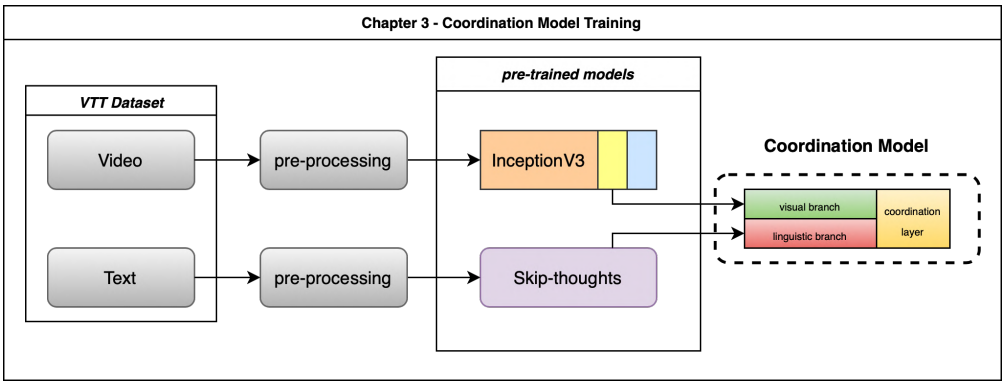

Figure 2-1.: Visual-linguistic coordination model training scheme.

The benefits of using coordination models in image and video tasks are that knowledge and semantics from linguistic data can be transferred to the visual features and vice versa, and can be used later to perform numerous tasks such as video retrieval, video classification and video description among others. In figure **2-1** it can be observed the general stages we used to train the video and text coordination model using a publicly available video-to-text dataset. First, video frames and sentences are described using InceptionV3 and skip-thoughts pre-trained models respectively and entered to the coordination model, which penultimate layers are both the same size, and finally, a coordination loss function is employed to penalize negative $(video, text)$ pairs and rewards positive pairs.

It is important to mention that as a video-to-text dataset contains only positive pairs, i.e., videos and a set of sentences describing them, a negative pairs selection criteria are needed to build the coordination dataset. This criteria is discussed in chapter 3.

2.4. Representativeness and Uniformity by Hierarchical Segmentation and k-medoids over Coordinated Space

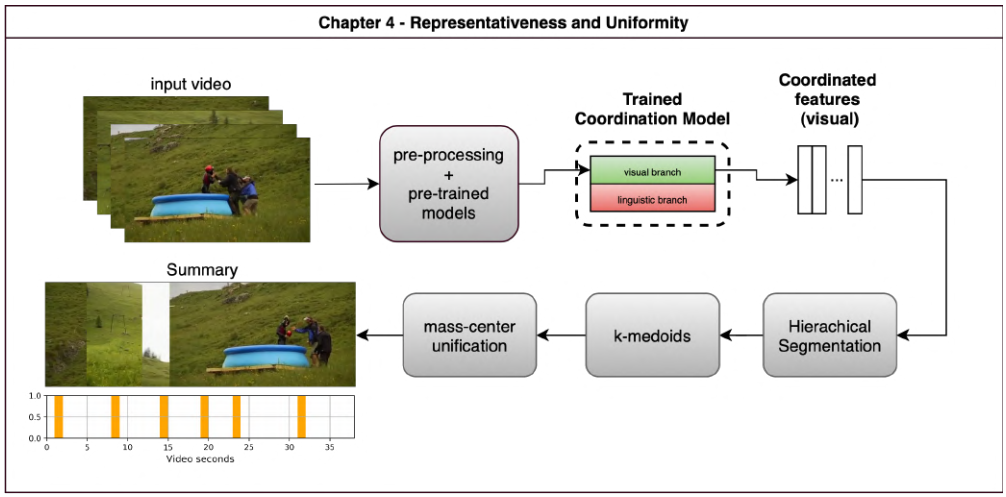

Figure 2-2.: Representative and uniform VSUMM scheme by hierachical segmentation and k-medoids clustering over coordinated space.

Once the coordination model is trained, we use it as a feature extractor by removing its coordination layer and computing the activations from its visual branch for each video frame (see Fig. **2-2**). These activations respond to visual but also to linguistic phenomena as it is discussed in chapter 3 for which we expect to have significant general description power in semantic tasks as VSUMM, compared with only-visual descriptors.

We then apply a clustering approach to extract the most representative frames or the subset of video frames, which minimize the distance from the complete video. Commonly, clustering techniques are parametric, which requires to set the number of clusters to be found. As the input video can be of arbitrary length and content, a fixed number of clusters can not be used. For this reason, we applied a hierarchical segmentation method (see Fig. **2-2**) to automatically estimate the number of segments to be considered by the clustering method.

Finally, a mass center approach is employed to achieve temporal uniformity over the extracted representative segments.

2.5. Interestingness by Visual and Categorical Diversity

As previously discussed in chapter 1, interestingness understood as the local importance of a video is commonly approached as a regression or ranking task, which takes into account exclusively visual features. In chapter 5 a method to estimate a score of interestingness given an input video, not only visual but categorical diversity [77] will be presented.

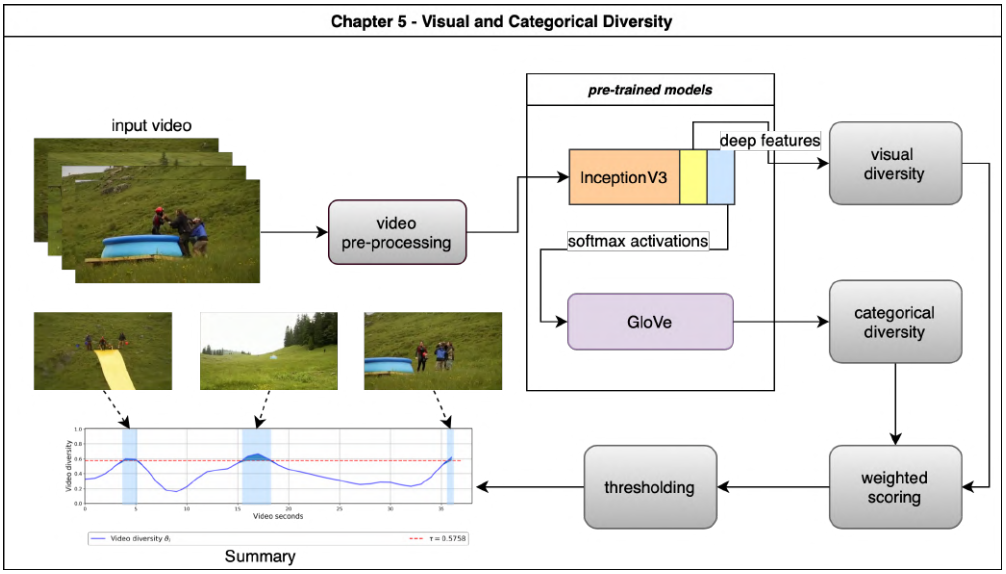

Figure 2-3.: Interesting VSUMM by visual and categorical diversity using pre-trained visual models and word-embeddings.

In order to obtain the visual diversity of an input video, we used deep-features from the pre-trained InceptionV3 model as a visual descriptor and a differential scheme to get a highly diverse sequence of video frames. In the case of categorical features, we compute the softmax activations from the pre-trained model for each video frame, to get the categories with higher probabilities according to an user-defined threshold. Then, each category is mapped into a numeric space using a pre-trained GloVe model and weighted-averaged to obtain a single vector per frame. Finally, a differential scheme is applied to obtain the categorical diversity. The main advantage of using a word-embedding model such as GloVe in this task is that similarity/distance between frames does not only consider exact categorical matching but semantics between categories.

The main idea behind our proposed method is that homogeneous sequences of frames in terms of its visual features and also its content (categories) are considered not relevant, and interesting on the contrary, but the content is not measurable by exact matching but by linguistic similarity, which can be exploited using word-embeddings. A graphical depict of chapter 5 can be observed in figure **2-3**.

Finally, the coordination model was not employed for the interestingness VSUMM objective, because as coordination model maps linguistic and visual features in a common space, it is expected to be used for high generalization tasks such as video retrieval and video description, but not for high discriminatory tasks such as category detection, or in this case local visual and categorical diversity.

2.6. Importance by Query Injection over Categorical and Coordinated spaces

As discussed in chapter 1, importance objective can be approached by including the user intentions to guide the VSUMM task. User intentions can be injected in the form of textual queries, which will be used later to measure a degree of similarity between video segments and user queries. For this purpose, we developed two strategies to represent the user-queries and use them for importance scoring.

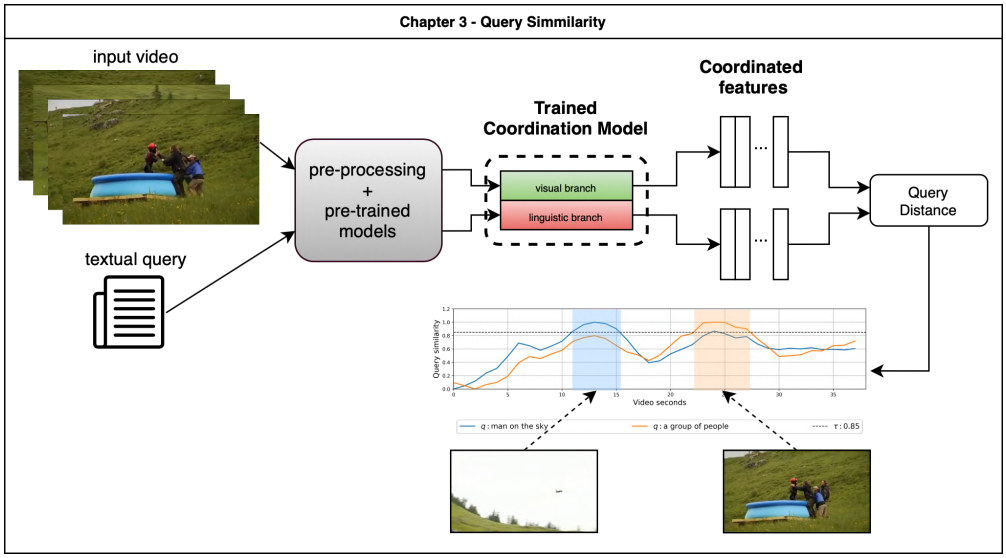

Figure 2-4.: Video and query sentence similarity using coordination spaces.

Employing a coordination model trained using a VTT dataset, given an input video and a query in the form of a sentence, we first compute the coordinated activation from the linguistic branch for the input sentence, then we use a fixed-width moving window on the video to compute coordinated visual features, and finally compute a degree of similarity between both features using a distance function. Then, using threshold criteria, we obtained the more important segments concerning the input sentence. This process will be further detailed in chapter 3. In figure **2-4**, it is shown the similarity response between an input video and two sentences using this approach.

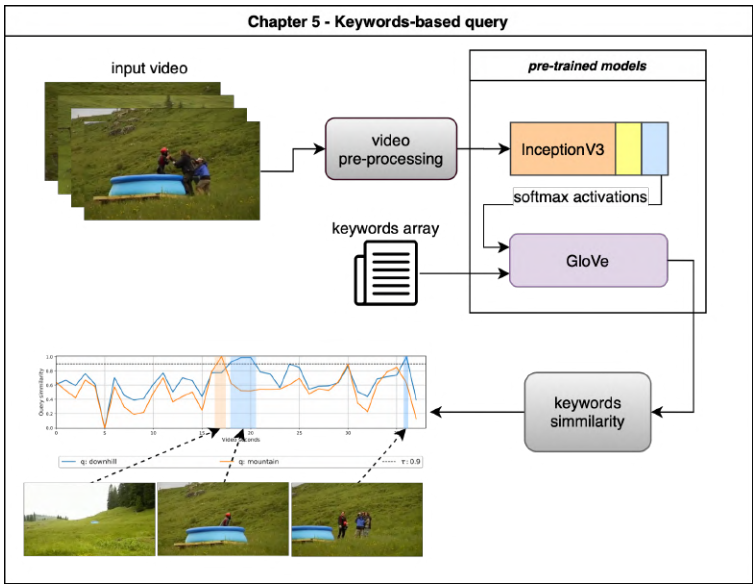

Figure 2-5.: Video and key-words similarity using pre-trained deep visual models and word-embeddings.

User intentions can also be represented as a list of words in the form $q = [word_0, word_1, ..., word_n]$ generally containing categories, but can also contain verbs, nouns or adjectives. This situation makes difficult the representation of q, for which many approaches have been explored, such as ontologies and semantic trees, among others, as it was discussed in chapter 1.

Our approach relies on the use of a pre-trained word-embedding that is used to transform q in a matrix of word-vectors, which allows us to apply a distance function between the categorical softmax activations from InceptionV3 model and q. In figure **2-5** it is shown the described method. This approach is discussed and analyzed in chapter 5.

2.7. Integration

Once all VSUMM objectives are computed, we developed a knapsack problem optimization approach to obtain a single summary that considers one or all objectives. This approach allows us to generate multi-objective video summaries in a simple manner by using a set β parameters to weight each objective and configure how each one impacts the final summary. This integration scheme will be further explored in chapter 6. In figure **2-6**, a graphical depiction of this integration is shown.

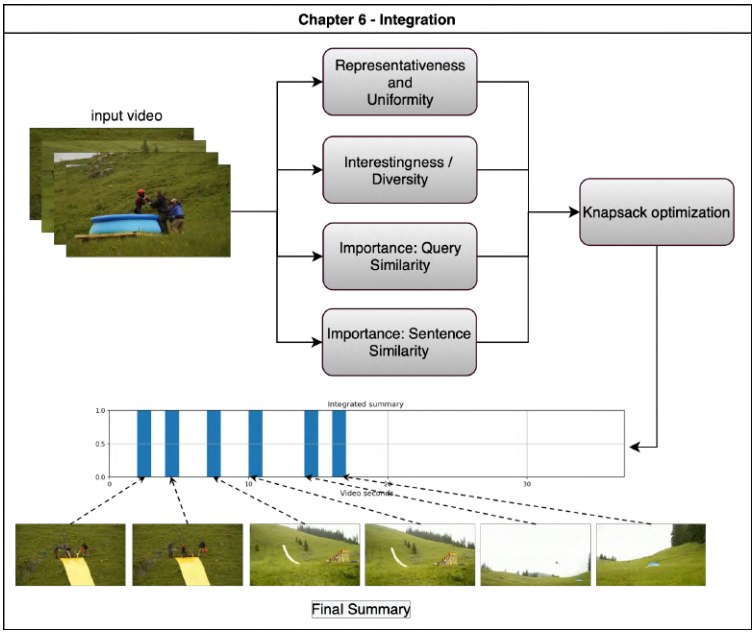

Figure 2-6.: Integration of multi-objectives VSUMM using a knapsack problem approach..

2.8. Chapter summary

In this chapter we have presented the global elements of our proposed computational framework to perform a query-based VSUMM task solving multiple VSUMM objectives such as **representativeness** and **uniformity** by hierarchical clustering and k-medoids over a visual-linguistic coordination space, **interestingness** by visual and categorical diversity using pre-trained visual and linguistic models, and **importance** by video/query similarity using free-form sentences and categorical keywords queries. All of the presented elements will be further analyzed and discussed in the next chapters.

3. Visual-linguistic space construction

In this chapter, we explore and develop deep learning approaches to construct an n-dimensional space where videos and text co-exist, in order to represent a video by what it contains in visual terms, and what it means extracted from text descriptions. The main idea behind this approach is to **enrich** visual information from the linguistic domain. Since summarization is a semantic task, we exploit this fact to represent a video using a semantic space in which co-exist visual, categorical, and linguistic information.

Video analysis and consequently, video summarization is a multi-modal task, in which we have multiple sources of information of different nature. These sources can be images (*video frames*), audio, text(*captions*, *annotations*), GPS data, etc., and constitute different modalities from the same phenomena as defined by Lahat et. al. in [83]. These modalities can be understood as complementary information, which, once fused, represents the whole. This complementarity or added value is known as diversity, as explained in [83].

> "*Diversity allows to reduce the number of degrees of freedom in the system by providing constraints that enhance uniqueness, interpretability, robustness, performance and other desired properties...*"[83].

In this chapter, we explore the use of deep multi-modal coordination models architectures and its possibilities to enrich video descriptions knowledge transferring from textual to visual domains and vice-versa.

3.1. Related work

Multi-modal fusion applied to video summarization

The main approach to VSUMM with multi-modal representation involves training a coordinated space that can be used later to describe video frames. As explained before, this approach is more convenient because all modalities may not be available in the production stage; that is, not all videos contain descriptions to generate a video summary.

Various approaches have been taken to use multi-modal representations for VSUMM. Plummer et al. in [47] trained a coordinated representation between a pre-trained ResNet architecture for video-frame representation and Hybrid Gaussian-Laplacian Mixture Model (HGLMM) features for text representation. Nonetheless, authors use a text-to-image dataset (Flickr30k), having the limitation that trained coordinated space lacks temporality of the video. Otani et al. [59] use a pre-trained DCN model (VGG16) for video frame description and skip-thoughts [29] sequence-to-vector model to represent text, and train a coordinated semantic space. The main limitation of this work is the averaged nature of the video representation, which does not take into account temporality. Yuan et al. proposed a coordinated representation based on a pair of autoencoders. Video frames are represented using AlexNet pre-trained architecture, and text is represented using skip-thoughts. In this work, the authors also processed video in a per-frame manner.

3.2. Model

In order to train a coordinated model that allows us to construct a latent space where video and language co-exists, we assume we have a dataset composed of a set of videos V, each annotated with a set of sentences S in natural language. Also, a set of architectures and an experimental framework are necessary by which we can test them to select by some criteria the best coordination model constrained to the dataset size and quality.

Generally, training an end-to-end model for a given task will lead to a better result than using pre-trained models. We assume that powerful classification models and linguistic embeddings will be useful for the general objective of this work, that is, be able to inject queries on a VSUMM task. Also, in a real-world application, for example, for a small business to apply this model, training stages should be on the low-cost side of technological possibilities.

In this section, we discuss, evaluate, and select a coordination model to obtain a video representation which also considers linguistic information. For this, we discuss: *a*) the input representation or features extraction for both, video and text, *b*) the neural network-based architectures for the coordination model, *c*) the alignment function used to train the architectures, and *d*) a discussion about how to select negative pairs considering the dataset nature.

3.2.1. Input representation

In this section, we discuss the strategies used to represent videos and sentences from a video-to-text dataset using pre-trained models following a similar approach as one presented in chapter 5.

Video representation

As previously explained in chapter 2, we rely on pre-trained DCN models for visual feature extraction, due to the following reasons:

- Allows to accelerate the training process using the trained latent space from the image-net classification challenge.
- As image-net considers 1000 categories organized according to word-net hierarchies [44], it has a linguistic nature that can be used in a query injection scheme.
- Latent space from image-net DCN models condense visual and categorical information which we can take advantage of for every video-to-text task

From the above, we treat a video $v_i \in V$ as a sequence of n frames $v_i = v_i^0, v_i^1, v_i^2, ..., v_i^n$ and extract the 2048-dimensional latent space from the fully connected layer at depth $nl - 1$ of a deep model M.

Sentence representation

Similarly to video representation, we could consider a sentence as a sequence of words and compute an encoded set of words using a pre-trained word-embedding as GloVe or word2vec, as in chapter 5. Nonetheless, order plays an essential role in the significance of a word inside a sentence, so it is necessary to encode sentences using recurrent models that treat sequences as a series of words. Karpathy, in [84] used a bidirectional LSTM approach to encode sentences for an image-to-text task. The main limitation of this approach is that due to the limited vocabulary in the sentences from an x-to-text dataset, a sentence embedding trained over it, would not be general enough to represent linguistic relations that can happen in arbitrary videos or queries in a VSUMM task.

From the above we used an implementation of a pre-trained skip-thoughts model proposed by Kiros et. al. in [29], which encodes sentences of a fixed max number of words into a 4800-dimensional vector with an RNN inspired in the skip-gram model [28], and trained with the BookCorpus Dataset [73], composed of $11,038$ books from the web and about 74 millions of sentences.

3.2.2. Architecture

We explore two architectures for video-to-text coordination, which mainly differ in how they receive the deep video features, as text representation using skip-thoughts model is static, i.e., a sentence is always represented as a vector. The first architecture treats video as the

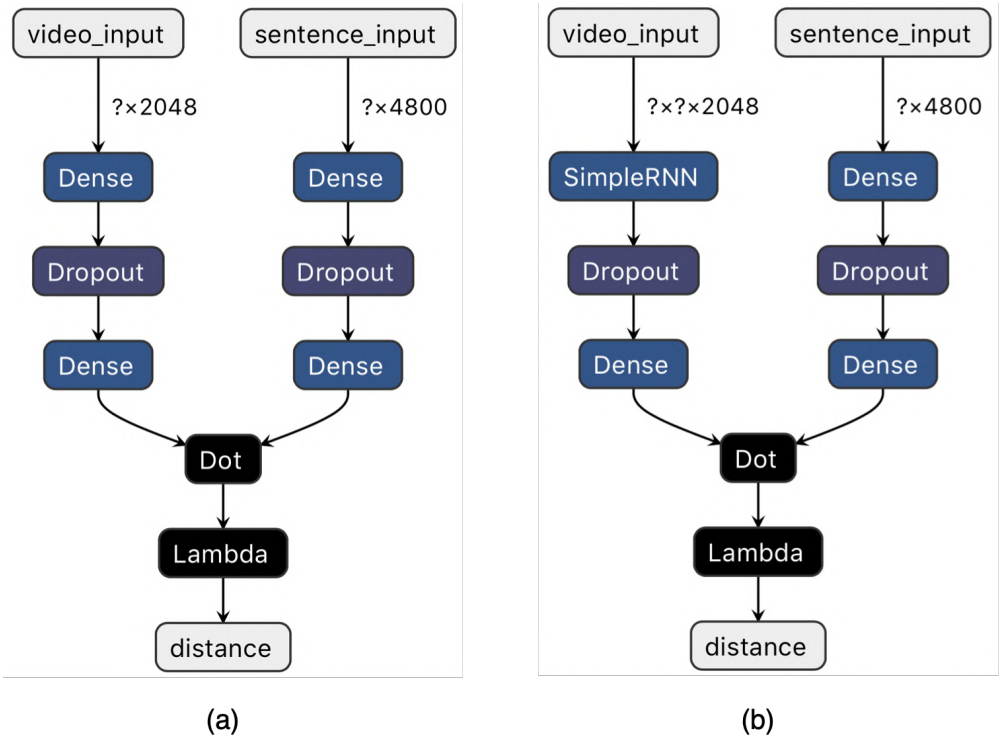

Figure 3-1.: **Coordination architectures**. Averaged (a) and recurrent (b) architectures used for video-to-text coordination problem.

average of deep features from the pre-trained model, and the second architecture as a sequence of deep features using a recurrent approach.

As it can be seen in figure **3-1**, both architectures are composed of two branches, one for the video and one for the sentence. Each branch is a stack of two hidden layers for computing the coordinated space, and both branches are coordinated by a lambda layer, which computes the cosine distance between the second hidden layer from each branch. The first hidden layer from both branches can have a different number of neurons; on the contrary, the second hidden layers are required to have the same number of neurons as it is needed to compute a distance metric between both.

Averaged model

Under some constraints, it is possible to assume that the averaged response $\overline{V}$ from a neural network M for a set of video-frames V will be close to the set itself, i.e., V will have low variance in the latent feature space. Some restrictions has to be considered in order to assume the latter:

- The feature space used to represent the set of video frames can not be exclusively visual. In the case of images, the average response will correspond to blurred information and will not represent the actual content of the video. On the contrary, a vector representation of the video frames extracted from a pre-trained DCN model M supports arithmetic operations as they encode semantic structure from data that was used to train the model, for example, word-net semantic structure [44] in the case of image-net models.
- The video length should be as short as possible in order to ensure the averaged response to be homogeneous, that is, long videos will probably have a higher visual variance, for which an averaged response from a DCN model will be close to any other video.

In figure **3-1**(a) it is shown the averaged model architecture. Notice that the video branch receives a single vector of dimension 2048, as all video frames features extracted from a previous DCN model M are averaged.

Recurrent model

Generally, actions and interactions between categories occurring in a video frame sequence can not be modeled using an averaged approach, as the order in which they appear is essential for the video understanding. From the previous, it is possible to use a recurrent layer to process the video frames as sequences of features before the coordination layer. In this case, we replaced the first hidden layer from a fully-connected structure to a recurrent one (see figure **3-1** (b)). Notice that the received sequence of video frames is not constrained to a fixed length, but arbitrary and dependant of video length.

3.2.3. Coordination / Alignment function

Suppose we have a set of encoded videos $V = \{v_0, v_1, v_2, ..., v_n\}$ annotated with a set of encoded sentences $S = \{s_0, s_1, s_2, ..., s_n\}$, for which each pair (v_i, s_i) can be positive if v_i corresponds with the description s_i, and negative on the contrary. In order to coordinate V and S, we need to find a common space Υ where $|\Upsilon(v_i) - \Upsilon(t_i)| \simeq 0$. A basic approach to solve this problem using a deep learning strategy, consists in using a L2 distance loss function (equation 3-1) [84]:

$$L = \sum_{k=1}^{n} \parallel \Upsilon(v_i) - \Upsilon(s_i) \parallel_2^2 \tag{3-1}$$

The main problem with this loss function is that a trivial solution exists where $\Upsilon(x) = 0$,

that is, coordinated space Υ converge to 0. Then, it is necessary to employ loss functions that consider positive and negative pairs, i.e., margin-based loss functions, which enforce positive pairs to be inside a margin α and negative pairs outside it.

$$L = \sum_{i=1}^{n} \left[\| \Upsilon(x_i^a) - \Upsilon(x_i^p) \|_2^2 - \| \Upsilon(x_i^a) - \Upsilon(x_i^n) \|_2^2 + \alpha \right] \tag{3-2}$$

Triplet loss, proposed by Schroff et al. in [85], is an alignment function commonly used to train Siamese networks for binary recognition problems, that is, problems where it is needed to validate if two objects from the same modality are equal. A typical application for these networks is face recognition.

In equation 3-2 it can be observed the triplet loss, where x_i^a is an anchor or reference, x_i^p is a positive example and x_i^n is a negative example. Finally, α is an arbitrary margin value to enforce positive examples to be inside it and negative examples to be outside it. As authors mention, in order to ensure fast convergence and avoid bad local minima during the training stage, it is needed the selection of special samples known as semi-hard positive and semi-hard negative such that satisfy $\| f(x_i^a) - f(x_i^p) \|_2^2 < \| f(x_i^a) - f(x_i^n) \|_2^2$, that is, positive examples with distances from anchor are less than negative examples distances from the anchor, but not too much.

Positive and negative examples are precisely defined for problems where a single modality is used and where there are multiples samples per object, e.g., face recognition or voice recognition. Nonetheless, this could be ambiguous in cases where the meaning of objects is considered, as video and text relation. Although it is possible to consider $x_i^a = v_i$ and $x_i^p = s_i$, in the case of video-to-text a negative pair (v_i, s_i) does not assure a semi-hard negative case, i.e. although a video v_i and a sentence s_i are not paired, they could have a similar content. Plummer et. al. [47] used triplet loss for an image-to-text coordination problem over Flickr30k dataset. Although authors do not explain how semi-hard positive and semi-hard negative samples were selected, it is possible to assume that for the case of a video-to-text problem, the complexity of the pairing task could be unfeasible.

$$L = \sum_{i=1}^{n} t_i D(\Upsilon(v_i), \Upsilon(s_i)) + (1 - t_i) max(0, \alpha - D(\Upsilon(v_i), \Upsilon(s_i))) \tag{3-3}$$

$$L = \sum_{i=1}^{n} \frac{1}{2}(t_i)\{D(\Upsilon(v_i), \Upsilon(s_i))\}^2 + \frac{1}{2}(1 - t_i)\{max(0, \alpha - D(\Upsilon(v_i), \Upsilon(s_i)))\}^2 \tag{3-4}$$

Hadsell et. al. [86] proposed the contrastive loss in its general and exact forms (see equations 3-3 and 3-4), for a dimensionality reduction problem, learning an invariant mapping function

can transform data into a low-dimensional manifold through a neural network, using prior knowledge. This problem is similar to the Siamese neural network training for face recognition but applied to coordinate a common space where similar objects (from the same modality) are close, and far on the contrary. This loss function has the advantage over the triplet-loss in the use of the label t_i, which simplifies the pair selection task in the dataset generation stage. Also, it allows us to train the model using a binary classification scheme and generalizes the use of a distance function D, which makes it possible to test different measures of distance if it is necessary. In equations 3-3 and 3-4 it can be observed the general and exact contrastive losses, where $t_i = 1$ if (v_i, s_i) is a positive pair, and $t_i = 0$ on the contrary, D is a measure of distance, e.g., Euclidean or cosine distance, and α is an arbitrary margin value. A better understanding of the behavior of general and exact contrastive loss can be observed in figure **3-2**. Notice how loss for negative pairs is zero in the range $[\alpha, max(distance)]$, and on the contrary, tends to zero in the range $[\alpha, 0]$ for positive pairs.

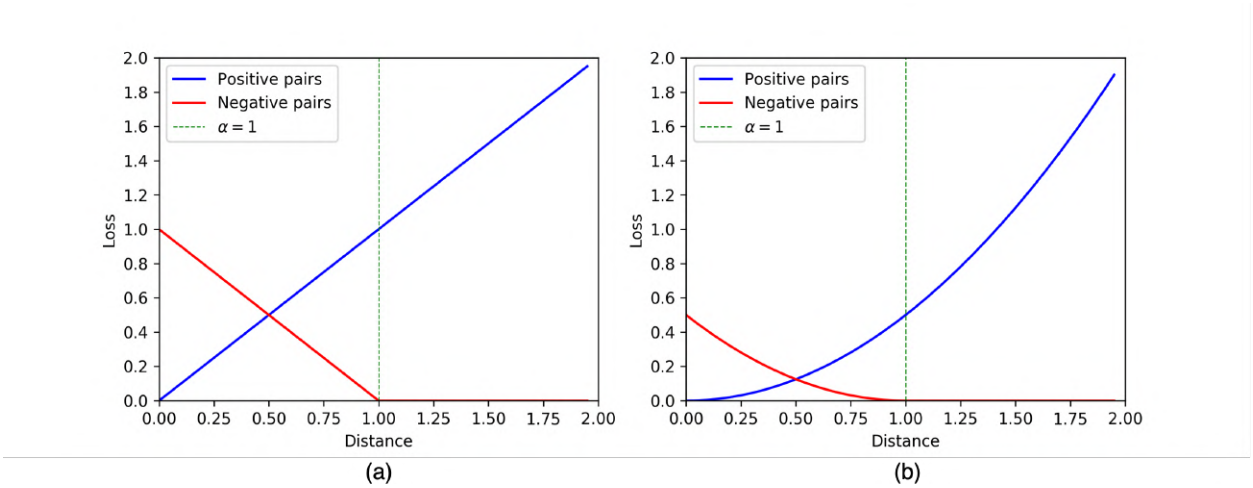

Figure 3-2.: **Contrastive Loss**. General (a) and exact form (b).

Otani et. al. [59] used general contrastive loss for coordinating video and text in a latent semantic space, assigning α the max Euclidean distance between positive examples before training to force negative pairs to be outside the worst positive scenario. Every video and its user-generated descriptions from a VTT dataset was taken as positive pairs, and 20 random descriptions were used as negative pairs for each video. Although authors assumed a random selection could be used to generate negative pairs, we consider this approach could generate a high number of soft-negative pairs, since a video v_i and a sentence s_j although not paired, could be close, which eventually could lead the training to converge to a bad local minimum. From the above, we consider it is necessary to complement the random selection of negative pairs using distance criteria.

Negative pairs selection

It is possible to assume that for a given set S of sentences paired with a video v_i, the distances between them (positive pairs) should be closer than distances computed between S and sentences from random videos. From the above, we can find a threshold of distance τ for which a random sentence pair (s_i, s_j) could be considered at least, a semi-soft negative pair if $D(s_i, s_j) \geq \tau$. In the case of positive pairs, the same approach could be used in order to find a threshold that helps to select semi-soft positive pairs. Nonetheless, we consider that as humans annotators performed the sentence generation task, it is essential to include any possible deviation coming from positive pairs.

Using a statistic approach, percentiles 0 and 100 of distances between positive and negative sentence pairs, are equivalent to a minimum and maximum distances for both cases. In other words, positive and negative percentile 0 can be considered as soft positive cases and hard negative cases; on the contrary, positive and negative percentile 100 can be considered as hard positive cases and soft negative cases, respectively. From the above, in order to generate negative samples that can be differentiated from positive pairs allowing to diminishing intersection between classes, but not so separated that do not represent general cases, we need a semi-soft negative, determining a threshold τ such that $P_0^{neg} \gg \tau < P_{100}^{neg}$.

3.3. Dataset

Table 3-1.: Public datasets for video-to-text task.

	Features		
Dataset	**Number of videos**	**Annotations per video**	**Video length (approx.)**
MSR-VTT [9]	10K	20	5 minutes
TRECVID-VTT [8]	1.9K	2	5 seconds

Once modalities A and B to be coordinated are identified, it is selected a dataset that contains those, commonly an $A \rightarrow B$ dataset, for example, for $image \rightarrow text$ coordination task, it can be used an image-to-text dataset [9, 87, 67]. In this case, we selected a public video-to-text dataset generally used for video-retrieval and video-captioning tasks. MSR-VTT dataset [9] and TRECVID-VTT [8] are well known public video-to-text datasets. Both contain videos associated with sentences written by human annotators. In table **3-1**, it can be observed the main features of each dataset, that is, the number of videos they contain, the number of sentence descriptions made by human annotators, and the approximated video length.

As discussed previously in section 3.2.2, for this work, it is desired videos of short length,

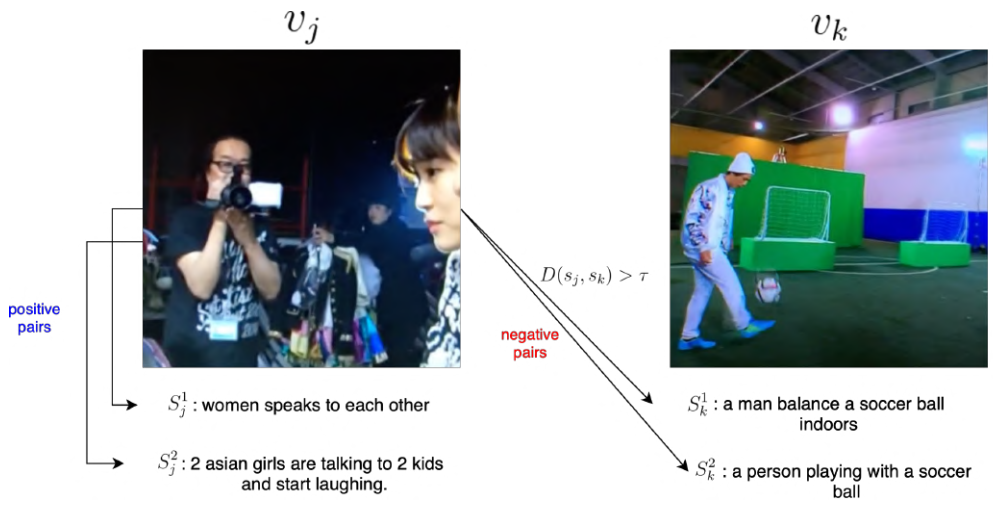

Figure 3-3.: **Pairs generation scheme**. Example of videos and their sentences from dataset TRECVID-VTT. Each video v_j and its paired sentence s_j are considered a positive pair for which $y = 1$. On the contrary, any video v_j and a random sentence s_k are considered a negative pair if $D(s_j, s_k) > \tau$ for which $y = 0$.

whose sentences describe them concisely to use the coordinated space subsequently as a linguistic-enriched video descriptor. From the latter, it was used the TRECVID-VTT dataset to train and validate the coordination models in the experimental framework.

In order to use the contrastive loss alignment function discussed in section 3.2.3, the dataset needs to be expressed in the form $\{x : (v_i, s_i), y : 1\}$ for positive pairs and $\{x : (v_i, s_k), y : 0\}$ on the contrary (see figure **3-3**). For this purpose, two positive pairs per video were obtained as each one is annotated with two sentences (see table **3-1**). For the negative pairs, first it was computed the percentiles of cosine distances between random sentences (see table **3-2**) following the selection criteria discussed in section 3.2.3 in order to compute τ, which we determined experimentally as percentile 75 of negative pairs distances, i.e., $\tau = 0.5477$, and finally, random pairs (v_i, s_k) such that $D(s_i, s_k) > \tau$ were used as negative samples.

Table 3-2.: Percentiles for Cosine distances between positive and negative (random) sentence pairs from dataset TRECVID 2017.

	Percentile				
	0	**25**	**50**	**75**	**100**
Positive	0.0000	0.3327	0.4011	0.4610	0.6867
Negative	0.1686	0.4381	0.4934	0.5477	0.7545

3.4. Experiments

As coordination models are used mainly for retrieval tasks, quantitative evaluation is performed using retrieval-based metrics such as Recall@k or median rank [84, 54, 88, 89]. On the other hand, works that use coordination models for video analysis, generally report only qualitative metrics to evaluate the quality of coordination architectures [11, 47, 90] and not much information about data processing and quantitative analysis is presented by authors. Although Recall@k and median metrics rank are not entirely suitable for the coordination problem in this work, we consider using a quantitative evaluation in order to have selection criteria to choose the best model.

In the following experiments, we perform a grid search over a set of meta-parameters associated with the architectures and the loss function. Next, we select the best two models from each general architecture (see section 3.2.2) using classification-based criteria. Finally, we evaluate both models using retrieval-based metrics as well as their qualitative performance over the test set.

Data preprocessing

We sample each video from the TRECVID-VTT dataset to 1-frame per second and compute its vector representation from InceptionV3 DCN-model and the vectors for the associated sentence pairs using the skip-thoughts model (see section 3.2.1). For the averaged architecture, we transform the video into a single 2048-dimensional vector by averaging. For the recurrent architecture, we use the complete sequence of vectors with dimensions $2048 \times n$ where n is the video length in seconds. Each input (video and sentence) is standardized using general formula $x = \frac{x-\mu}{\sigma}$ in order to have mean zero and unit variance in the data. Finally, to each video and sentence pair (v_i, s_i) is assigned a label $y = 1$ if is a positive pair and $y = 0$ on the contrary.

Grid search

Table 3-3.: Fixed and iterable parameters for grid search.

	Parameter	Value
fixed	η	1e-6
	epochs	1000
	batch size	256
iterable	activation function	$[tanh, relu, mish]$
	units	$[512, 1024]$
	α	$[0.8, 0.9, 1.0]$
	drop rate	$[0.2, 0.3]$

In order to perform an exhaustive grid search over coordination models, we fixed some meta-parameters, as it is shown in table **3-3**. The *learning rate*(η) was fixed to a low value to avoid fast convergence on local-minima, but that did not require too many epochs. The number of *epochs* was fixed to a high value with early-stopping to avoid over-fitting. The *batch size* was fixed to an intermediate value between 64 and 512, which balances between high generalization and computing time. Iterable parameters (see table **3-3**) can be described as follows:

- *activation function*: This parameter corresponds to the non-linear function used in the first hidden-layer of each branch (see figure **3-1**. The second hidden-layer is linear for both branches, as non-linear mapping between modalities it is performed in previous layers. We used the most commonly used activations functions, as reported in coordination models works.
- *units*: The number of neurons of the first hidden layer. The second hidden-layer is always 512 in order to compute a distance between both branches, as explained in section 3.2.2.
- α: Separation margin from exact contrastive loss function explained in section 3.2.3 - equation 3-4.
- *drop rate*: Probability of dropping neurons in first hidden layers from both branches of coordination models using a dropout regularization.

Quantitative evaluation

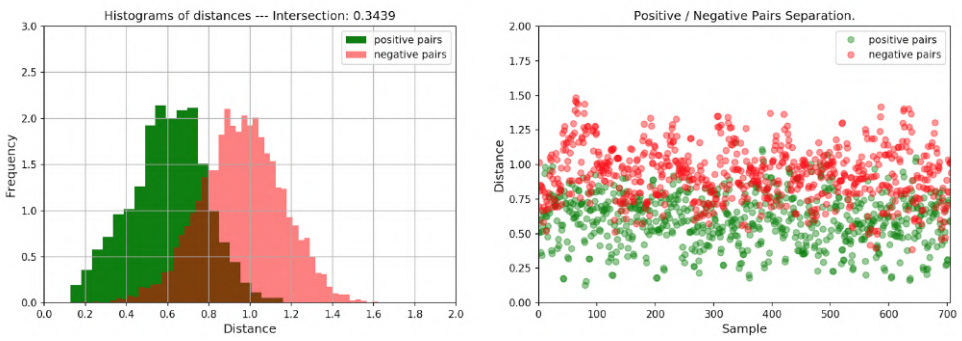

Figure 3-4.: **Example of a coordination model**. Retrieved distances for positive and negative pairs from a trained coordination model. Left) Discrete histogram of distances between pairs. Right) Positive and negative samples vs distance.

It is possible to consider a coordination model as a classifier, where distance D in coordinated space between a given video and a sentence pair is related to the probability the pair is

considered a negative case. Figure **3-4** shows the distances from positive and negative pairs for a coordination model from grid search exploration. Notice that the intersection between histograms of distances is related to the discrimination power of the model, in the sense that a good model is expected to have zero intersection in the distance domain but also a high inter-class separability, i.e., the histograms cover a wide distance range. From the latter, we can evaluate our model's discrimination power using a classification approach, e.g., using a ROC curve analysis.

Table 3-4.: Grid search for averaged and recurrent coordination models. Best area under the curve (auc) from averaged and recurrent architectures are shown in bold.

activation	units	alpha	drop_rate	auc (averaged model)	auc (recurrent model)
relu	1024	0.8	0.3	**0.930429**	0.647643
mish	1024	0.9	0.2	0.918503	**0.915619**

In table **3-4**, the models with the highest auc for averaged and recurrent architectures can be observed. Complete evaluation by grid search appears in table **A-1**. In general terms, recurrent architecture reached overfitting in fewer epochs than the averaged, which led to values auc < 0.6 in some cases due to a low number of training epochs. A visual evaluation of the discriminatory power of best models can be made from figure **3-5**. Notice that the averaged model gets a lower intersection and higher AUC v1alues than the recurrent model (bottom).

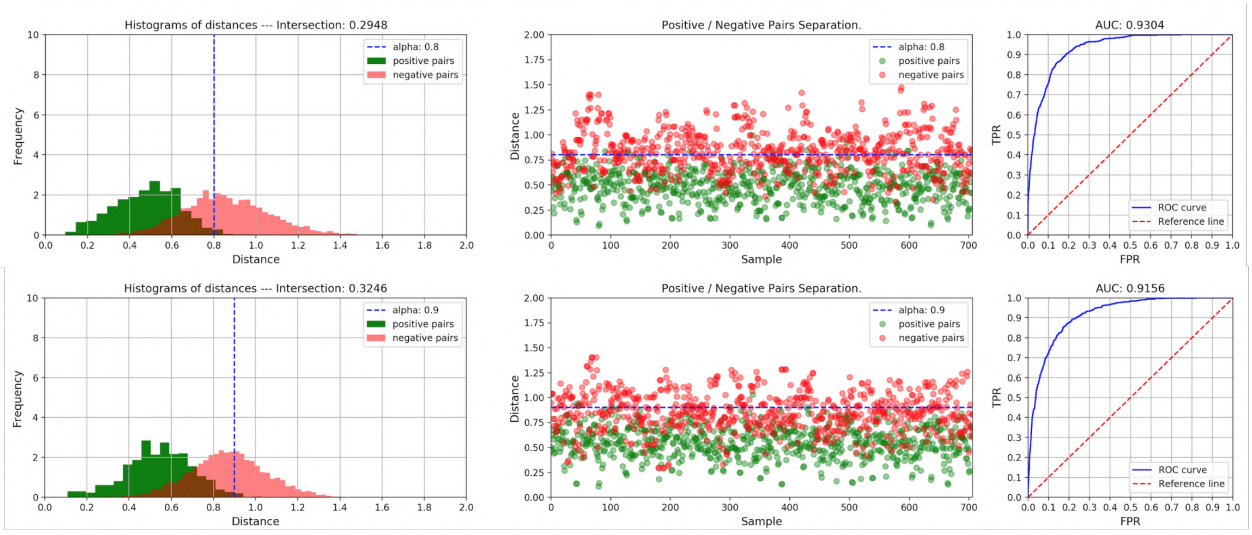

Figure 3-5.: **Intersection and auc.** Retrieved distances for validation test from last training epoch. Top) Averaged model. Bottom) Recurrent model.

Ranking metrics. Since coordinated models are commonly used for retrieval tasks, such as image/video annotation and image/video search, we expect that for a given object x_i, its

Table 3-5.: Ranking metrics. Recall@K (high is good) and med$r$ (low is good) metrics for the video annotation task using averaged and recurrent models.

Model	R@1	R@5	R@10	Med$r$
Averaged	**5.95**	**17.28**	**28.61**	**29**
Recurrent	2.55	11.33	23.51	33

associated pair y_i appears in the first position using a score. In other words, we expect its positive pair $y_i \in Y$ gives the best possible score.

Recall at K ($R@k$) and median rank (mean$r$) are metrics used in most multimodal retrieval works [84, 91, 92, 21] based on the latter. $R@K$ measures the percentage of times the ground-truth pair of a given object appears in the first K elements in a previously ordered array of distances. Similarly, mean$r$ measures the median position at which the positive pair appears for all objects in a dataset. For mean$r$ best and worse cases are 1 (first position) and the length of the dataset (last position), respectively. In table **3-5** it is shown the Recall@K and med$r$ for averaged and recurrent models in the sentence retrieval task. Averaged model obtains a better Recall@k and med$r$ metrics than the recurrent model.

Table 3-6.: Distance at K (D@K) metrics (low is good) for video to sentence(s) ($v_i \rightarrow s_k$) and sentence to sentence(s) $s_i \rightarrow s_k$ approaches.

	$v_i \rightarrow s_k$			$s_i \rightarrow s_k$		
Model	**D@1**	**D@5**	**D@10**	**D@1**	**D@5**	**D@10**
Averaged	**0.3172**	**0.3356**	**0.3455**	0.7195	0.7626	0.7626
Recurrent	0.4581	0.4802	0.4962	**0.6856**	**0.7354**	**0.7592**

Distance metrics. As we need and enriched video latent space that can be used in a video summarization task, we expect the video and a set of retrieved sentences to be close, and the same for the ground-truth sentence of the video and the set of retrieved sentences. The latter can be expressed as two main questions:

- How close is a video to a set of k-nearest sentences in coordinated space?
- How close is the ground-truth sentence of a video to a set of k-nearest sentences in coordinated space, measured over a linguistic space?

The first question considers the discrimination power of the coordination model, and the second the semantic stability of it.

We then, inspired in the Recall@K and med$r$ metrics, measured the mean distance from the video v_i to the k–nearest sentences ($D@k$), in the cases of coordinated space ($v_i \rightarrow s_k$) and linguistic space ($s_i \rightarrow s_k$). In table **3-6**, it can be observed the distances for the video to sentence(s) and sentence to sentence(s), as explained before. Notice that although the

averaged model obtains a better performance for video and text matching, the recurrent model tends to match sentences that are more similar to the ground-truth sentence of the input video.

Qualitative evaluation

In this section, we present some application examples of the trained model, particularly the averaged model, due to its quantitative performance presented in the previous section.

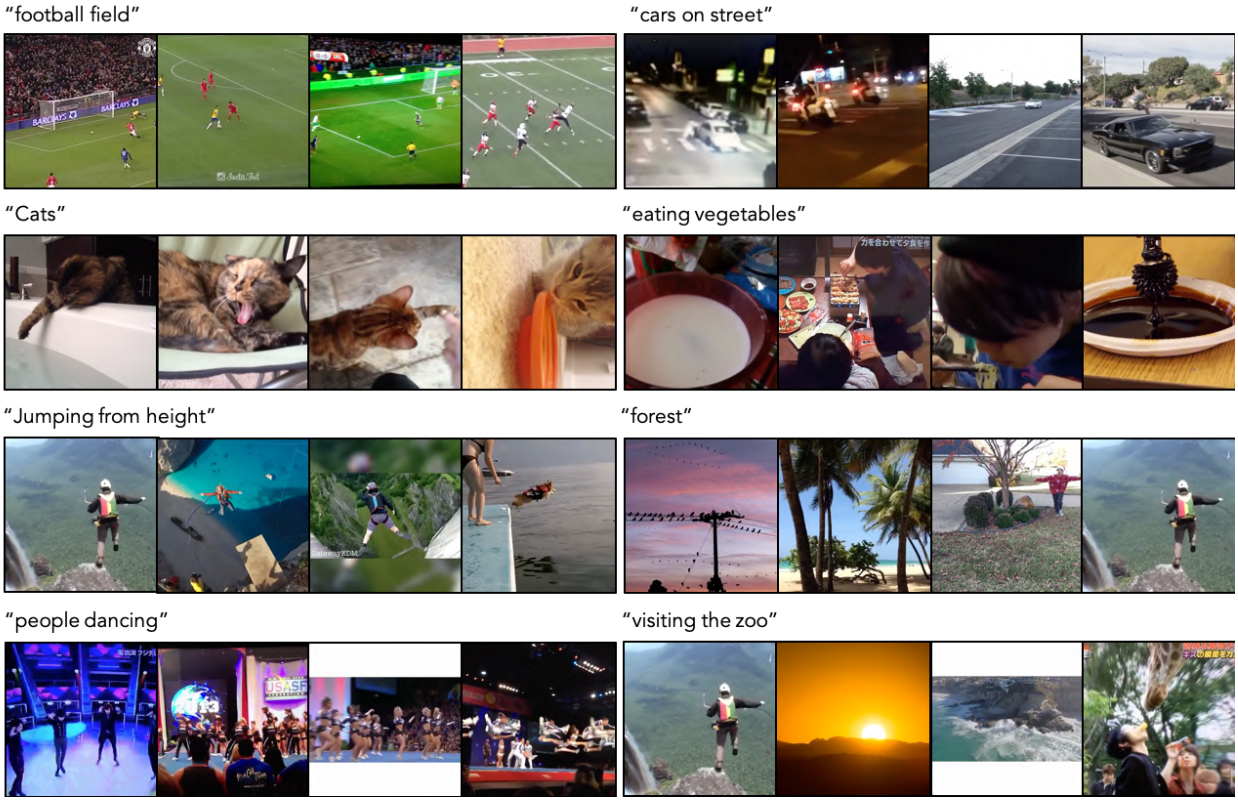

Figure 3-6.: **Video retrieval.** Examples of retrieved videos from TRECVID dataset using an input query using averaged coordination model. Videos are ordered by distance from left to right.

Video retrieval from sentence. A common application for coordination models is the retrieval from one modality to another. For example, retrieve similar videos to an input query in natural language, that is, given a set of videos V, a query sentence s and a trained coordination model M, we can retrieve the $k-nearest$ videos from V with indexes equal to $argsort_0^k(M(v,s))\forall v \in V$, as the last layer in M is a distance metric (see section 3.2.2). In figure **3-6**, it can be observed the best $4-nearest$ videos for some query examples, ordered from left to right according to query similarity. Notice that the trained model is

sensitive to word context. For example, the query *football field* retrieved *football soccer* and *American football* videos as both are related to word *football* in user sentences and in the pre-trained skip-thoughts model. Although our model does not consider object interactions and actions, semantic relations in linguistic space are transferred to the visual domain. For example, the query *jumping from height* retrieved videos related to skydiving and bungee jumping situations, but notice the last video which associated description is: *a dog jumps into water at a lake*. Also, the model is able to align some salient visual features related to word concepts, although they are not explicitly related in user descriptions. For example, videos retrieved from query *forest* relate mainly with visual features of plant cover and animals, although explicit linguistic relation does not exist in its descriptions. Finally, notice that retrieval quality is related to the diversity of the videos in the training dataset. For example, for the query *visiting the zoo*, although videos are related in linguistic space with plant cover and natural landscapes, retrieval quality has low specificity due to the fact that TRECVID dataset does not contain a significant number of videos related with zoos.

Figure 3-7.: **Sentence retrieval.** Examples of retrieved sentences from an input video from the SumMe dataset, using averaged coordination model. Each keyframe is titled with the original videoname from SumMe dataset and subtitled with the retrieved description from dataset TRECVID.

Video retrieval from sentence. A common application for coordination models is the retrieval from one modality to another. For example, retrieve similar videos to an input query in natural language, that is, given a set of videos V, a query sentence s and a trained coordination model M, we can retrieve the $k-nearest$ videos from V with indexes equal to $argsort_0^k(M(v,s))\forall v \in V$, as the last layer in M is a distance metric (see section 3.2.2). In figure **3-6**, it can be observed the best $4-nearest$ videos for some query examples, ordered from left to right according to query similarity. Notice that the trained model is sensitive to word context. For example, the query *football field* retrieved *football soccer* and *American football* videos as both are related to word *football* in user sentences and in the pre-trained skip-thoughts model. Although our model does not consider object interactions and actions, semantic relations in linguistic space are transferred to the visual domain. For example, the query *jumping from height* retrieved videos related to skydiving and bungee

jumping situations, but notice the last video which associated description is: *a dog jumps into water at a lake*. Also, the model is able to align some salient visual features related to word concepts, although they are not explicitly related in user descriptions. For example, videos retrieved from query *forest* relate mainly with visual features of plant cover and animals, although explicit linguistic relation does not exist in its descriptions. Finally, notice that retrieval quality is related to the diversity of the videos in the training dataset. For example, for the query *visiting the zoo*, although videos are related in linguistic space with plant cover and natural landscapes, retrieval quality has low specificity due to the fact that TRECVID dataset does not contain a significant number of videos related with zoos.

Sentence retrieval from video. In a similar fashion to the video retrieval from sentence task previously addressed, we can retrieve the closest description sentence to a video v from a set of sentences S using $argmin(M(v,s))\forall s \in S$. Three examples of sentence retrieval can be seen in figure **3-7**. It is important to mention that we are retrieving sentences from the TRECVID dataset using as input videos from dataset SumMe, that is, v and S come from different datasets. Also, this description is extracted for a segment of 5 seconds from the original SumMe video. Finally, sentence extraction is also sensitive to word context, for instance, in the *Bike Polo* video although *bicycle* is a frequent object, in conjunction with *night* and *street* contexts, is near sentence on TRECVID dataset is related with the object *motorbike*.

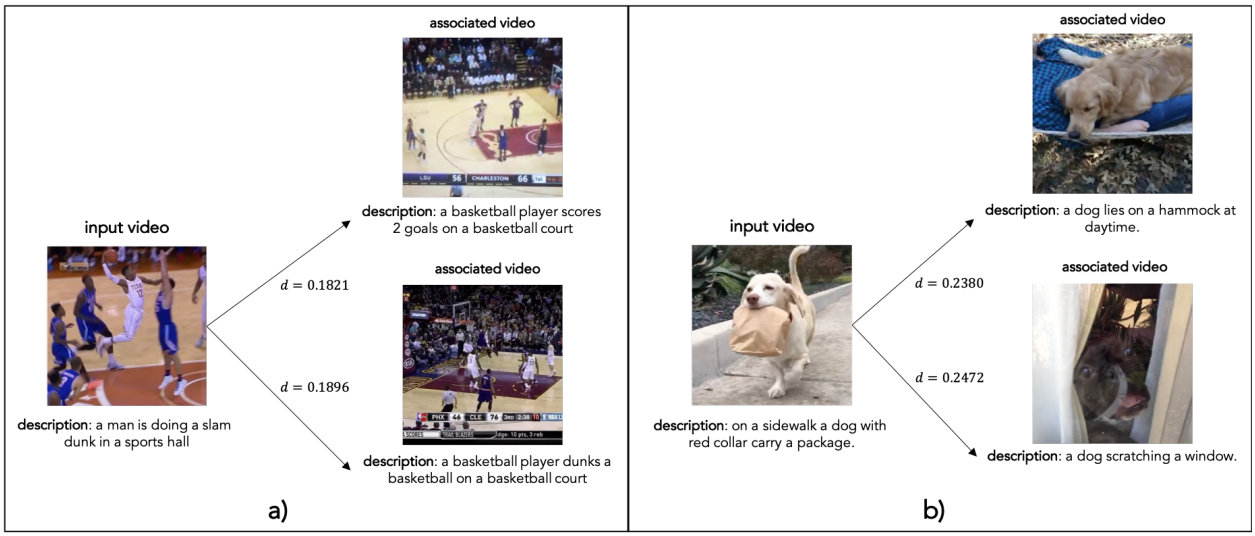

Figure 3-8.: **Video retrieval from coordinated bridge.** Examples of video retrieval from an input video and sentence matching.

Video retrieval from multi-modal bridge. It is possible to search inside a set of videos V using an input video v as a reference, by a multi-modal bridge as $v \rightarrow S \rightarrow V$. In other words, to retrieve videos by using the distance from v to the set of descriptions associated with a set of videos. In practical applications, videos may not have associated descriptions but

text metadata such as titles, comments, or tags that could be used instead. It is important to notice that this kind of retrieval is not exclusively visual but semantic, as the similarity between video v and video dataset V is not being measured between visual features, but between visual and linguistic features. In figure **3-8** we show two examples of video retrieval using this approach. Notice that in both cases, visual similarity between the input video and retrieved ones is not exactly high, but it does exist a semantic relation in terms of their content.

Query similarity

Using a similar approach to the sentence retrieval by video task (see section 3.4), it is possible to compute a degree of similarity between the segments of a video v and a query q in natural language. The previous allows us to consult which frames or segments from v are related or have significance concerning the user's interest. Given a coordination model M then, a query similarity scheme between a video v and an input query q will be given by $M(v_i^{i+w}, q) \forall i \in \{0, 1, 2, ..., n-1\}$ where n is the video length, and w is a window or segment size to take a set of frames on each iteration.

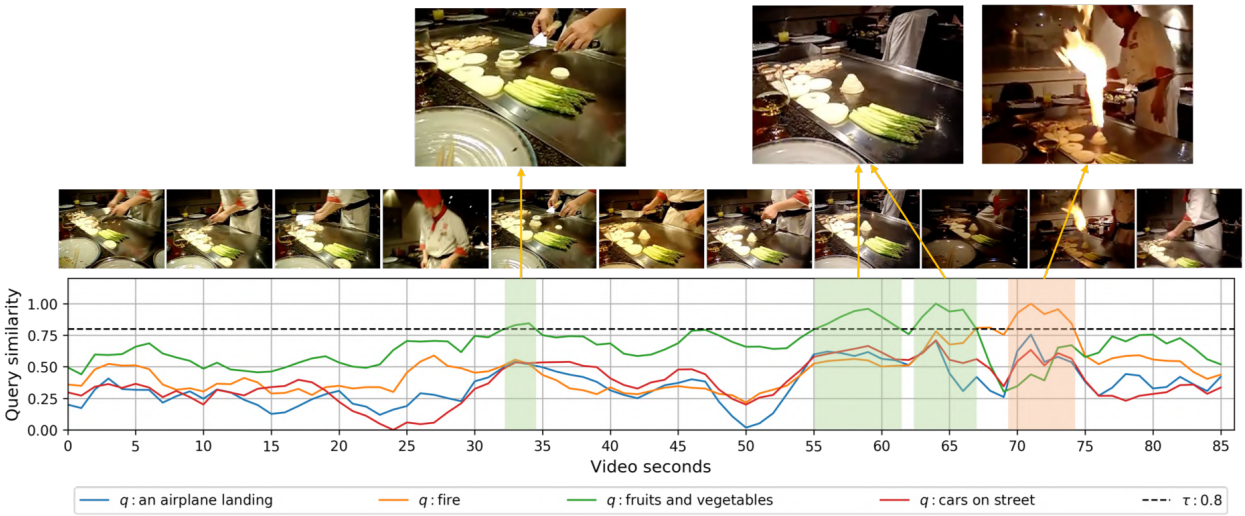

Figure 3-9.: Query example using three free-form texts over video *Cooking* from dataset SumMe.

Figures **3-9** and **3-10** show examples of query similarity for videos *Cooking* and *St Maarten Landing* from the dataset SumMe, and a set of predefined queries: *an airplane landing*, *fire*, *fruits and vegetables*, *cars on street*. In all cases, similarities are normalized between 0 and 1 to be able to use a fixed threshold τ by which a segment or sequences of frames are considered relevant if superior to it. Queries *fruits and vegetables* and *fire* give a high similarity for video *Cooking* on frames where visual content its related. Notice that although vegetables appear

in most of the video content, a high similarity is obtained where no other entity or category is present, for instance, where chief is not present on the scene.

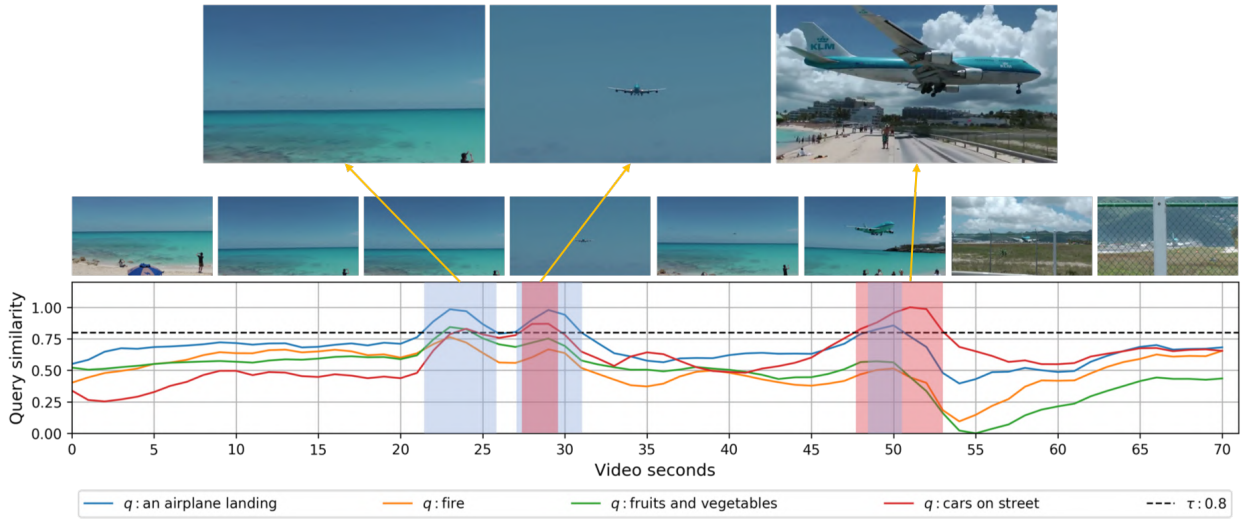

Figure 3-10.: Query example using three free-form texts over video *St Maarten Landing* from dataset SumMe.

In the case of video *St Maarten Landing*, queries *an airplane landing* and *cars on street* give the highest similarity. Notice that query *St Maarten Landing* gives a dominant response where airplane appears isolated on the scene, and on the contrary query *cars on street* gives a higher response when airplane appears over a street.

3.5. Conclusions

In this chapter, we have discussed the design, construction, training, and evaluation of a visual-linguistic coordination model that can capture and transfer phenomena from the visual to the linguistic domain and vice versa. This model can be used to estimate a degree of similarity between a video and a query in the form of a sentence, which allows injecting user intentions in a VSUMM task.

Although quantitative and qualitative evaluations bring evidence about the advantages in using a coordination model to extract video features with semantic relations, we are aware that the performance of a coordination model is directly related to the diversity and completeness of the dataset employed to train the model. We also consider that, because the TRECVID-VTT dataset, employed to train our model, is mainly composed of funny and casual videos extracted from Vine, our model lacks generalization power. Future work can explore the integration of different VTT datasets, which allows improving this situation.

Another future work we consider is the evaluation of different losses functions, for example, the family of triplet losses, which requires complex criteria to build the $(positive, negative, anchor)$ triplets from a VTT dataset.

Finally, although we evaluated recurrent schemes for video-frames processing, the best results were achieved by employing averaging windows. We believe that further exploration of recurrent layers can be made in future works, considering that these layers require more data to obtain better performance.

4. Representativeness and uniformity

4.1. Related work

The representativeness of a video summary can be understood as the property of a video summary to contain the set of segments or frames which returns the minimum distance to the original video. Del molino et. al. define representativeness in [3] as: «*the most similar instances to the rest of the video*», that is, the subset $S \subset V$ such that $min\|S - V\|_2^2$, where S is the summary and V the original video. On the other hand, a temporal coherence or uniformity is desired for the final summary [3, 2, 10, 47], in order to avoid frantic video summaries which could confuse observers. Both objectives (representativeness and uniformity) are closely related in the sense that a combination of a segmentation method apply over the input video and a selection of the most representative segments by a clustering technique will force a temporal coherence in the final summary.

$$F(S) = \sum_{x \in X} min_{s \epsilon S} \|x - S\|_2^2 \tag{4-1}$$

From a computational approach, representativeness is commonly addressed as a *k-medoids* clustering over video segments [59, 2, 47, 10, 13] (see figure **4-1**). In equation 4-1 it is shown the k-medoids formulation minimization objective, where X is the set of feature vector from video V, and S is the set of features from a set of segments.

Recent approaches use the determinantal point process (DPP) in order to obtain a representative video summary. Gong et. al. in [33] developed a sequential DPP (seqDPP), which demonstrates good performance for VSUMM task. Nonetheless, in this work, we do not follow a DPP strategy since it requires a training stage, making the evaluation dependant of the (X, Y) set, and dependant of the dataset splitting method, for example, $(train, test)$, k-fold and leave-one-out. In order to test the coordinated-space performance for the VSUMM task, we decided to isolate the representation power from a trainable model, and use a general approach, i.e., a clustering method.

«*In the k-medoids problem, we find a subset* $S = S_k | k = 1, ..., K$ *of video segments, which are cluster centers that minimize the sum of the Euclidean distance of all video segments to their*

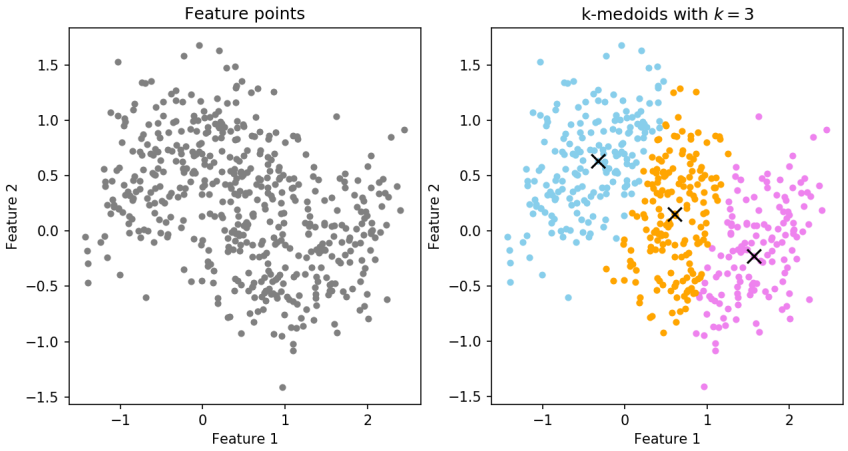

Figure 4-1.: k-medoids example with $k = 3$ in a R^2 space.

nearest cluster centers $S_k \in S$ *and* K *is a given parameter to determine the length of the video summary.*»[59].

In other words, the k-medoids algorithm take segments X and generate a subset S with minimal distance to X. Generally, segments are generated using uniform sampling. We generate X using a hierarchical clustering algorithm and a correlation matrix from coordinated-features in order to obtain non-uniform sampling segments which will be used to unify semantically related sequences of frames.

4.2. Model

Overview. In figure **4-2**, a graphical depiction of the developed model for representative and uniform video summary generation using coordinated models is shown. Our model first computes the coordinated visual features given an input video V, using a pre-trained video-language coordination model, as explained in chapter 3. Then, we apply a hierarchical segmentation algorithm to detect the main segments of V in the coordinated space, and the number of them, which we use later as an estimation of the parameter k for k-medoids. Next, the k-medoids algorithm is employed over the coordinated visual features in order to obtain a set of medoids, which are filtered by removing meaningless and redundant medoids, to obtain a representative video summary R. Finally, a representative and uniform summary U is obtained using the set of segments previously detected by hierarchical segmentation, and the filtered set of medoids, applying a mass-center unification strategy. In the case of VSUMM tasks where it is required that the length of the output summary is a percentage α of the input video V, we change the parameter k estimation as a relation expressed in

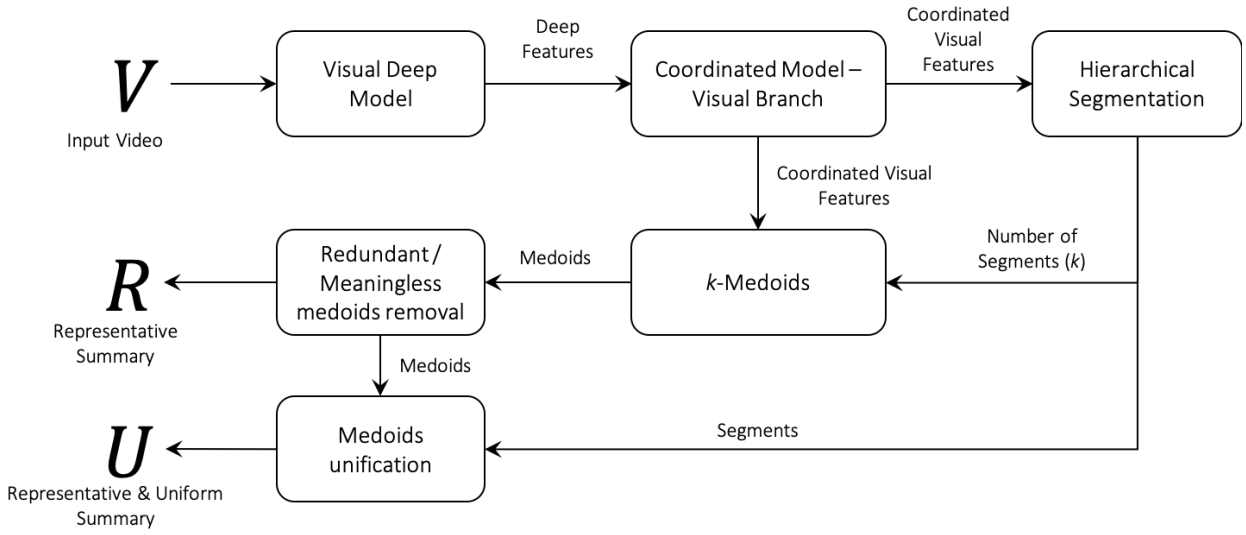

Figure 4-2.: General stages for the k-medoids approach for representative video summarization over coordinated space given an input video V.

equation 4-2.

4.2.1. k-medoids clustering over coordinated space

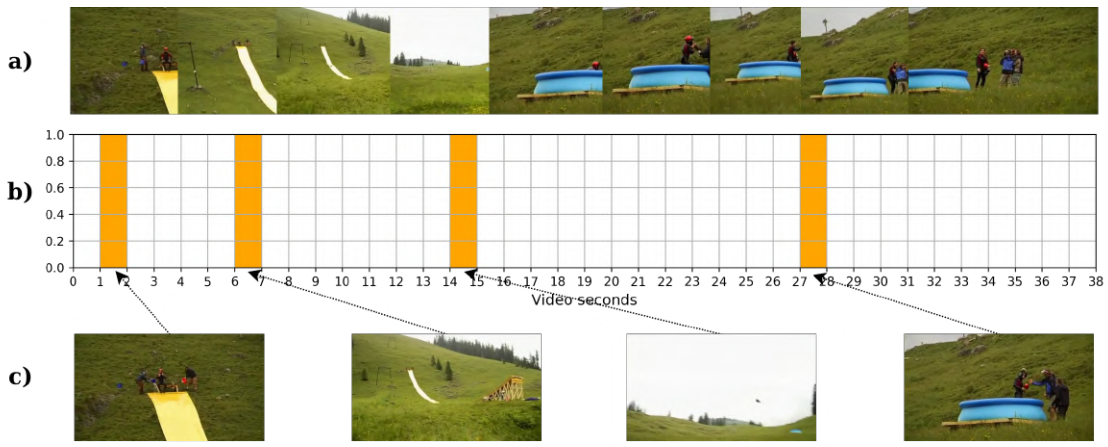

Figure 4-3.: Example of representative summary by k-medoids over coordinated space, with $k = 4$. **Video**: Jumps, from dataset SumMe. **a)** V input video, **b)** Medoids selected using coordinated space, **c)** Key-frames from select segments.

Using the coordinated model previously trained and shown in chapter 3, we take an input video V and use the visual branch of the model for computing its coordinated activation, which we demonstrated also contains linguistic relations. Then, computing the k-medoids

algorithm over coordinated activations, we obtain the more representative k segments from the input video. An example of this model is shown in figure **4-3**. Notice that medoids do not correspond with a uniform segments selection, that is, long shots with low variance can be clustered in a single center by the k-medoids algorithm.

As the k-means clustering technique, k-medoids require the selection of a parameter k, which defines the number of clusters centers that will be found. In order to generate a video summary using k-medoids it is then required to define k by two approaches:

- k **as a percentage of the video length**: For which it is needed a uniform segmentation stage in order to compute the number of clusters in relation to the segment size. In equation 4-2 it is shown the relation between k, the percentage α of the input video length $|V|$ and the uniform segment size $|S|$.

$$k = \left\lfloor \frac{\alpha |V|}{|S|} \right\rfloor \tag{4-2}$$

- k **as a measure of the change in the video content**: For which it is needed, a dynamic segmentation/shot detection method and k can be defined as the number of video shots detected (1 cluster per shot). This approach will be further explained in section 4.2.2.

4.2.2. Hierarchical segmentation and k estimation

As previously mentioned in section 4.2.1, in order to perform the k-medoids algorithm, it is necessary to select or estimate the parameter k. Although it is possible to estimate this parameter by using a user-criteria or a percentage α of the original input video length $|V|$ ([59, 2, 10, 47, 13]), as we later explore in section 4.3.1, the evaluation of representative summaries by user-selected keyframes requires the estimation of k by a diversity-like feature, as many users can select summaries with different lengths given an input video V.

De Avila et al. in [93] estimate k using a differential approach, for which they compute pairwise distances between consecutive frames and increment k as this distance is higher than a threshold τ. This approach has some limitations, such that the use of a fixed threshold $\tau = 0.5$ and the consecutive difference calculations consider only abrupt changes in the video content, such as video transitions or scene changes. Iparraguirre and Delrieux propose a similar approach in [94], where the authors use an estimated amount of noise and a sensitivity threshold to select keyframes from an input video.

We used a segmentation method, which allowed considering a temporal window and hierarchical relations between video features, in order to estimate k as the number of shots or

sequences of frames with similar features. As segments must be consecutive, we computed distances as pair-wise cosine between coordinated activations C_i, C_j for frames i and j, and cutting values at a temporal threshold τ, as proposed in [7], in order to prevent disjoint long-term relations between frames.

$$\begin{aligned} w_{i,j}^{t} &= \frac{1}{t} max(0, t - |i - j|) \\ D_{i,j}^{coordinated}(C_i, C_j) &= 1 - w_{i,j}^{t} e^{-\frac{1}{\Omega} cos(C_i, C_j)} \end{aligned} \tag{4-3}$$

This procedure is shown in equation 4-3, where t is an arbitrary threshold expressed in number of frames, C_i is the coordinated activation of frame i, and Ω is the mean of distances among all frames. Cosine distance between vectors A and B is expressed as: $cos(A, B) = 1 - \frac{A \cdot B}{|A||B|}$.

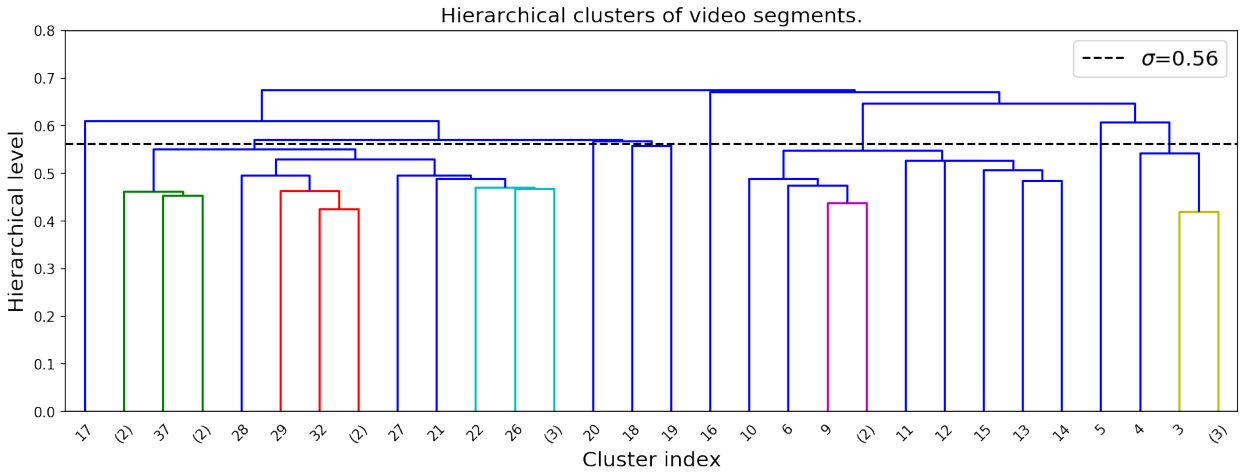

Figure 4-4.: Dendogram that represents an example of clusters obtained from hierarchical clustering over coordinated space. In this case, using $\sigma = 0.56$, seven clusters are finally selected.

Next, we use the agglomerative z-linkage algorithm to obtain hierarchical clusters of video-frames with low distance over coordinated features. An example of hierarchical clusters detected by the z-linkage algorithm can be observed in figure **4-4**.

Then, hierarchical clusters are flattened according to the arbitrary threshold σ that controls the sensitivity of the cluster agglomeration, i.e., a low value of σ returns a high number of clusters, and the contrary for a high value of σ. For automation purposes, σ can be estimated using a measure like the mean of the distances of the clusters, or a weighted approach such as $\beta\mu(Z)$ where β is an arbitrary weighting parameter, and $\mu(Z)$ is the mean of the clusters distances Z. In figure figure **4-4** it can be observed that a $\sigma = 0.56$ flattened hierarchical clusters to seven ones.

In figure **4-5**, it can be observed the dissimilarity matrix over coordinated features $D^{coordinated}$ obtained using equation 4-3 and the detected shots using the hierarchical clustering algorithm.

Finally, the parameter k for the k-medoids algorithm (see section 4.2.1) can be automatically estimated as the number of segments obtained from the hierarchical clustering process, as the segments have relations with the set of unique scenes in a video over a feature space.

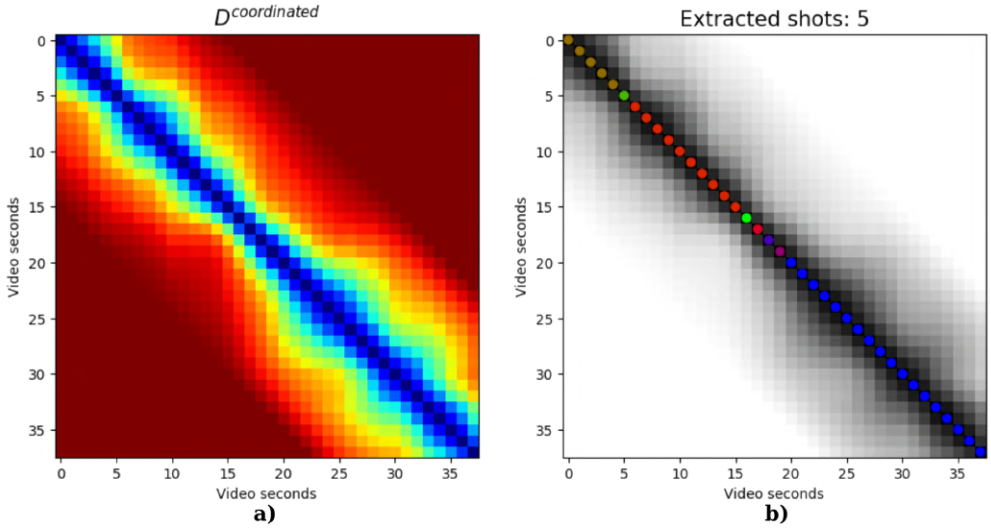

Figure 4-5.: **a)** Dissimilarity matrix and **b)** extracted shots using linkage algorithm. In this case a temporal window t = 15 was used. **Video:** Jumps, from the SumMe dataset.

In figure **4-6** it can be observed the retrieved keyframes from representative segments obtained by the method previously presented. Notice that the segments in figure **4-6**(b) correspond with the hierarchical clustering from figure **4-5**(b), and the number k of medoids is equal to the number of shots extracted from hierarchical clustering. Also, notice that there is no 1:1 correspondence between shots and medoids. The latter, because the segmentation scheme considers sequentiality, differs from the k-medoids clustering, which is not, i.e., similar non-sequential video sections produce one medoid, but at least two shots.

4.2.3. Redundant and meaningless medoids removal

Transitions between video scenes can contain black or white frames (fade-in / fade-out), which can be detected as medoids due to the high variance of its features concerning previous and next video frames. Moreover, when k from k-medoids is higher than the number of segments in the video, redundant medoids could be selected. Due to this, it is necessary to perform a redundant and meaningless medoid removal strategy.

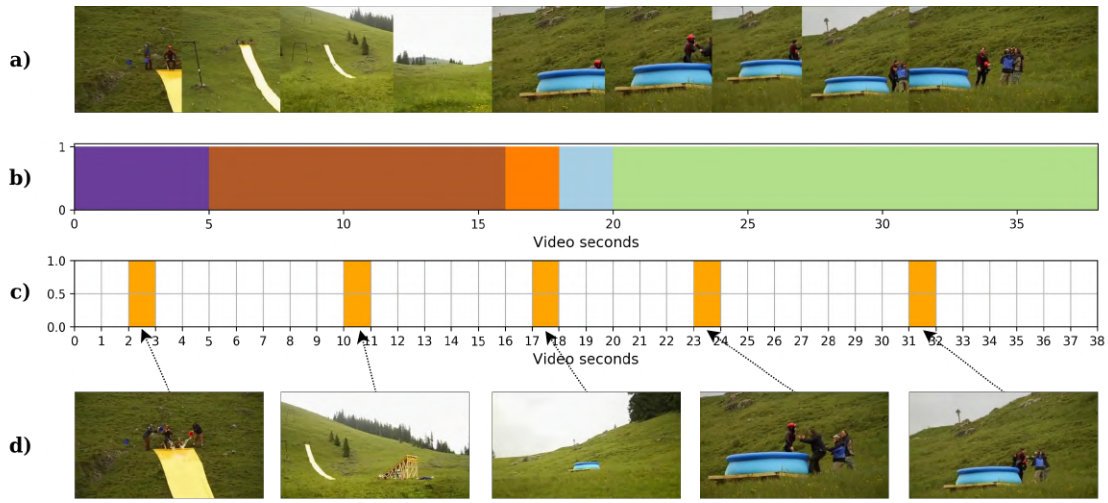

Figure 4-6.: Example of a representative summary by hierarchical segmentation and k-medoids. **a)** Input video, **b)** extracted segments / shots, **c)** k-medoids $k = 5$ and **d)** keyframes from medoids. **Video:** Jumps, from the SumMe dataset.

Algorithm 1: Redundant and meaningless medoids removal over coordinated space.

Input : Set of medoids in the coordinated space S, Set of frame sequences F for each medoid, Similarity threshold τ_1, Contrast threshold τ_2

Output: A subset of medoids R

```
begin
    for s ∈ S, f ∈ F do
        R = []; // empty list
        for r ∈ R do
            l = True;
            if D(s, r) < τ1 then
                l = False;
                break;
            end
            if l & (σ(f) < τ2) then
                R.add(r);
            end
        end
        return R;
    end
end
```

Inspired by the strategy proposed by De Avila et. al. in [95], we followed the next stages in order to remove the latter:

- **Redundant medoids:** Given a set of medoids S over a **coordinated space**, we remove redundant medoids by sequentially measuring the distances $D_i = (S_i, S)$ for each medoid S_i with respect all set S, and removing those medoids for which $D_i < \tau$, where τ is a threshold of similarity experimentally defined. An algorithm for this stage can be studied in Algorithm 1.
- **Meaningless medoids:** Given a set of **frame sequences** F for each medoid in S, we remove meaningless medoids measuring its contrast using standard deviation, and comparing it with a threshold τ_2, for which those medoids where $F_i < \tau_2$ are removed. An algorithm for this stage can be studied in Algorithm 1.

In figure **4-7** it can be observed an example of the removal stages applied to video **v25** from the SumMe dataset. Notice that from 12 initial medoids, 6 medoids were preserved, 1 was removed for low contrast ($a4$) and 5 were removed for redundancy: $a2 \rightarrow a1$, $(b2, b3) \rightarrow b1$, $c1 \rightarrow b4$ and $c3 \rightarrow c2$.

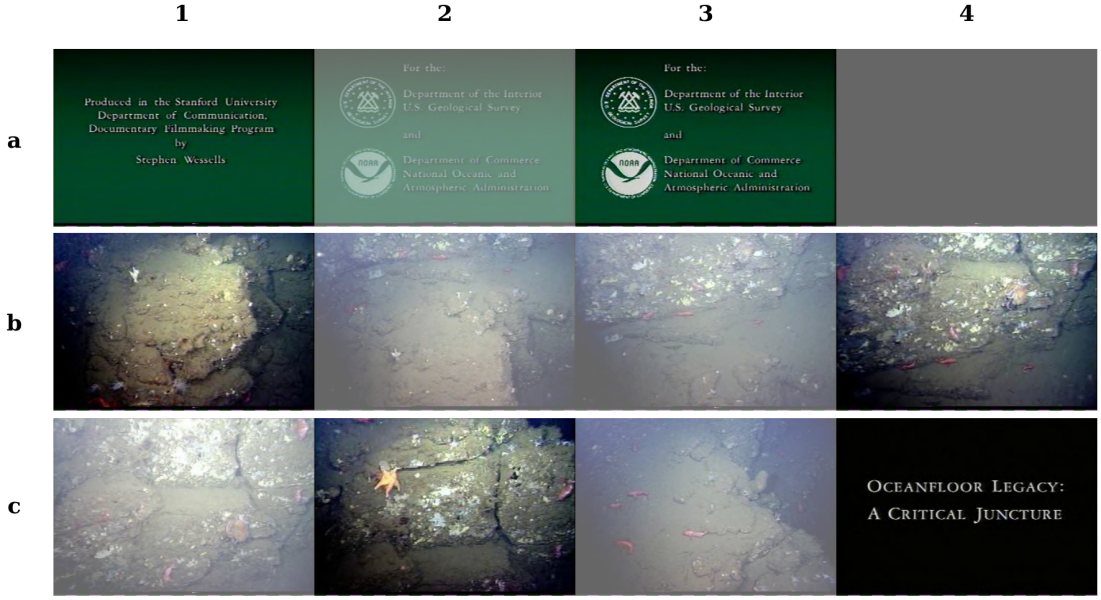

Figure 4-7.: Example of the redundant and meaningless removal stages over video **v25** from dataset OVP [95]. In high contrast, the segments that were preserved.

4.2.4. Uniformity

Generally, representativeness is complemented with uniformity, which can be understood as temporal coherence [3], in the sense that significant gaps between scenes in the video

summary can affect the flow of the story [47] and therefore the user interpretation of the video summary. Nonetheless, a complete unification of the video summary is contrary to the representativeness objective, so a balance criterion is needed to consider both VSUMM objectives.

Considering a segment/shot as a sequence of frames close in a feature space, we can assume that segments are independent of each other and can be unified locally in terms of its indexes. From the previous, we propose a mass-center based uniformity method which works as follows: using the hierarchical segmentation (see section 4.2.2), we obtain a set of segments over coordinated space, by which it is possible to consider that sets of medoids inside each segment can be unified due to is closeness in the coordinated space. Finally, inside each segment, a contiguous sequence of frames are obtained by computing the mass-center of the medoids inside it and taking using a centered window of size equal to the number of medoids inside the segment.

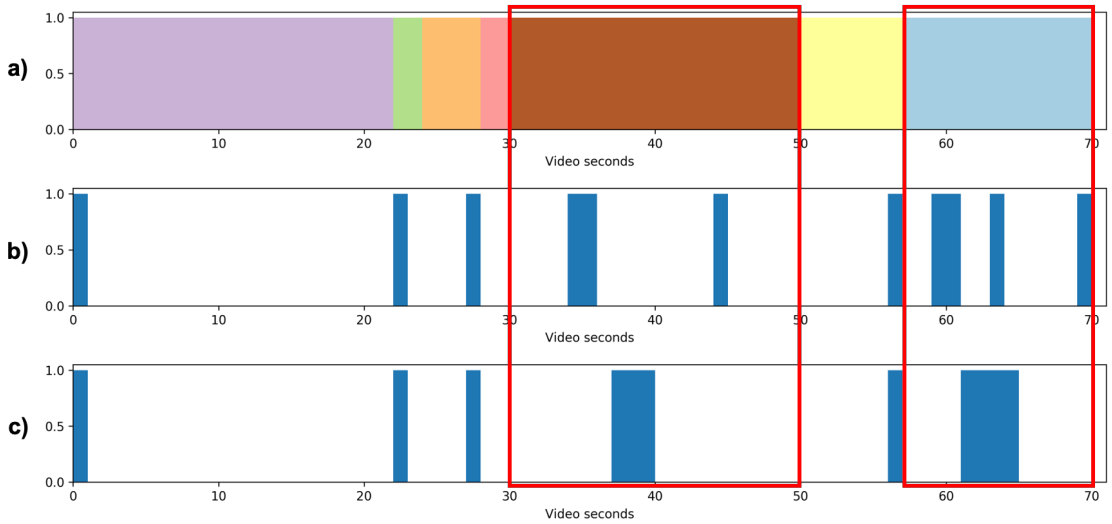

Figure 4-8.: Example of the uniformity process by merging medoids inside each video shot using its centroid. **a)** Video shots by hierarchical segmentation, **b)** k-medoids and **c)** unified medoids. In red, unified video shots by the centroid of its medoids.

In figure **4-8** it is shown an example of this method. In a), it can be observed the set of segments obtained by hierarchical segmentation, b) the set of medoids using k-medoids clustering over coordinated space, where k is equal to the number of segments detected, and c) the uniform summary by the unification of medoids inside segments.

An advantage of this approach is that it allows the control of the uniformity strength by using the σ parameter, which controls the hierarchical segmentation sensitivity, i.e., the higher the σ parameter, the higher the uniformity strength. This is explained in section

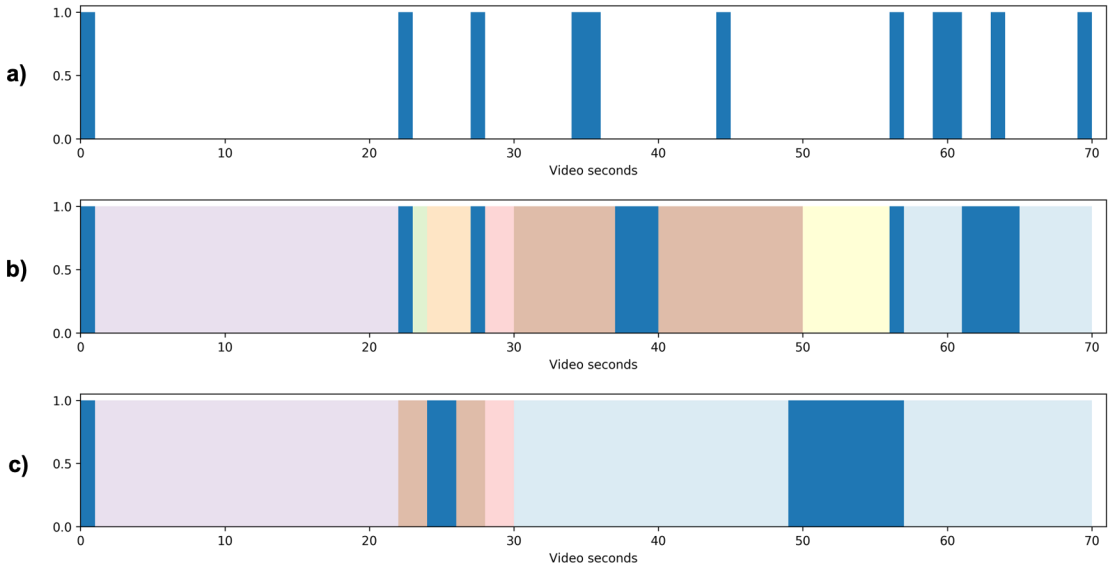

Figure 4-9.: Example of the sensitivity of hierarchical segmentation process and uniformity. **a)** k-medoids, **b)** medoids unification for $\sigma = 0.58$, and **c)** medoids unification for $\sigma = 0.67$. In the background of **a)** and **b)** it can be observed the segments from the hierarchical clustering algorithm.

4.2.2. A graphical depiction of segmentation sensitivity and uniformity can be observed in figure **4-9**, where two different values of threshold σ bring different uniformity results.

4.3. Experiments

We performed both quantitative and qualitative evaluation of our method. For quantitative evaluation we used publicly available datasets (OVP+Youtube) and compared it with baselines and state-of-art methods. For qualitative evaluation, we analyzed examples of generated summaries over SumMe and TVSum datasets.

4.3.1. Evaluation of a representative video summary method

Since a complete VSUMM method needs to consider multiple optimization objectives (see chapter 1), its evaluation is a non-trivial task, which requires user-annotated datasets that isolate specific VSUMM objectives. For example, datasets such as SumMe [2], TVSum [31] and UTE [7], are intended to be used for evaluate importance and interestingness objectives under a fixed VSUMM length α (see chapter 5), considering the experiments by which those datasets were build, which required users to assign an *importance score* to video frames from

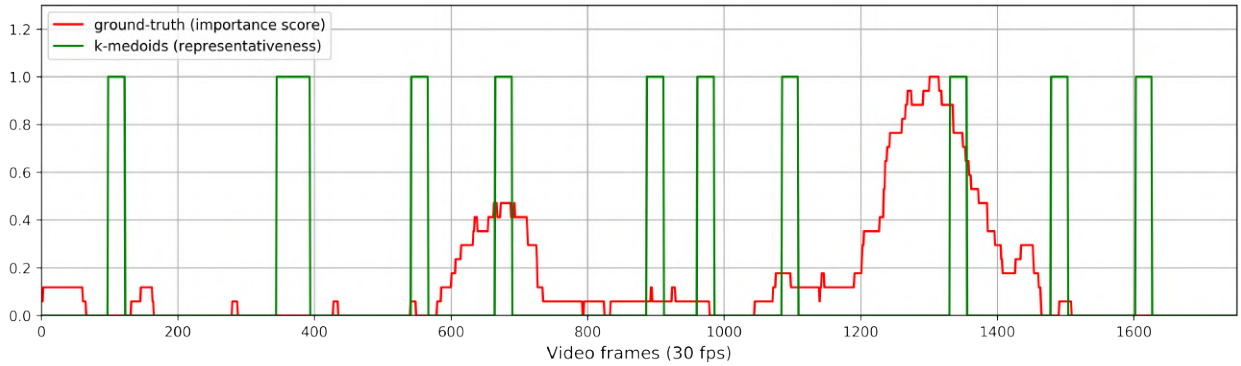

Figure 4-10.: User annotated importance score (red) for video "St Maarten Landing"from the SumMe dataset vs representative summary by k-medoids (green). Notice the low agreement between importance and representativeness.

different videos.

Although a clear taxonomy of VSUMM datasets by VSUMM objectives does not exist in the literature, it is essential to consider its intention by analyzing how they were built and what does it mean the (X, Y) pairs. It is common to find examples on the literature were proposed method was intended to work with a particular objective, but used dataset had another, for example, Otani et al. in [59] used SumMe dataset (importance score) to evaluate a representative VSUMM method. Plummer et al. [47], who used UTE and TVSum datasets, to evaluate a VSUMM method, which also considers representativeness and uniformity. An example of this situation can be observed in figure **4-10**. Notice that there is a low agreement between detected medoids (green) and the importance score (red) annotated by five users at $\alpha = 0.15$.

Keyframe based F-Score

For the general nature of a representative VSUMM, which does not take into account importance but the closest set of segments to the original video, the primary approach for evaluation is to measure a degree of agreement concerning a set of reference keyframes selected by a group of users, given the task of selecting the set of frames which contains the general content of an arbitrary video V. In general, given a set of segments/shots/medoids S selected by a computational method, and a set of keyframes K selected by a group of n users, we need to measure how many times elements in S corresponds with elements in K.

A precision/recall F-Score approach is the standard measure in the VSUMM literature. Nonetheless, in order to use F-Score between S and K it is needed a matching function

between elements in both sets, and considering that perfect matches are unlikely, a threshold-based matching function is required.

Following [95, 33] we evaluate distances in coordinated space between elements in S and K and count the number of unique matches when $D(S_i, K_i) < \tau$, where τ is an user-defined similarity threshold. It is important to ensure that matching pairs are counted only once, similarly to the redundant segments removal scheme explained in section 4.2.3.

Finally, F-Score is computed using the number of matches as positive predictions over the number of elements in S and K as can be observed in equations 4-4.

$$P_{S,K} = \frac{matches(S,K)}{|S|} \quad , \quad R_{S,K} = \frac{matches(S,K)}{|K|} \quad , \quad F_{S,K} = 2 \cdot \frac{P_{S,K} \cdot R_{S,K}}{(P_{S,K} + R_{S,K})} \tag{4-4}$$

4.3.2. Quantitative evaluation

Data

We evaluated our method over the open video project (OVP) dataset developed at the Interaction Design Laboratory at the School of Information and Library Science, University of North Carolina Chapel Hill [1], and an extension (Youtube), proposed by De Avila et al. in [95] [2]. Both datasets contain 50 videos and reference keyframes by five users each.

Metrics

We employed precision, recall, and F-score (see section 4.3.1) metrics to compare the agreement between video summaries generated by our method, and keyframes selected by a group of users. For each video in both datasets (OVP+Youtube), we averaged the F-Score for the five users and also averaged F-Score per dataset.

Reference methods

We compared our method with the pre-computed summaries available in the OVP+Youtube datasets, i.e., we did not make implementations of state-of-the-art methods but compute F-score with the already computed keyframes for each one. Each method is described as follows:

[1] https://open-video.org/project_info.php
[2] https://sites.google.com/site/vsummsite/

- Delaunay Triangulation (DT): Mundur et al. in [96] proposed a VSUMM method based in which performs clustering using Delaunay Triangulation. Authors expose this method does not require user-defined parameters and performs better than k-means clustering methods.
- STIll and MOving video storyboard (STIMO): An on-the-fly method for VSUMM was proposed by Furini et al. in [97]. Authors use HSV frame color distribution and fast clustering algorithm for this purpose. An interesting property of this method is that it considers aural information (audio) to detect video shots.
- Static video summaries ($VSUMM_1$, $VSUMM_2$): Proposed by De Avila et al. in [95], this methodology uses color features extraction, k-means clustering and shots detection by frame difference thresholding. The main difference between $VSUMM_1$ and $VSUMM_2$ is that the latter uses key cluster selection as in [98].

Table 4-1.: f-score, precision and recall (higher is good) in OVP and Youtube datasets for different representative VSUMM methods. In each case, in bold it is shown the best result and the second best in underline.

	OVP					**Youtube**	
	DT	**STIMO**	**$VSUMM_1$**	**$VSUMM_2$**	**Ours**	**$VSUMM_1$**	**Ours**
F	0.6163	0.6516	**0.8142**	0.7467	<u>0.7611</u>	**0.7771**	0.6712
P	<u>0.7318</u>	0.6217	0.7818	**0.8111**	0.7111	**0.7860**	0.6366
R	0.5707	0.7396	**0.8844**	0.7272	<u>0.8644</u>	**0.8088**	0.8061

In table **4-1** it is shown F-score, precision, and recall results for OVP and Youtube dataset between the state of the art methods and ours. It is essential to mention that we did not re-implemented or executed the baselines methods but used the keyframes dataset given by authors for each method. Although our method does not achieve the best results, for the representativeness objective, the obtained results show that the coordinated visual space computed from the coordination model trained in chapter 3, is useful as a general visual descriptor for computer vision tasks, but also will allow to perform more complex tasks such as linguistic comparison as will be detailed in chapter 6.

4.3.3. Qualitative evaluation: Keyframes over OVP dataset

Below, we present some visual examples selected keyframes from our method and $VSUMM_1$ method proposed by Avila et al. in [95], for three videos in the OVP dataset. In each case, the reference keyframes for the video (1-user), the keyframes selected from both methods, and its F-score, precision, and recall are presented.

In figure **4-11** it is shown the results for video 27 from OVP dataset. Notice that our method obtained a higher f-score due to higher recall, i.e., more frames from the reference keyframes,

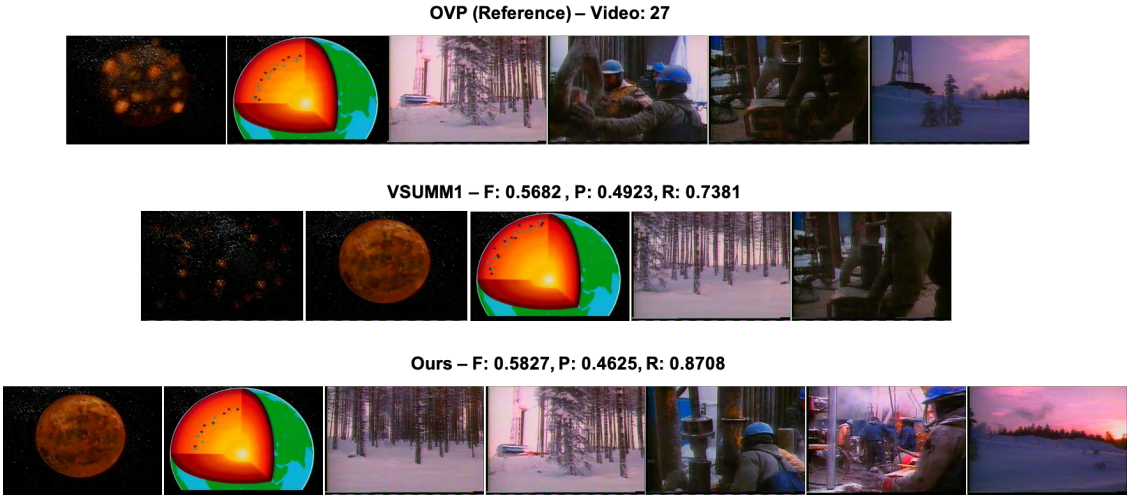

Figure 4-11.: Example of keyframes generated by our method and $VSUMM_1$ method for video 27 from the OVP dataset. f-score (F), precision (P) and recall (R) using a matching approach, are presented.

are contained in the keyframes generated by our method than the keyframes generated by $VSUMM_1$.

An opposite case can be observed in figure **4-12**. A higher f-score is achieved by $VSUMM_1$ method, due to higher recall, even when both methods obtained a precision equal to one.

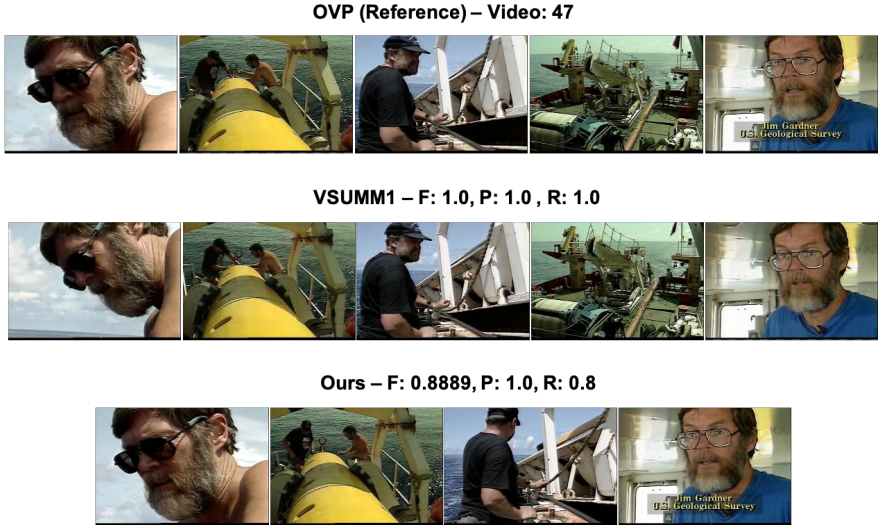

Figure 4-12.: Example of keyframes generated by our method and $VSUMM_1$ method for video 47 from the OVP dataset. f-score (F), precision (P) and recall (R) using a matching approach, are presented.

Finally, a case where higher precision is achieved by our method can be observed in figure **4-**

13. Notice that all of the keyframes obtained by our method are inside the keyframes obtained by $VSUMM_1$, and both methods have a recall equal to one. However, while $VSUMM_1$ selected extra frames not included in the reference, its f-score was penalized by its precision.

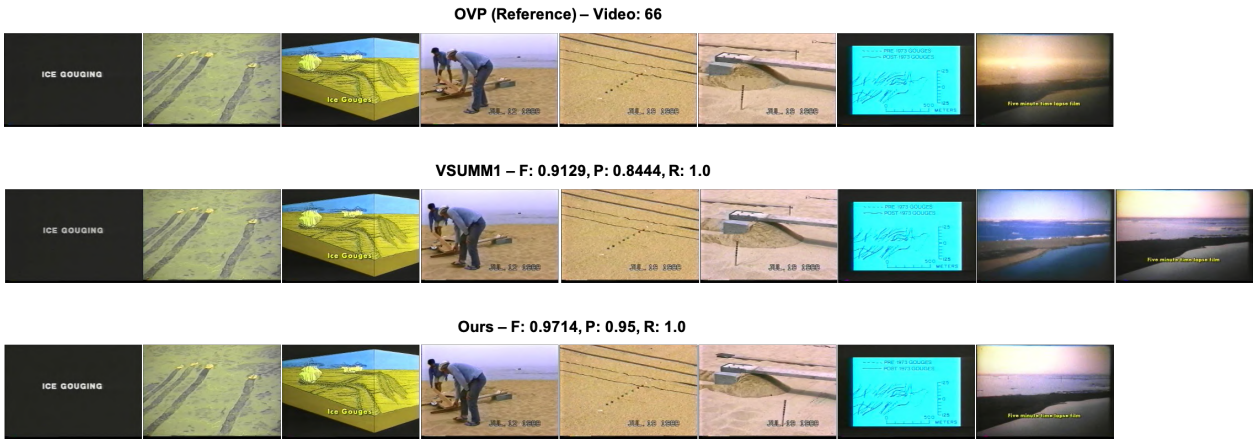

Figure 4-13.: Example of keyframes generated by our method and $VSUMM_1$ method for video 66 from the OVP dataset. F-score (F), precision (P) and recall (R) using a matching approach, are presented.

4.3.4. Qualitative evaluation: Video summary

In order to visually evaluate the performance of the developed method in non-keyframes datasets, as SumMe or TVSum, we modified the k estimation (see section 4.2.2) to work traditionally, considering the length of the summary as a ratio of the length of the input video, as previously explained in equation 4-2. This scheme will also be used later in chapter 6 in order to allow multiple VSUMM objectives integration. In all cases an α = 0.15 was employed, that is, a summary equals to the 15 % of the original video length, and a) the sampled input video, b) the unified medoids and segments, and c) the central frames from each unified set of medoids are shown.

In figure **4-14** it is shown a 15 % representative and uniform summary generated by our method for the video:*St Maarten Landing* from dataset SumMe. Notice that, although the original video is mainly composed by a scene where the beach and the sea are shown, airplane landing is also selected by our method as a representative set of medoids **(b,c)**, which differs from a uniform approach **(a)**. Also, notice that at the beginning of the input video, it can be observed a scene where there is a group of people and an umbrella **(a)**. However, as in the middle of the video, a **longer** scene where a group of people and an umbrella appears again, both scenes are considered a single medoid (third medoid in **(b)**) by our method.

An example where scenes with high visual variance are considered more representative,

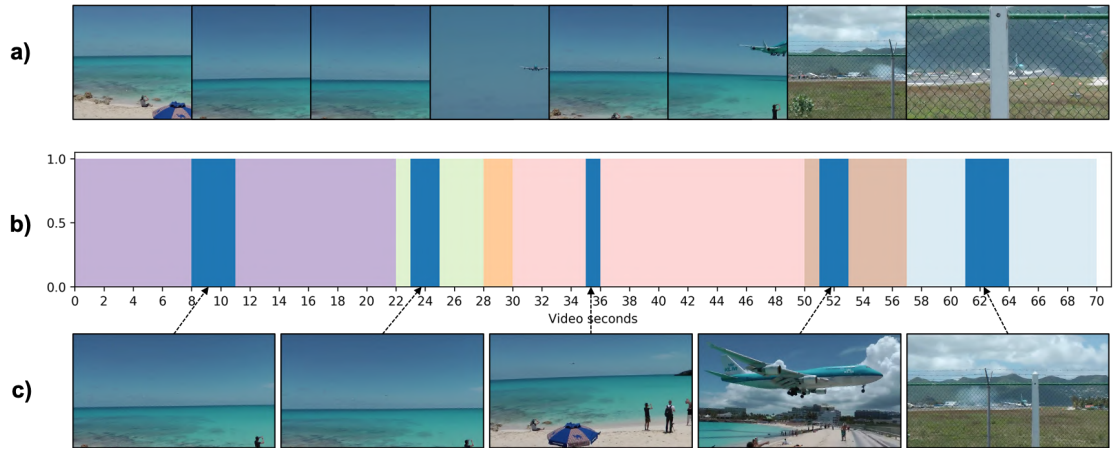

Figure 4-14.: Example of representative and uniform summary U for video:*St Maarten Landing* from dataset SumMe, generated by our method, using an $\alpha = 0.15$. a) Input video sampled to 1 frame each 10 seconds, b) extracted segments by hierarchical segmentation with $\sigma = 0.6$ and unified medoids (blue), and c) central frames from each unified medoids sets obtained by our method.

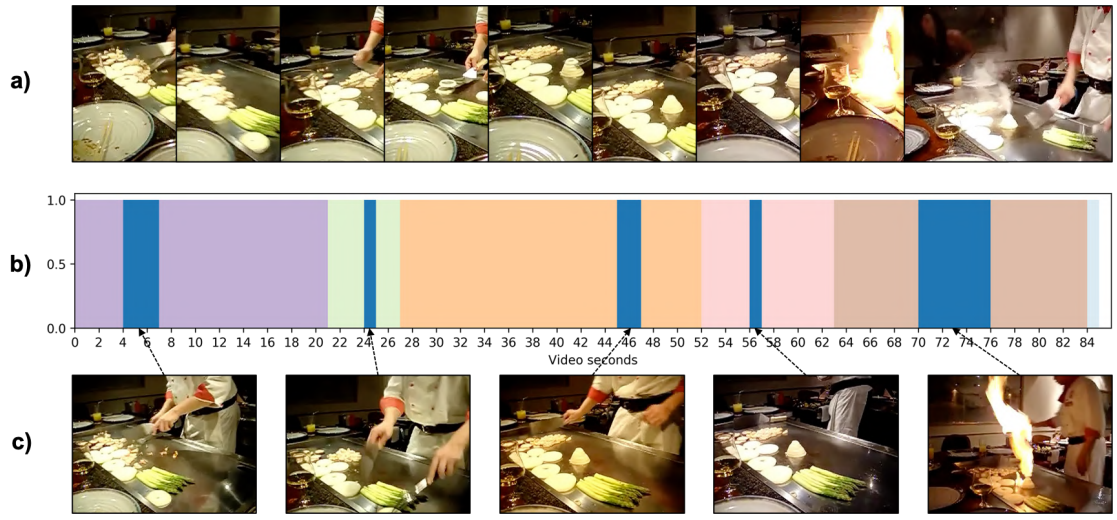

Figure 4-15.: Example of representative and uniform summary U for video:*Cooking* from the SumMe dataset, generated by our method, using an $\alpha = 0.15$. a) Input video sampled to 1 frame each 10 seconds, b) extracted segments by hierarchical segmentation with $\sigma = 0.6$ and unified medoids (blue), and c) central frames from each unified medoids sets obtained by our method.

can be observed in figure **4-15**. This video (*Cooking*) from the SumMe dataset, shows a chef cooking oriental food (almost $\frac{4}{5}$ of the video), and finally, he makes a flame in the preparation. Notice that, as the input video is highly static, the final scene (fire) is selected as the longest unified medoids set by our method, as it contains great visual dynamics, and

therefore a high number of medoids.

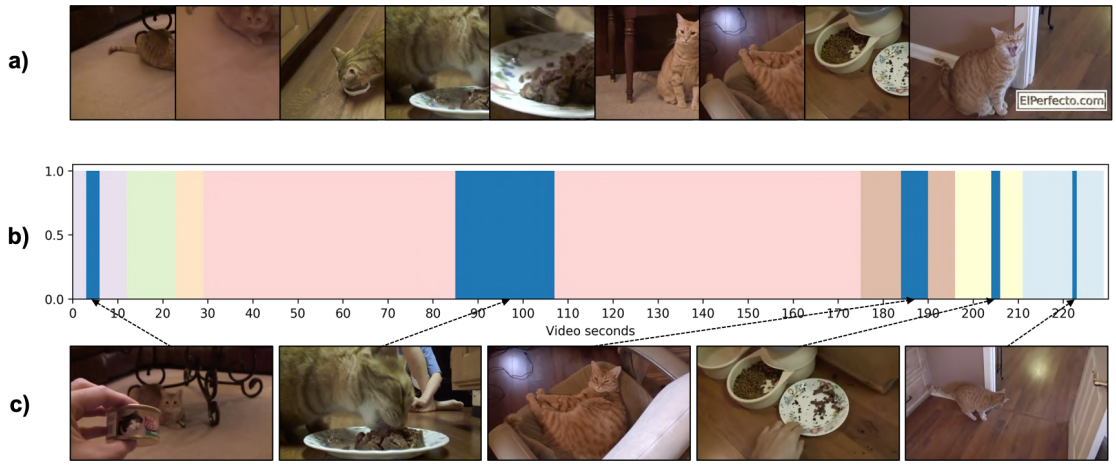

Figure 4-16.: Example of representative and uniform summary U for video:-*esJrBWj2d8* from TVSum50 dataset, generated by our method, using an α = 0.15. a) Input video sampled to 1 frame every 30 seconds, b) extracted segments by hierarchical segmentation with σ = 0.6 and unified medoids (blue), and c) central frames from each unified medoids sets obtained by our method.

Finally, an example for a video (*-esJrBWj2d8*) from dataset TVSum50 it is shown in figure **4-16**. This video is mostly composed of scenes of a cat eating. Even when different angles, views, and positions are taken, the main element is *a cat eating*. It is important to notice that the hierarchical segmentation process over the coordinated features (video-language) was capable of capturing this relation, which can be observed in the large pink segment in **(b)**, which was finally unified in a single set of medoids.

4.4. Conclusions

In this chapter, we have developed a computational method to obtain representative and uniform video summaries using visual-linguistic coordination models. Main advantages of our method are a) generated summaries considers semantically related scenes, useful in VSUMM of highly dynamical egocentric videos, b) automatically estimates parameter k from k-medoids, using hierarchical clustering segmentation, useful for long videos, and c) textual queries can be performed directly over coordinated features as discussed in chapter 3, which simplifies the complexity of implementation.

As future work, we consider it is crucial to perform a more rigorous parameter optimization using, for example, an exhaustive grid-search strategy. Also, it is crucial to building a more diverse evaluation dataset that is not limited to news like the Youtube dataset or historical

videos such as OVP, which brings more information about the performance of representative VSUMM schemes.

5. Interestingness: Visual and Categorical Diversity

Video summarization is a semantic task, and it can be observed that semantics is mainly expressed in words [6], for which it is necessary to represent video content in terms of visual and linguistic information. For query-based VSUMM, it is necessary to work with linguistic information, i.e., we need to represent query inputs made by a user and combine them with visual information from the input video. The main problem with this approximation lies in the combined representation of this multi-modal information (images and words). We expect to construct a joint-representation architecture that uses pre-trained deep neural networks architectures.

In this chapter, we propose an experimental design to select best pre-trained deep neural network architectures and word-embeddings for visual representation and categorical representation, respectively, for a generic task of video summarization. Concretely we are given a set of DCN architectures pre-trained on the image-net dataset and a set of word-embeddings pre-trained on Wikipedia+Gigaword datasets. Then, we want to select the combination of DCN architectures and word-embeddings that have the best response concerning a task of video summarization.

For this purpose, we use a data-driven approach. We designed and developed a greedy model (see Figure **5-1**) that takes as input a video and generate a summary using a hybrid representation of visual and categorical models, and measure its response for a human predicted score of importance from the SumMe dataset. Finally, we select the best model using the F-Score measure.

5.1. Related work

VSUMM can be treated as a regression or ranking problem where some features are extracted from video-frames and used as inputs, and a set of key-frames or user-annotated scores [2, 53, 69, 59, 33, 10] as outputs. Earlier approaches focused exclusively on supervised visual features extracted from video [32, 33, 94] as SIFT, HoG or optical flow, among

others. Lee et al., in [7], used features as eye fixation, object frequency, and interaction to predict scene importance. Depending on the nature of the video or the search task, domain-specific features may be an aid in VSUMM. For example, game-specific rules for sport video analysis was proposed by Shih et al., in [99], actor recognition [100], and subtitle analysis [39] for movie summarization. Egocentric video analysis lately emerged as another significant VSUMM context, because of its characteristic high volume and diversity, which has motivated researchers to propose general VSUMM methods that could complement visual information with associated annotations and meta-data. Recently, deep learning has been applied to VSUM from multiple approaches. Otani et al. [59] proposed a method to train a coordinated representation space from a video-to-text dataset and after that, they used it to generate a regression model for VSUMM. Temporal analysis using long-short term memory networks (LSTM) and transfer learning from DCN (Deep Convolutional Network) was used in [15]. Generative adversarial networks (GANs) were used in [101] to formulate the VSUMM problem as a generator/discriminator challenge, where the generator selects the best frames (summary) from the input video and reconstruct it from these video frames. Then the discriminator compares the input video and the reconstructed video regarding this comparison as a classification problem. For this purpose, an architecture based on DCN and LSTM was constructed.

It is possible to approximate a frame importance parameter by its uniqueness or diversity for a group of frames. Uniqueness is related to the dissimilarity or difference of descriptors (*e.g.*, color histogram) in consecutive frames [3]. Classical approaches consider a processing pipeline where video frames are first pre-processed to improve quality, after which they are represented using a static set of descriptors [52]. These descriptors are mainly low-level visual features, i.e., color, textures, or histograms. Finally, a supervised criterion is designed using specific descriptors to estimate an importance score that allows selecting frames for the resulting summarized video. This approximation has some limitations that impair the possibility of using multi-modal information. In particular, the use of hand-crafted descriptors, and also the criterion of importance, have a high impact on the resulting summary, due to their *ad-hoc* nature.

Human attention is another information source closely related to diversity and commonly used for video summarization. Visual-auditive saliency and attention have been explored for the VSUMM task in [42, 102]. Varini et al., in [49], used a combination of HMM and diversity from Bag of Words difference between consecutive segments to create a video summary. Gygli et al. [2] represented attention for video summarization as a nonlinear combination of spatial features and temporal salience expressed as temporal differences.

Nevertheless, many claim that frame importance cannot be done entirely without previously known context information, including recording purpose, user preferences, and overall history

contained in the video, among other things [2]. Personalizing is one of the most recent topics of interest in video summarization because different users will summarize a video differently based on their specific interests. Initial approximations explored this venue by capturing more inputs from the user while doing this task, for example, gaze-tracking [51], BCI devices [50, 48], or states of attention [49]. These works have the limitation that user intention is not known previous to the video summarization task, and also require extra equipment.

Recent work focused on introducing queries in natural languages during the VSUMM process, as a means to specify a specific purpose. These queries may be expressed either as a vector of words of interest, which can be a sentence in natural language [46] or as a set of categorical terms (objects of interest) [14, 4, 5]. The first approximation requires NLP techniques to transform an arbitrary sentence into a manageable structure. The second approximation represents queries as a vector of words related to the objects of interest for the user. However, as far as a through exploration of recent advances in VSUMM may reveal, diversity from combined deep visual and categorical features has not been previously used. Also, the possibility to deliver personalized video summaries guided by a user query is an important feature that is certainly sought for in popular video repositories like YouTube and others. For these reasons, we propose a novel VSUMM method based on a categorical diversity estimation found combining both visual features and semantic categories inferred by user queries. The direct nature of our method without the requirement of a training stage allows rapid adoption and implementation for industrial and commercial applications.

5.2. Model

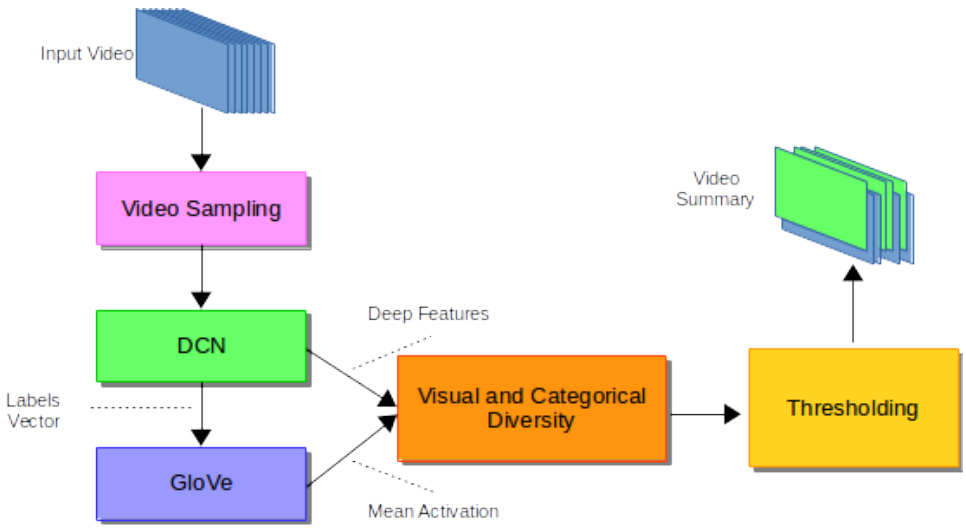

Figure 5-1.: Graphical scheme of proposed greedy model.

Overview. We are given a set of videos paired with human-made importance score annotations ([2]). Then, we need to use a ranking measure that allows us to compare how close is a machine-generated video summary scores and human-based scores for a given video V_i.

Classical approaches consider a processing pipeline where video frames are first pre-processed to improve quality and posteriorly described using a set of hand-crafted descriptors [52], mainly composed of low-level visual features, i.e., color, textures, histograms, features among others. Finally, a hand-crafted criterion is designed with selected descriptors to estimate an importance score that allows selecting frames for video-summary. This approximation has some limitations in terms of description and importance criteria stages and scaling using multi-modal information. For example, the selection and extraction of hand-crafted descriptors have a high impact on the performance of the VSUMM method due to the non-general purpose nature of these kinds of descriptors. The same problem is commonly found in the importance criteria stage.

Our method simplifies this pipeline using a pre-trained DCN architecture as a general visual descriptor of V_i extracting deep-features [64] from internal layers. Then, using the last layer of DCN we can obtain words related to detected categories that appear in V_i, which we represent semantically using a pre-trained word-embedding. Finally, a simple criterion of mutual penalization of visual and categorical representation is constructed. A graphical depiction of the proposed method is shown in Figure **5-1**.

We show that **1)** our method can combine information from visual and categorical nature for video summarization task, 2) visual and categorical information extracted from V_i using a DCN architecture are not necessarily correlated and 3) different combinations of DCN architectures and word-embeddings generates different scores with respect to human-annotated scores for V_i.

5.2.1. Visual representation

As mentioned in [64], using the right set of features, almost any AI problem can be solved. Notably, visual data representation is a complex problem due to the non-structured high-dimensional nature of images. Deep learning allowed not only to map from an input (stimulus) to output (response) and discover the best representation of the input. Literature named this approach as representation learning [64] or deep features, which consists of using a hidden layer of a pre-trained network model M as a general representation or descriptor of an input data.

For visual data representation, it is common to use DCN architectures previously trained for a classification task. The main idea behind this approach is that it is possible to transfer learned knowledge from a previous task to accelerate training for a new task.

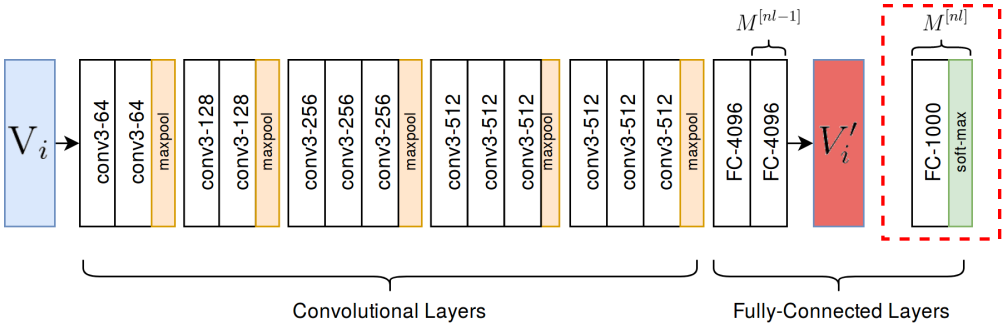

Figure 5-2.: VGG16 architecture. We compute $M^{[nl-1]}$ activation as a descriptor of input frame V_i.

We use a pre-trained DCN model M using image-net dataset [44], a as feature extractor for a video-frame V_i. For example, VGG-16 [63] model, has 16 layers and approximately 130 millions of parameters. Penultimate layer $M^{[nl-1]}$, i.e., layer before softmax 1000-dimensional probabilities of image-net categories, is 4096-dimensional. Then, we describe every frame V_i of input video V as follows:

$$V_i' = M^{[nl-1]}(V_i)$$

We compute penultimate layer activation for a DCN model M with nl number of layers, given a video-frame V_i. For this, we remove the layer of DCN model and compute a feed-forward propagation through the network. A graphical depiction of this representation scheme is shown in **5-2**.

5.2.2. Categorical representation

The complete forward propagation of a pre-trained DCN model generates an activation from the last layer $M^{[nl]}(V_i)$ given an input video frame V_i. This activation consists of a 1000-dimensional vector with the probabilities (softmax output) for every ImageNet category to appear in V_i.

Using this vector, we select first k categories indexes, i.e., the indexes of the categories with the highest probability to appear in V_i. Let define these indexes as an array of values γ. Then, each index in γ is transformed to a one-hot encoding representation t of size $[nd, 1]$ using word-embedding matrix vocabulary, where nd is the dimension of the embedding representation.

Finally a dot product between word-embedding matrix B and one-hot vector t is computed to obtain an embedded representation E_j for a visual category j from image-net (see Figure **5-3**).

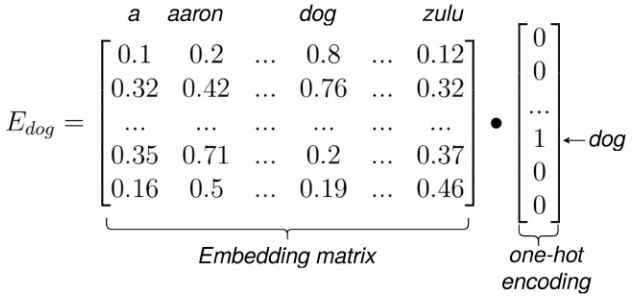

Figure 5-3.: Example of word embedding representation for word *dog* using one-hot encoding vector.

Using word-embeddings E_j for each category j, we compute distributed representation G_i for input video-frame V_i as the mean distributed representation of E in order to obtain a unidimensional vector that best describes the input video-frame V_i, i.e., that maintains the main direction of labels in V_i.

Since each category j in video frame V_i is associated with a probability from the softmax layer $M^{[nl]}$, we compute a weighted average representation as similar as Oosterhuis et al. in [5].

$$G_i = \frac{1}{k}\sum_{j}^{k} \psi_j E_j$$
$$\text{where} \tag{5-1}$$
$$\psi_j = M_j^{[nl]}(V_i)$$

In Equation 5-1 it can be observed the weighted average word embedding representation G_i for video-frame V_i. Notice that categories probabilities ψ are obtained from last layer from DCN model M.
A complete sequence to obtain categorical representation G_i is presented in Algorithm 2.

5.2.3. Visual and categorical diversity

Estimation of importance for a frame or a group of frames inside a video is a task that can not be done entirely without previously known context information (intention of the recording, context, and user preferences) and overall history contained in video [2]. This task is commonly formulated as a regression or ranking problem, where some features extracted from each video-frame are used as inputs, and human-made scores of importance are used

Algorithm 2: Categorical representation of V_i using embedding matrix.

Input : video frame V_i, embedding-matrix B, DCN model M, k first visual categories

Output: weighted average word-embedding activation G_i

begin
- $\gamma = argsort(M^{[nl]}(V_i))[0:k]$;
- $E = zeros([nd, 1])$;
- **for** $j \leftarrow 0$ **to** $k-1$ **do**
  - $t = zeros([nd, 1])$;
  - $t[\gamma[j]] = 1$; `// one-hot encoding`
  - $E_j = B \cdot t$; `// embedding representation`
  - $\psi_j = M_j^{[nl]}(V_i)$; `// weight from softmax layer`
  - $G_i = G_i + \psi_j E_j$; `// weighted sum`
- **end**
- $G_i = \frac{1}{k} G_i$; `// weighted average`
- **return** G_i;

end

as outputs [10]. However, it is possible to approximate importance by the uniqueness or diversity of a frame or group of frames. Uniqueness is related to the dissimilarity or difference of features, e.g., color histogram, for consecutive frames [3].

Attention is another information closely related to diversity and commonly used for video summarization. Varini et al., in [49, 53], used a combination of HMM and diversity from Bag of Words difference between consecutive segments to create a video summary. Gygli et al., in [2] represented attention for video summarization as a nonlinear combination of spatial features and temporal saliency expressed as temporal differences. We use diversity as an indirect approach from which we can obtain an interestingness score from which to obtain a summary given an input video V_i. Using the concept of diversity expressed as temporal differences, we obtain visual diversity D_i for an input video-frame V_i as follows:

$$D_i = \left\| \frac{dV_i'}{dt} \right\| = \left\| \frac{V_{i+\Delta t}' - V_i'}{\Delta t} \right\|$$

Where V_i' is the deep-features vector extracted from the penultimate layer of DCN model, as explained previously. Δt is 1 because the input layer is previously sampled to $\frac{1 frame}{second}$. Notice that we compute the L2-norm of the derivative in order to obtain a scalar-valued visual diversity. Using the same approach as visual diversity D_i we compute categorical diversity K_i as temporal differences from categorical representations G_i.

$$K_i = \left\| \frac{dG_i}{dt} \right\| = \left\| \frac{G_{i+\Delta t} - G_i}{\Delta t} \right\|$$

Finally, we scale D_i and K_i in the range $[0, 1]$ in order to avoid magnitudes differences from visual and categorical representations V_i and G_i.

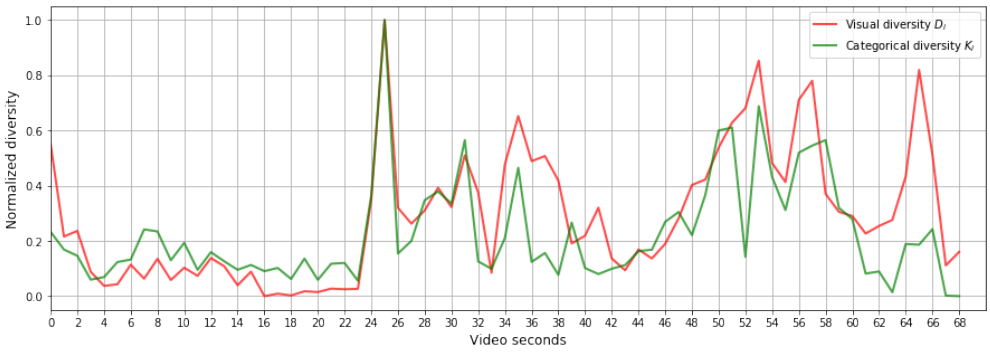

Figure 5-4.: Normalized responses of visual diversity D_i and categorical diversity K_i for video *St. Marteen Landing* from the SumMe dataset [2].

In Figure **5-4** it can be observed the visual diversity D_i and categorical diversity K_i for video *St. Marteen Landing* from dataset SumMe, using DenseNet as DCN model and GloVe-100 as a word-embedding. In seconds 39 and 52, there exists a negative correlation between both diversities, indicating that a sequence of frames can have inverse proportions for D_i and K_i.

5.2.4. Combined visual and categorical diversities

Visual and categorical diversity, D_i and K_i, respectively, for a video frame V_i are related but measure different domains from video, that is, for a given video frame V_i we can obtain a high visual diversity and low categorical diversity or the contrary. Then, we can assume that we need to have both high visual diversity D_i and high categorical diversity K_i in order to consider a frame to have a high diversity ϑ.

In other words, if a video frame is visually diverse but not categorically, then it is penalized and the same for the contrary. From this analysis we balance visual and categorical diversity by a linear relation using coefficients c_0 and c_1. In equation 5-2, it is shown visual diversity ϑ_i expressed in terms of D_i and K_i. Notice that if $c_0 = c_1 = 0.5$ then we represent ϑ as the mean of visual and categorical diversity.

$$\vartheta_i = c_0 D_i + c_1 K_i \tag{5-2}$$

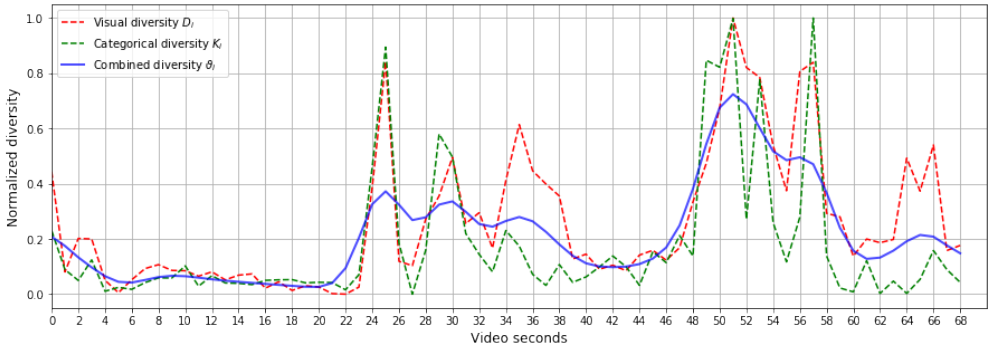

Figure 5-5.: Gaussian smoothed (σ =1.5) and normalized combined diversity ϑ_i with respect to D_i and K_i for video *St. Marteen Landing* from database SumMe [2].

Finally, we apply a Gaussian smooth to ϑ in order to generate a more continuous diversity along video seconds. The latter will allow us to generate video summaries with soft transitions between segments. In Figure **5-5** it is shown the combined diversity ϑ with respect to D_i and K_i for video *St. Marteen Landing* from database SumMe. As explained previously, inverse activations from D_i and K_i mutually penalize each other, for example, in seconds 44 and 52. On the contrary, both diversities enhance ϑ as in seconds 25 and 56. It is important to highlight that single magnitudes ϑ_i are not important by itself but as concerning each other, i.e., scaling ϑ does not change the summary result.

5.2.5. Summary generation

We generate a summary S of duration $\alpha|V|$ seconds, where α is user-defined parameter usually equal to 0.15 or 15 % of input video (V) length in seconds. For that purpose, we apply thresholding to ϑ, as described in Equation 5-3.

$$\begin{aligned} &S(\alpha) = V_j \mid \vartheta_j > \tau \\ &\quad \text{restricted to:} \\ &(|S| - \alpha\,|V|) \to 0 \end{aligned} \tag{5-3}$$

Where $|S|$ and $|V|$ are the lengths of S and V respectively. Then, we need to find a value τ such that the number of all frames with diversity ϑ_j greater than τ, is closely to $\alpha|V|$. We use a loop from $\tau = 1 \to 0$ using a small step $\Delta\tau$ until we accomplish previous restriction.

In Figure **5-6** it is illustrated the summary generation process. Notice that the dashed red line illustrates the value of τ, which was found using an iterative process, as explained previously. Blue shaded region illustrates the subset of frames with higher ϑ, which constitute a summary

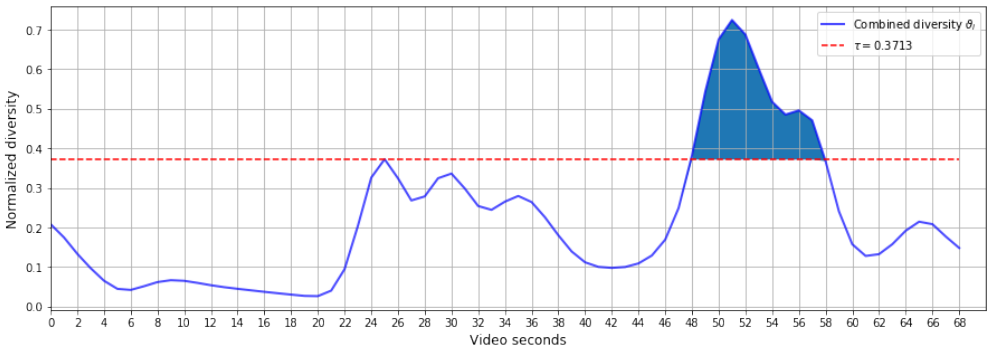

Figure 5-6.: Summary generation from ϑ_i for video *St. Marteen Landing* from database SumMe [2], using the proposed method. Frames V_i under shadowed (blue) region will be used as a summary of input video.

of 15 % of input video length $|V|$.

5.2.6. Query injection

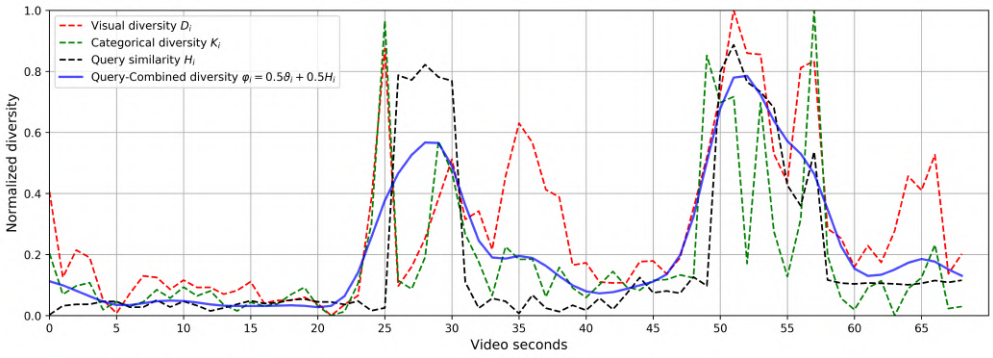

Figure 5-7.: Video diversity biased by query similarity for $q = \{airship, aircraft\}$ over video *St. Maarten Landing* from dataset SumMe. Query similarity in black, visual diversity in red and categorical diversity in green.

We represent an user query as a vector $q = \{w_0, w_1, w_2, ..., w_n\}$ of words w. In order to avoid direct-match between q and visual categories j detected by DCN model M, each word of the query is mapped to a vector space using a word-embedding matrix E as explained in previous sections, in a similar manner as in [5]. But, as opposed to that proposal, we assume that all words in the query have the same importance for the user, and for this reason, we do not weight words w in the query q. Then, we calculate the average of word-vectors for k words in query as follows:

$$G_{query} = \frac{1}{k}\sum_{j}^{k} E_j \tag{5-4}$$

The categorical representation G_i (section 5.2.2) extracted from video-frame V_i, allows us to compare categorical similarity H_i of q and V_i in a direct manner using the cosine similarity as follows:

$$\begin{aligned} cos(w_i, w_j) &= \frac{w_i \cdot w_j}{\|w_i\| \, \|w_j\|} \\ H_i &= cos(G_i, G_{query}) \end{aligned} \tag{5-5}$$

Finally, we can represent query-combined diversity φ as a linear relation between combined diversity ϑ_i and query similarity S_i as follows:

$$\begin{aligned} \varphi_i &= c_0\vartheta_i + c_1 H_i = \\ \varphi_i &= c_0 D_i + c_1 K_i + c_2 H_i \end{aligned} \tag{5-6}$$

In Fig. **5-7** it is shown an example of query injection to find combined diversity ϑ using Eqs. 5-4, 5-5, and 5-6.

5.3. Experiments

In the following experiments, we first compare visual and categorical diversity from different combinations of DCN models to determine if there are significant differences between them. Then, each combination is evaluated using SumMe dataset and a performance criterion defined later in this chapter, and we select the model with the best performance. Finally, we compare the best model with respect to state-of-the-art works to give an idea of future performance using a joint representation architecture.

5.3.1. Data

SumMe: Video to user-importance dataset. Proposed and published by Gygli et. al. in [2], this dataset contains 25 videos, organized by three categories: *egocentric*, *moving* and *static*. For each video, scores per segment were manually annotated by different individuals.

This score is related to a scale of importance in the range $[0, 1]$ (see Figure **5-8**), for minimum and maximum degree of importance for the user, respectively. The authors report that this was the first dataset with segment annotations rather than key-frames. This dataset has been widely used for different video summarization approaches on literature, allowing us to compare with respect to other authors.

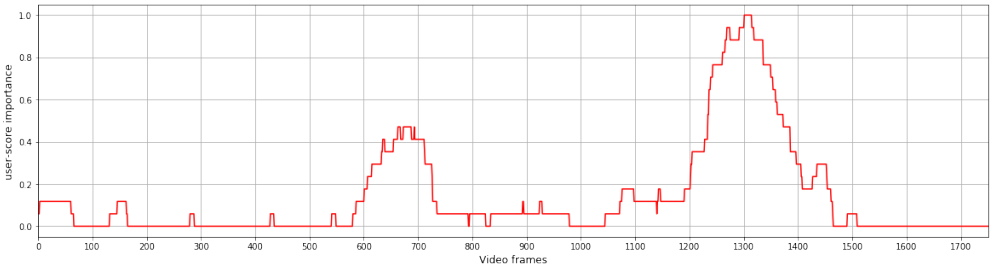

Figure 5-8.: Averaged user-importance score for video *St. Marteen Landing* from database SumMe [2].

Video Preprocessing. Input video V is uniformly sampled to one frame per second as similar to related works [59, 15, 6]. We use this approach in order to reduce computational processing and due to the need that our greedy model must as simple as possible, avoiding alternatives like clustering or super-frame segmentation [2].

Figure 5-9.: Input video V is sampled to one frame per second.

Videos from the SumMe dataset mostly have a 30 fps rate. We consider that at this frame

rate will not occur a significant visual and categorical diversity in less than a second. Then, we take 1-frame per second in order to reduce processing time, that is, $|V_{sampled}| = \frac{|V|}{fps}$. In Figure **5-9** it is shown a graphical depiction of this process.

5.3.2. Performance criteria

Evaluation of a generated summary using the SumMe dataset consists of performing a measure of test's accuracy between video summaries extracted from estimated diversity ϑ and video summaries extracted from estimated importance given by N human users. For this purpose, Gygli et. al. [2] proposed the use of pair-wise *f-measure* evaluation metric between a generated summary $S(\alpha)$ and human-made scores $U(\alpha)$ as expressed in Equation 5-7.

$$F(S,U,\alpha) = \frac{1}{N-1}\sum_{j=1}^{N} 2\frac{p_{(S(\alpha),U_j(\alpha))}r_{(S(\alpha),U_j(\alpha))}}{p_{(S(\alpha),U_j(\alpha))} + r_{(S(\alpha),U_j(\alpha))}} \tag{5-7}$$

Where $p_{(S(\alpha),U_j(\alpha))}$ and $r_{(S(\alpha),U_j(\alpha)}$ are precision and recall between generated summary and interestingness score made by user j. Notice that the final score is the averaged result of the summary S with respect to each user annotation U_j at $\alpha|V|$ length of the original video. This performance criterion has been adopted as the evaluation standard in video summarization as a prediction problem [59, 15, 103], allowing us to compare with respect to state-of-art methods without the need of replicating third-party works.

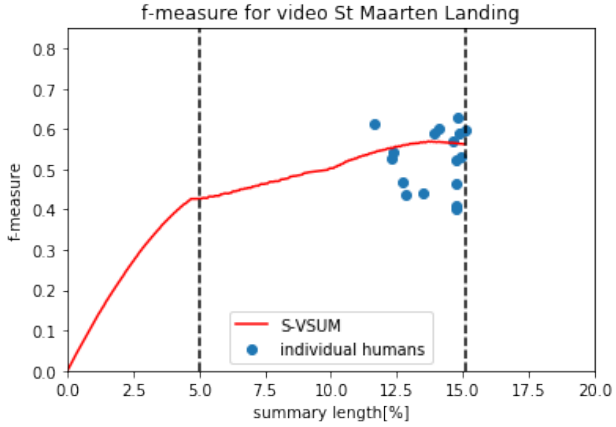

Figure 5-10.: F-measure for *St. Marteen Landing* from database SumMe [2] using our method and VGG19 as DCN model.

Figure **5-10** shows the evaluation of our method for a particular video from the SumMe dataset using VGG19 as DCN model. Notice that between 12 % and 15 % of the original video's length, our method reaches a performance similar to the average of human-users.

5.3.3. Baselines

We evaluate our method with respect to the following approximations:

- **Random sampling**: Video summarization by taking random frames is commonly used as a base comparison on literature because any proposed video summarization method must be superior to this approach.
- **Interestingness-based [2]**: Original work by Gygli et al., where it is proposed SumMe dataset and a video summarization method based on a regression model that uses a combination of features related with frame-interestingness: attention, aesthetics, presence of landmarks, faces, and object tracking, to predict per-frame importance.
- **Deep semantic features [59]**: As proposed by Otani et al., this method uses coordinated representations which authors refer as semantic features, trained over a video-to-text dataset. These representations are then used in a regression model to predict per-frame importance.

5.3.4. Results

Visual and categorical diversity relationship

As explained in previous sections, our method lies in the use of a pre-trained DCN and word-embedding models for extracting visual and categorical diversity, respectively. Visual diversity D depends exclusively on activations from DCN model. Categorical diversity K depends on DCN model and word-embedding activations. In this order of ideas, it is possible to ask the following questions:

- Will different DCN models generate different/similar visual diversity D?
- Will Different DCN models generate different/similar categorical diversity K?
- Will visual and categorical diversity D and K, be related for each DCN model?

To answer these questions, we measured correlation coefficients for visual and categorical diversity through each video in dataset SumMe, using different DCN models. In tables **5-1** and **5-2** it can be observed the mean correlation for D and K respectively.

Notice in table **5-1** that, in general, correlation coefficients for D are higher than 0.7, which can be interpreted as that visual diversity is strongly related across different DCN models. In other words, a similar visual diversity is expected to be obtained when using any of the evaluated DCN models.

Table 5-1.: Correlation coefficients for visual diversity D using different DCN models.

Model (Visual Diversity)	VGG16	VGG19	Xception	InceptionV3	ResNet50	InceptionResNetV2	DenseNet
VGG16	1.0	0.931	0.802	0.756	0.871	0.706	0.859
VGG19	0.931	1.0	0.797	0.754	0.864	0.705	0.862
Xception	0.802	0.797	1.0	0.776	0.811	0.753	0.816
InceptionV3	0.756	0.754	0.776	1.0	0.782	0.749	0.783
ResNet50	0.871	0.864	0.811	0.782	1.0	0.728	0.882
InceptionResNetV2	0.706	0.705	0.753	0.749	0.728	1.0	0.748
DenseNet	0.859	0.862	0.816	0.783	0.882	0.748	1.0

It is also possible to observe that models of the same family such as VGG16/VGG19 and DenseNet/ResNet50 will have a much stronger correlation for visual diversity.

Table 5-2.: Correlation coefficients for categorical diversity K using different DCN models.

Model (Categorical Diversity)	VGG16	VGG19	Xception	InceptionV3	ResNet50	InceptionResNetV2	DenseNet
VGG16	1.0	0.527	0.295	0.225	0.332	0.246	0.301
VGG19	0.527	1.0	0.289	0.262	0.351	0.280	0.342
Xception	0.295	0.289	1.0	0.280	0.284	0.266	0.327
InceptionV3	0.225	0.262	0.280	1.0	0.238	0.300	0.276
ResNet50	0.332	0.351	0.284	0.238	1.0	0.234	0.369
InceptionResNetV2	0.246	0.280	0.266	0.300	0.234	1.0	0.293
DenseNet	0.301	0.342	0.327	0.276	0.369	0.293	1.0

Correlation coefficients for categorical diversity K can be observed in **5-2**. Notice that, in general, categorical diversity correlation across DCN models is below 0.4, which can be interpreted as a weak relation between models. In other words, the selection of a particular DCN model will result in a different categorical diversity.
Similarly to visual diversity, models of the same family such as VGG16/VGG19 and DenseNet/ResNet50 will have the strongest correlations for categorical diversity.

Table 5-3.: Correlation coefficients between visual diversity D and categorical diversity K using different DCN models.

DCN Model						
VGG16	**VGG19**	**Xception**	**InceptionV3**	**ResNet50**	**InceptionResNetV2**	**DenseNet**
0.554	0.555	0.512	0.504	0.471	0.587	0.506

Finally, we obtained correlation coefficients between D and K for each evaluated DCN model, as can be observed in table **5-3**. For any DCN model, Visual and categorical diversity have a low relationship, as correlation coefficients are 0.5 approximately. In other words, as explained in previous sections, although D and K are related and depends of a DCN model, we can expect that for a given input video-frame V_i we can obtain a high value of D_i and low K_i or the contrary. Thus, we can expect the use of combined diversity ϑ in a video summarization task, will generate a similar response to human users.

Evaluation between DCN models

We evaluated the performance of our model using different pre-trained DCN models over SumMe dataset videos. Evaluation was made using f-score as presented in equation 5-7.

Table 5-4.: F-measures for each DCN model using combined diversity ϑ (higher is better). For each video in SumMe dataset, we show best result (bold). Finally we show the mean of the f-measures obtained by each DCN model.

		Computational Method ($\vartheta_i = \frac{1}{2}(D_i + K_i)$): Evaluation of DCN models						
Category	**Videoname**	**VGG16[63]**	**VGG19[63]**	**Xception [79]**	**InceptionV3 [78]**	**ResNet50 [80]**	**InceptionResNetV2**	**DenseNet [82]**
Egocentric	Base jumping	0.182	0.114	0.187	**0.200**	0.175	0.174	0.166
	Scuba	0.142	**0.230**	0.300	0.112	0.126	0.170	0.182
	Bike Polo	0.100	0.193	0.225	**0.290**	0.076	0.153	0.219
	Valparaiso_Downhill	**0.306**	0.283	0.216	0.260	0.235	0.203	0.233
Moving	Bearpark_climbing	0.088	0.107	0.173	0.195	0.133	**0.229**	0.147
	Bus_in_Rock_Tunnel	0.081	0.083	0.109	0.101	0.089	0.091	**0.117**
	Car_railcrossing	**0.117**	0.130	0.037	0.123	0.066	0.080	0.058
	Cockpit_Landing	0.140	0.139	0.124	0.126	0.189	**0.190**	0.147
	Cooking	0.075	0.128	0.132	0.204	**0.266**	0.207	0.265
	Eiffel Tower	**0.219**	0.152	0.147	0.101	0.144	0.135	0.129
	Excavators river crossing	0.092	0.118	0.086	0.081	0.056	**0.098**	0.089
	Jumps	0.063	0.049	0.051	0.175	0.040	0.044	**0.387**
	Kids_playing_in_leaves	0.339	0.263	**0.415**	0.319	0.390	0.221	0.196
	Playing_on_water_slide	0.040	0.046	0.048	0.074	0.055	**0.115**	0.047
	Saving dolphines	0.116	0.113	0.066	0.114	0.126	0.120	**0.165**
	St Maarten Landing	0.581	0.563	0.557	0.469	**0.610**	0.504	0.552
	Statue of Liberty	0.082	0.103	0.116	0.114	0.093	**0.152**	0.123
	Uncut_Evening_Flight	0.216	0.178	0.269	**0.300**	0.116	0.248	0.153
	paluma_jump	0.114	0.100	0.104	**0.259**	0.255	0.120	0.106
	playing_ball	0.142	0.092	0.152	0.097	0.053	**0.190**	0.154
	Notre_Dame	0.105	0.103	0.128	0.137	**0.144**	0.113	0.139
Static	Air_Force_One	0.375	**0.385**	0.263	0.348	0.356	0.145	0.222
	Fire Domino	0.094	**0.231**	0.155	0.103	0.215	0.121	0.099
	car_over_camera	0.426	0.414	**0.436**	**0.436**	0.435	0.413	0.414
	Paintball	**0.485**	0.471	0.478	0.480	0.378	0.423	0.432
mean score		0.189	0.192	0.199	**0.209**	0.193	0.186	0.198

In table **5-4**, it can be observed the score by video in the SumMe dataset, using our method with different popular DCN pre-trained models. InceptionV3 and InceptionResNetV2 models obtains the highest f-score (bold) for most videos (6 videos each model). In terms of mean or averaged score, **InceptionV3** is the model with the highest f-score and will be used as a base DCN for comparison with state-of-art works. It is important to mention that although InceptionV3 is the best DCN model in general terms, the mean f-score for each evaluated DCN model is similar to the others, with 0.186 as the lowest f-score and 0.209 as highest f-score.

State-of-art comparison

In table **5-5** it is shown the results of quantitative evaluation for our method and different computational methods as explained in section 5.3.3. We showed scores for human annotators as reported by the author in [2] as follows:

- **Minimum score (Min.)**: The lowest score of all human annotators with respect to mean

score.

- **Mean score (Mean)**: Average of scores made by each human annotators with respect to others. For example, if a video is annotated by 20 users, then the f-score is computed for each annotator with respect to the others (19). Finally, all previous are averaged.
- **Maximum score (Max.)**: Highest score by human annotators with respect to mean score.

Table 5-5.: F-measures for different computational methods (higher is better). For each video in SumMe dataset, best results are shown in bold. Finally we show the mean of the f-measures obtained by each computational method.

		Human Annotators			Computational Method			
Category	**Videoname**	**Min**	**Mean**	**Max**	**Random**	**Gygli (2014)**	**Otani (2016)**	**Ours (DCN: Inception V3)**
Egocentric	Base Jumping	0.113	0.257	0.396	0.144	0.121	0.077	**0.200**
	Bike Polo	0.190	0.322	0.436	0.134	**0.356**	0.235	0.290
	Scuba	0.109	0.217	0.302	0.138	**0.184**	0.154	0.112
	Valparaiso Downhill	0.148	0.272	0.400	0.142	0.242	0.258	**0.260**
Moving	Bearpark climbing	0.129	0.208	0.267	0.147	0.118	0.178	**0.195**
	Bus in Rock Tunnel	0.126	0.198	0.270	0.135	0.135	**0.151**	0.101
	Car Rail Crossing	0.245	0.357	0.454	0.140	**0.362**	0.328	0.123
	Cockpit Landing	0.110	0.279	0.366	0.136	**0.172**	0.165	0.126
	Cooking	0.273	0.379	0.496	0.145	0.321	**0.329**	0.204
	Eiffel Tower	0.233	0.312	0.426	0.130	**0.295**	0.174	0.101
	Excavators River Crossing	0.108	0.303	0.397	0.144	**0.189**	0.134	0.081
	Jumps	0.214	0.483	0.569	0.149	**0.427**	0.015	0.175
	Kids Playing in Leaves	0.141	0.289	0.416	0.139	0.089	0.278	**0.319**
	Playing on Water Slide	0.139	0.195	0.284	0.134	**0.200**	0.183	0.074
	Saving dolphines	0.095	0.188	0.242	0.144	**0.145**	0.121	0.114
	St Maarten Landing	0.365	0.496	0.606	0.143	0.313	0.015	**0.469**
	Statue of Liberty	0.096	0.184	0.280	0.122	**0.192**	0.143	0.114
	Uncut Evening Flight	0.206	0.350	0.421	0.131	0.271	0.168	**0.300**
	Paluma Jump	0.346	0.509	0.642	0.139	0.181	**0.428**	0.259
	Playing Ball	0.190	0.271	0.364	0.145	0.174	**0.194**	0.097
	Notre Dame	0.179	0.231	0.287	0.137	**0.235**	0.093	0.137
Static	Air Force One	0.185	0.332	0.457	0.144	0.318	0.316	**0.348**
	Fire Domino	0.170	0.394	0.517	0.145	**0.130**	0.022	0.103
	Car Over Camera	0.214	0.346	0.418	0.134	0.372	0.132	**0.436**
	Paintball	0.145	0.399	0.503	0.127	0.320	0.274	**0.480**
Mean		0.179	0.311	0.409	0.139	**0.234**	0.183	0.209
Relative to human avg.		58%	100%	131%	45%	**75%**	59%	67%
Relative to human max.		44%	76%	100%	34%	**57%**	45%	51%

For computational methods, we report the best scores per video in bold and performance relative to human annotators.

Our method obtains higher performance than the method proposed by Otani et al., with 67 % and 59 %, respectively, relative to average human performance.

The method proposed by Gygli et al. still obtains the highest mean f-score with performance with respect to the human average of 75 %. Also we obtained bests scores in $\frac{9}{25}$ videos, Gygli in $\frac{12}{25}$ videos, and Otani in $\frac{4}{25}$ videos.

Performance by video category is presented in table **5-6**. Notice that our method obtains the highest score for *Static* category with a remarkable difference concerning the other computational approaches. *Moving* category represents the lowest score for our method and the higher difference in performance with respect to the method proposed by Gygli. Finally, in

Table 5-6.: Mean f-measure per video category for different computational methods (higher is better). For each video category, best result is shown in bold.

	Computational Method		
Category	**Gygli [2]**	**Otani [59]**	**Ours (DCN: InceptionV3)**
Egocentric	**0.226**	0.181	0.216
Moving	**0.225**	0.182	0.176
Static	0.285	0.186	**0.342**

Egocentric category, we obtain similar performance to the last model.

We consider it essential to mention that our method is more straightforward than computational methods proposed by Gygli and Otani in terms that relies on using a single DCN pre-trained model and word-embeddings, which do not require a training stage. Also, computes information in a feed-forward direction using deep features as transfer learning, allowing us to not depend on a set of hand-crafted features that can vary for different future applications.

Error analysis

Our method got low scores when videos contain complex stories in terms of actions and interactions where there is not high diversity in visual properties of V_i or objects on the scene. In other words, when video importance is not related to diversity but story. Examples of this kind of videos are *playing_ball*, *Excavators River Crossing*, and *Playing on Water Slide*, where our method obtains its lowest scores.

In figure **5-11**, we show examples of videos from dataset SumMe, with associated human scores of importance (red) and summary made by our method (blue). In video *playing_ball* main visual objects or categories are ball, dog, and bird. These objects are always present in video-frames, so it is expected a low categorical diversity. Moreover, although the video is moving, there are not important transitions or scene changes that generate high visual diversity. In this case, importance as annotated by users, is related with interactions between *bird*, *dog* and *ball*.

It is also important to consider that since our method relies on a pre-trained DCN model, the performance of summarization is related to the performance of our DCN model. In other words, if a DCN model does not correctly detect visual categories on a video frame, diversity will be affected. Possible approximations to solve this limitation consist in the using of action recognition approaches that helps mapping time-related phenomena like interactions.

On the contrary, in videos like *car_over_camera* and *St. Maarten Landing* (see figure **5-11**), importance is highly related to visual differences and changes of objects in scenes, which our method is able to capture as visual and categorical diversity. In these cases, we obtained

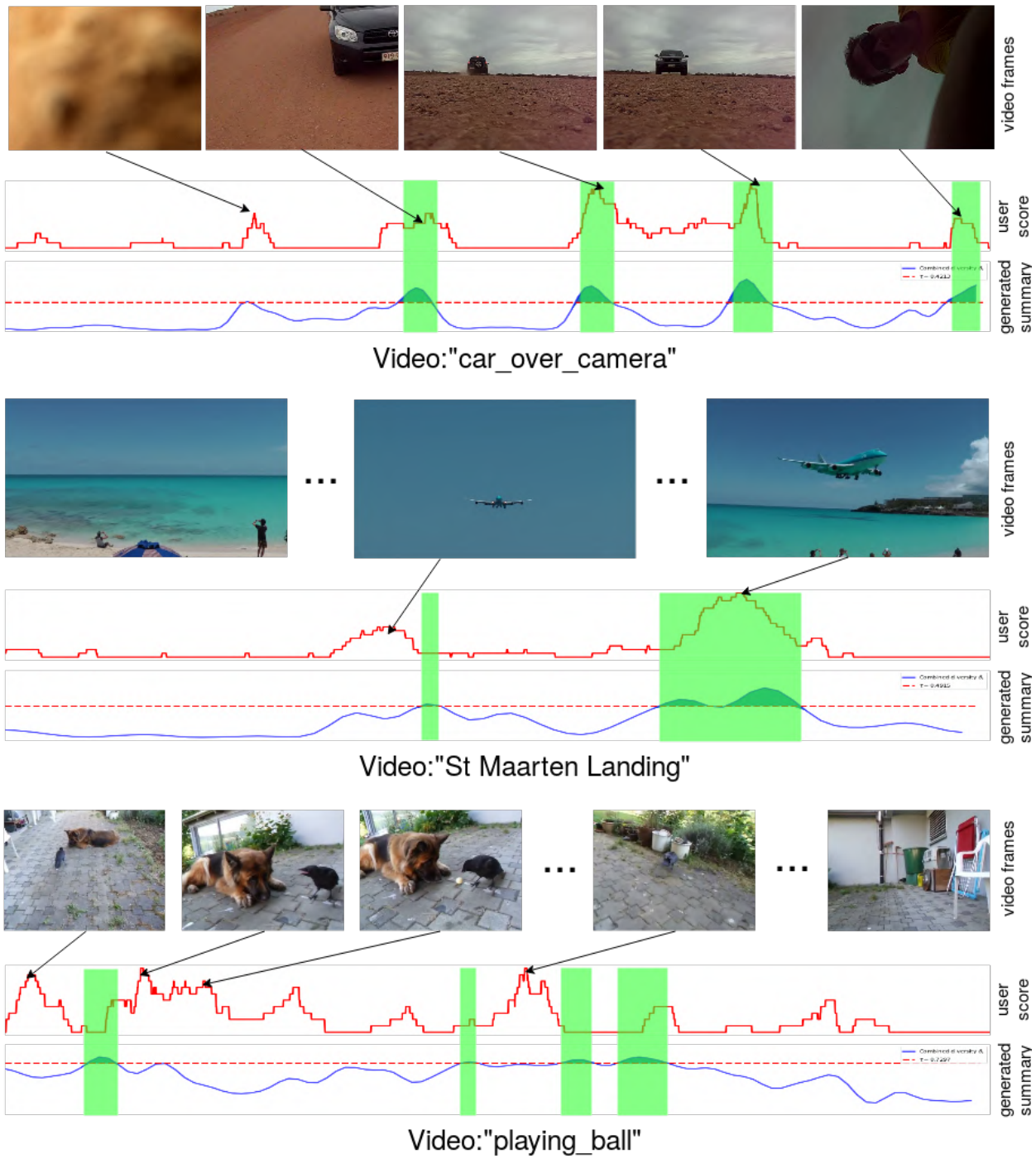

Figure 5-11.: **Example summaries**. Three video examples from dataset SumMe. For each video it is presented the mean user score in red, generated summary (at 15 %) by our method in blue, and intersection of generated summary and user scores in green. It is also shown, video frames with high importance to human annotators.

higher performance than other computational methods.

Table 5-7.: Query-injection examples for three videos on dataset SumMe. For each video it is shown score using combined diversity ϑ, query vector q and score using query combined diversity φ.

Videoname	**Query** q	**Global VSUM** ϑ	**Query-based VSUM** φ
Playing Ball	{german, shepperd, ball}	0.097	0.281
Playing on Water Slide	{water, kids}	0.074	0.183
St. Maarten Landing	{airship, aircraft}	0.469	0.553

Query as keywords can be included in the visual and categorical diversity to improve the performance of VSUMM task. In table **5-7**, it can be shown three examples of f-score improvement using a list of keywords to guide the diversity scheme, as previously discussed in section 5.2.6. Notice that in all cases, f-score improve over the general approach when used keywords injection.

5.4. Conclusions

We have developed an interestingness model based on a combined architecture of a pre-trained DCN model to represent visual information as visual diversity and word-embeddings to represent categorical information as categorical diversity, for the following purposes:

- Evaluate and select a pre-trained DCN model for the visual representation of video-frames.
- Evaluate the performance of a combination of visual and categorical (linguistic) representation for a video summarization task.
- Identify limitations and future work from an evaluation of a simple model based on visual and categorical representation for a video summarization task.

Experiments show that it is possible to use both visual and categorical representation in a combined fashion for a video summarization task. Although the simplicity of the constructed model in terms of its architecture and the linear relation of visual and categorical diversity, performance (f-score) is close to or superior to state-of-art works. This motivates us to continue exploring in this direction of research. Some conclusions we can make based on experiments and results are:

- Visual representation, in this case, visual diversity D, obtained from different DCN models is highly correlated, i.e., we can obtain a similar visual representation using any pre-trained DCN model.

- Categorical representation, in this case, categorical diversity K, obtained from different DCN models and GloVe word-embedding [27] presents a low correlation, i.e., categorical representation vary significantly for different DCN models.
- Although K depends on D, both representations are not highly related, i.e., a video frame can be highly visually diverse but not categorically, or the contrary.
- We obtained the best mean performance (f-score: 0.209) using InceptionV3 as a pre-trained DCN model.
- It should be possible to improve results of this combined approximation using a sophisticated extension of this approach in terms of **a)** represent temporal nature of video to improve performance in videos where importance is related with actions and interactions between elements on scene, **b)** use a multi-modal representation to combine visual and categorical representation, and **c)** allow the model to be trained in order to predict video importance as a regression model.

6. Framework Integration and Conclusions

6.1. Introduction

Sub-modular optimization is commonly used to optimize a multi-objective VSUMM [10], which requires defining each VSUMM objective as a sub-modular function. We also consider that a limitation of the sub-modular optimization approach is that it requires a training stage, making the final summary dependant of the data used and the separation strategy to obtain the set (x, y) in each case.

From the previous, we decided to use a more straightforward computational approach based on the knapsack optimization problem, which allows us to easily separate summaries from each objective (representativeness, interestingness, and importance), and also allows us to integrate them in a single summary.

In this chapter, we show the complete framework for video-summarization using video and language relations, combining what has been shown in previous chapters and integrating it in a single computational scheme.

6.2. Framework

In figure **6-1** it can be observed the integration framework developed. Our framework receives an input video V, and computes each VSUMM objective, as follows: **a) Representativeness and Uniformity**, as the k-medoids over coordinated visual-language features, unified by hierarchical clustering and a mass-center strategy, **b) Interestingness** by computing the visual diversity over a deep-feature space using InceptionV3, and the categorical diversity over a linguistic space using GloVe word-embedding, **c) Importance (categorical)**, computing the similarity between video frames and a query vector $q = [word_1, word_2, ..., word_n]$ of words over the GloVe space, and returning the most similar frames with respect to an arbitrary threshold, and **d) Importance (text)**, computing the similarity between video frames and a query vector $q = [sentence_1, sentence_2, ..., sentence_n]$ of free-form sentences, using an

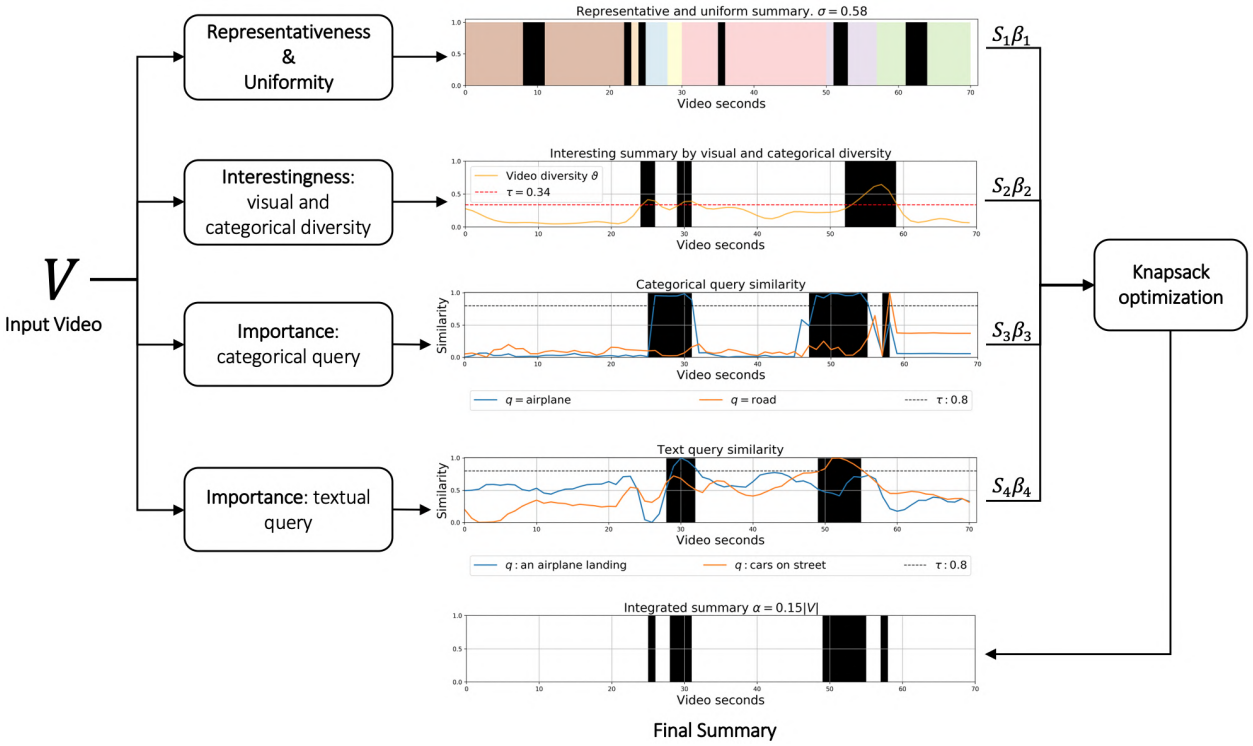

Figure 6-1.: Complete VSUMM framework. Integration scheme for multi-objective query-based VSUMM. In this example, all VSUMM objectives where optimized, i.e., all $\beta_i = 1$.

ad-hoc coordination model and Skip-thought unidirectional and bidirectional models.

From each VSUMM objective i, a binary array S_i can be computed, where 1 indicates that the video segment/frame/second must be included in the summary and 0 on the contrary.

Depending on the task or user intention, each VSUMM objective can be arbitrarily weighted using a scalar β_i, which in the binary case, can be 1 if the objective i will be considered in the final summary or 0 on the contrary.

Due to the binary nature of the final summary (select which frames will be preserved), and the intention to obtain the best possible summary which considers one or all VSUMM objectives, constrained to a length $\alpha|V|$, where α is a user-defined parameter, the integration can be treated as a knapsack problem optimization.

6.2.1. Knapsack problem optimization

The knapsack problem requires to define three inputs: A vector *weights*, which contains the associated cost of each element, a vector *values*, which contains an associated quality or

importance of each element, and a scalar *capacity* related with the problem constraint.

$$\begin{aligned} weights = \omega &= 1 \forall V_i \in |V| \\ values = \upsilon &= \sum_i \beta_i S_i \\ capacity = \tau &= \alpha |V| \end{aligned} \tag{6-1}$$

In terms of a VSUMM problem, we defined each input according to equation 6-1. As we employed a 1/frame per second sampling during all of the videos pre-processing, the vector *weights* is equal to one for all seconds in V. The vector *values* is the sum of all the VSUMM objectives; that is, the value of a video second is proportional to the total frequency of appearance in the summaries S. The use of the weight β_i also affects the value of the video seconds with respect to the objective i. Finally, the constraint τ is equal to a user-defined ratio α of the length $|V|$ of the original video.

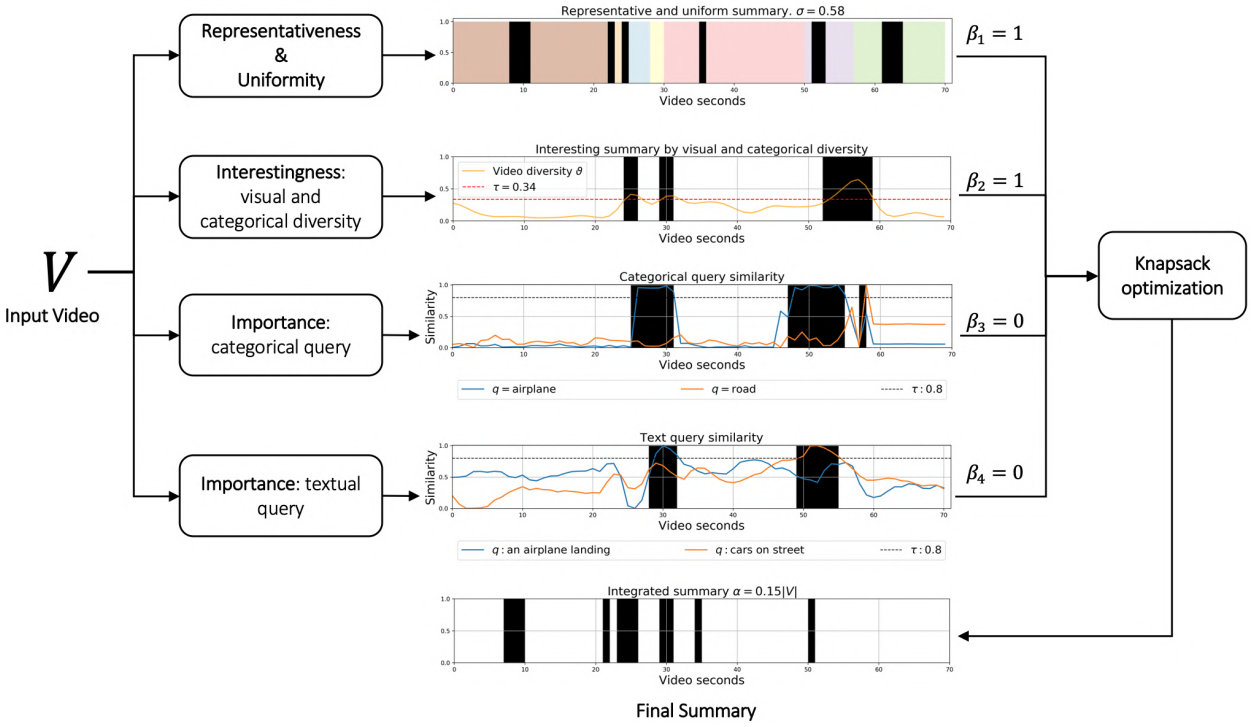

Figure 6-2.: Multi-objective VSUMM integration example, where only *Representative+Uniform* (S_1) and *Interestingness* (S_2) objectives are considered for the final summary.

Once defined all of the elements of the knapsack problem, the optimization can be expressed as in equation 6-2, where z is the final summary to be optimized, which is a binary array with value 1 for segments to be preserved and 0 otherwise. Notice that, although in our case,

$z_i\omega_i$ is equal to z_i, it is essential to include ω in the optimization, as another implementation can consider a non-uniform sampling of V.

$$\begin{aligned} &\max_{z} \sum_{i=1}^{|V|} v_i \\ &s.t. \sum_{i=1}^{|V|} z_i\omega_i \leq \tau \end{aligned} \tag{6-2}$$

In figures(**6-3**,**6-3**) it can be observed two examples of the knapsack problem optimization considering different combination of VSUMM objectives: *representative+uniform+interesting* and *interesting+important(categorical)*, respectively.

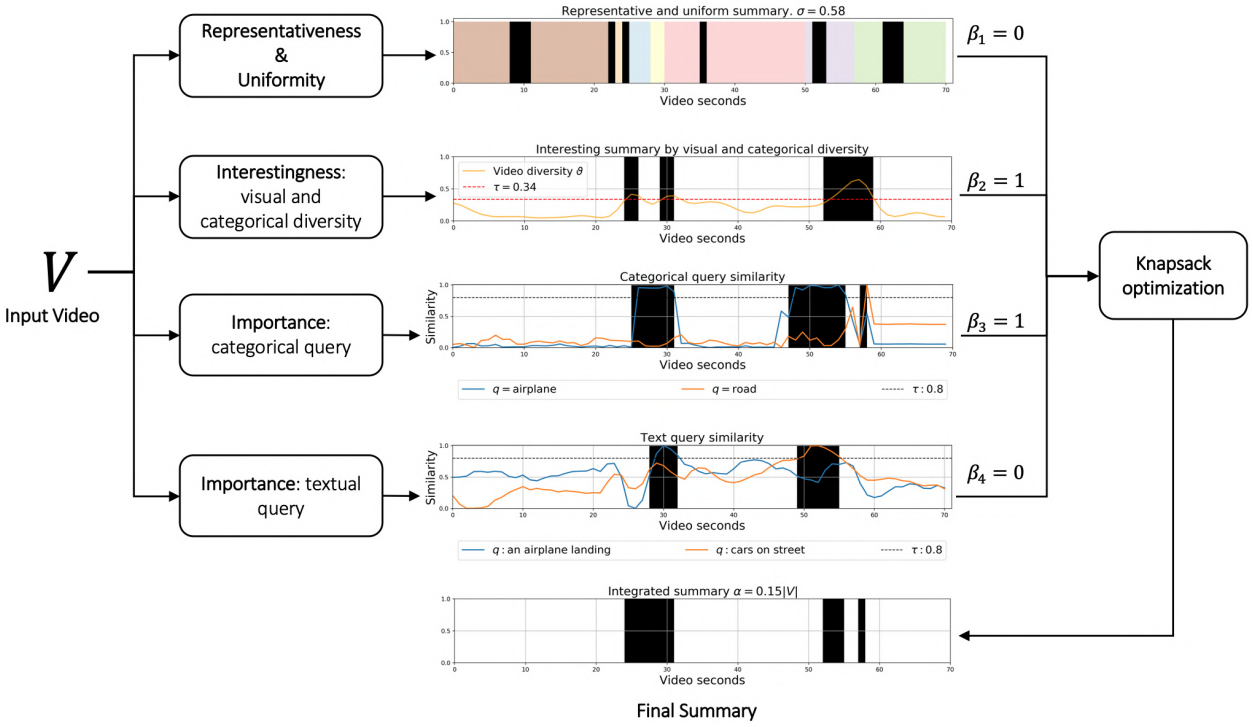

Figure 6-3.: Multi-objective VSUMM integration example, where only *Interestingness* (S_2) and *Importance: categorical query* (S_3) objectives are considered for the final summary.

6.3. Examples

In this section, we present some examples of input videos and summary results using our integration framework and selecting different combinations of VSUMM objectives.

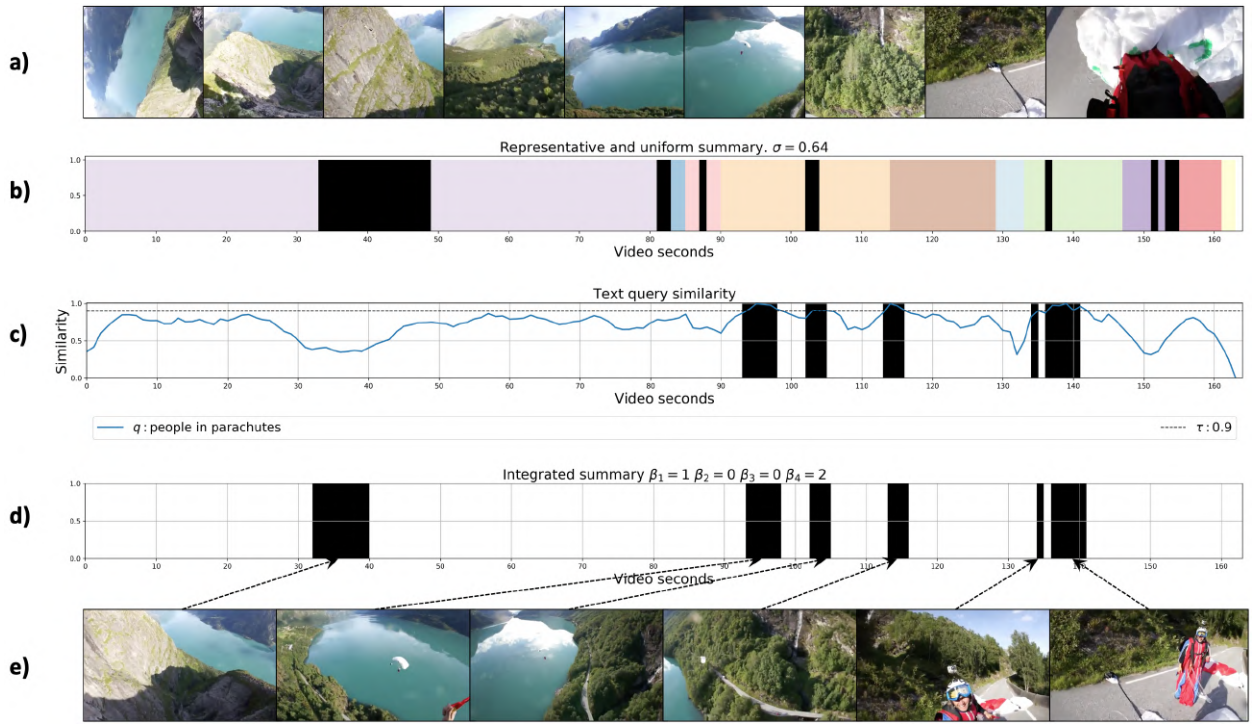

Figure 6-4.: Example of the multi-objective VSUMM integration scheme for video: *Base jumping* from dataset SumMe. **a)** Input video V (sampled to 1 frame each 20 seconds), **b)** representative and uniform summary, **c)** similarity with respect to the sentence query q =*[people in parachutes]*, **d)** integration by knapsack problem optimization, and **e)** central frames from each video segment from final summary.

In figure **6-4** it is shown an example of integration of multi-objective VSUMM. The use of the weight factors β allows to decide which VSUMM objective will be preserved in the final summary, but also which one will be more important than others. In this case notice that $\beta_4 = 2$ and $\beta_1 = 1$, that is, *importance* from text query similarity is considered twice as important as the *representativeness+uniformity*. From the previous, scenes related with *people in parachutes*, are more frequent in the final summary (see figure **6-4-(e)**), nonetheless, scenes from objective *representativeness+uniformity* are also included in the final summary.

In the example in figure **6-5**, *interestingness* and importance from categorical query similarity were optimized in the final summary. Notice that query similarity for q =*[fish, animal]* is almost the same for both categories in two scenes where *sea urchin*, *lion fish* and *fish* were found by InceptionV3 model. This, illustrates the semantic relations captured by the GloVe word-embedding between both words. Finally, the optimized summary preserve scenes from both VSUMM objectives, equally weighted in this case.

A final example can be observed in figure **6-6** where *representativeness* and *interestingness*

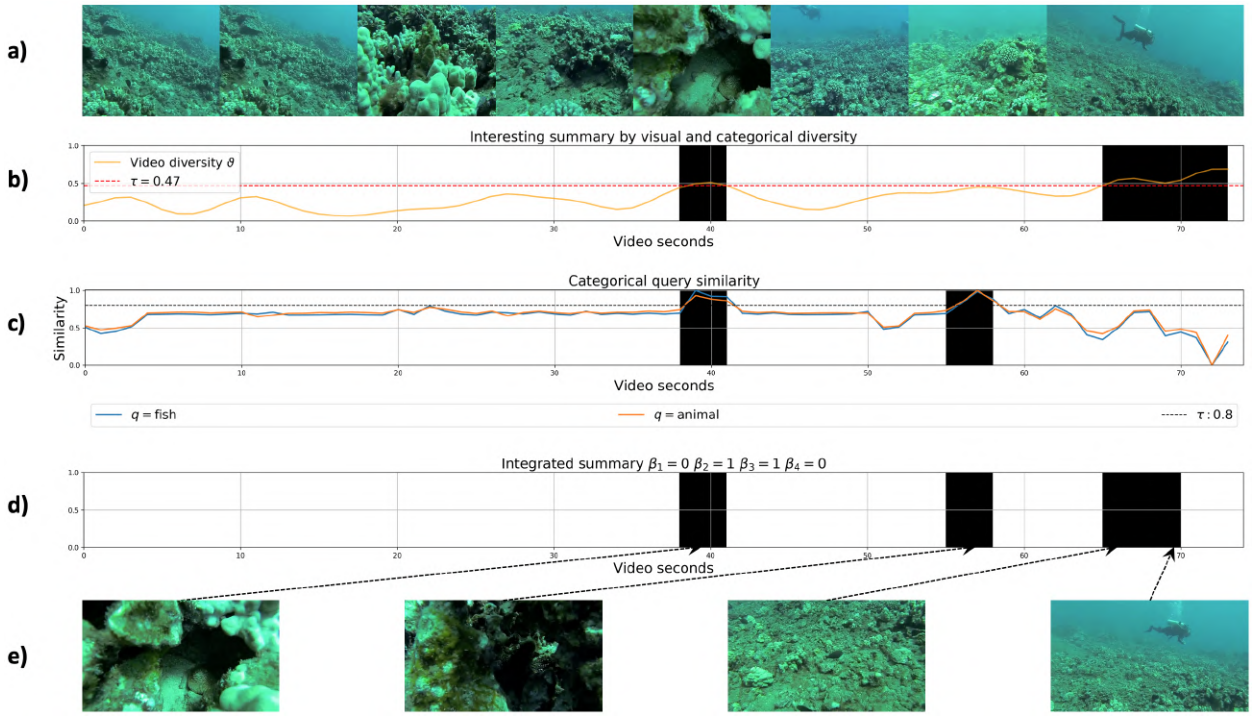

Figure 6-5.: Example of the multi-objective VSUMM integration scheme for video: Scuba from dataset SumMe. **a)** Input video V (sampled to 1 frame each 10 seconds), **b)** summary for the *interestingness* objective, **c)** similarity with respect to the categorical query q =*[fish, animal]*, **d)** integration by knapsack problem optimization, and **e)** central frames from each video segment from final summary.

VSUMM objectives were optimized.

6.4. Conclusions and Future Work

In this thesis, we have presented a query-based video summarization framework which uses information from visual and linguistic domains, particularly visual deep features from InceptionV3 model, linguistic features from GloVe and skip-thoughts models, and coordinated features from an ad-hoc coordination model which can be used as a numeric space where visual and linguistic semantics coexist.

The achievements of this thesis regarding the specific objectives, and future work is presented in the following:

- **(Objective 1) - To elaborate a coordinated representation space, from video data and its human-made textual annotations**: We have built and trained an ad-hoc deep model which coordinates visual and linguistic in chapter 3. To train this

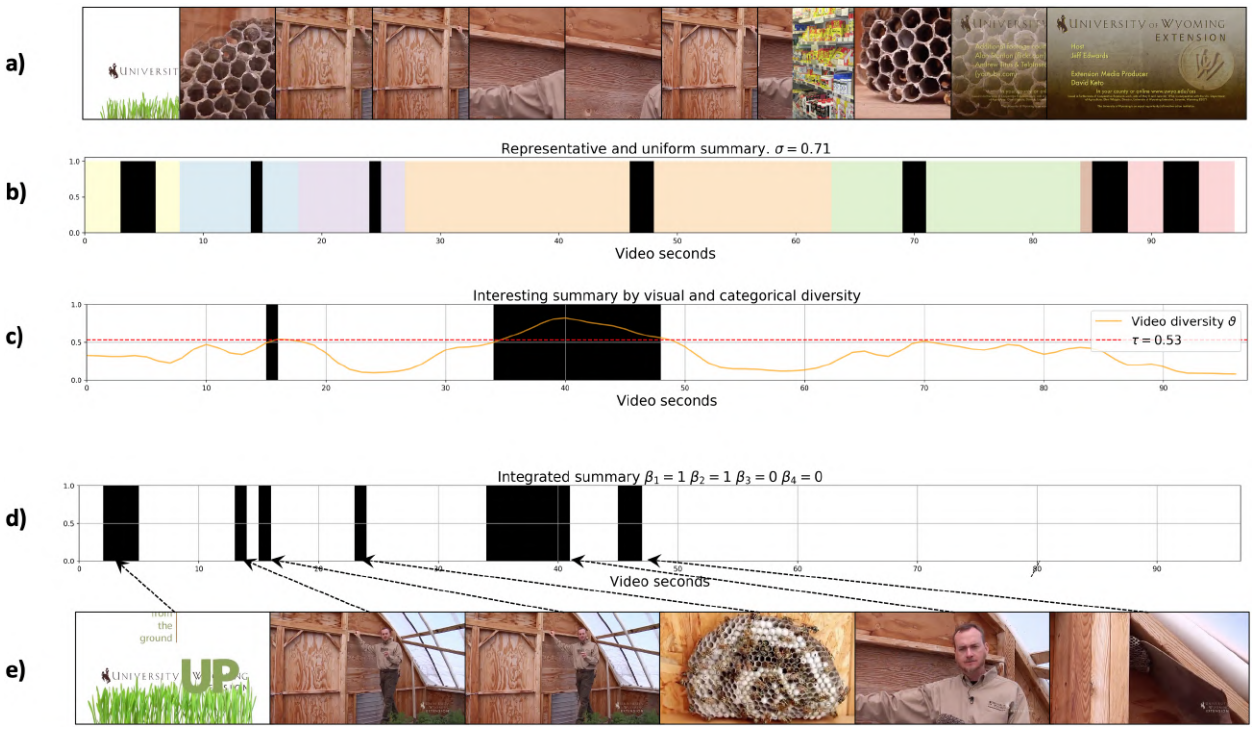

Figure 6-6.: Example of the multi-objective VSUMM integration scheme for video: *EE-bNr36nyA* from dataset TVSum50. **a)** Input video V (sampled to 1 frame each 10 seconds), **b)** representative and uniform summary, **c)** summary for *interestingness* objective, **d)** integration by knapsack problem optimization, and **e)** central frames from each video segment from final summary.

model, we used the TRECVID dataset, which contains short videos and sets of textual descriptions for each one. A series of evaluations were made to test the quality of the model, particularly, AUC analysis, retrieval metrics, and qualitative evaluations. Further exploration of recurrent layers can be made in future works. Considering that these layers require more data to obtain better performance, it is important to explore or construct massive VTT datasets that allow training these types of models.

- **(Objective 2) - To develop a computational method to obtain representative video segments, using the coordinated representation space**: We have developed a computational model to obtain representative and uniform summaries, which uses k-medoids and hierarchical clustering over coordinated features of an input video. Our method allows us to automatically select the k parameter from k-medoids using the number of segments from hierarchical clustering. Nonetheless, an α-based summary can also be generated. A keyframes-based quantitative evaluation with respect to state-of-the-art methods was performed, and also qualitative evaluations over OVP, SumMe, and TVSum datasets.

- **(Objective 3) - To develop a computational method to obtain relevant segments, which allows textual user-made queries**: Relevance was worked from two VSUMM objectives: *interestingness* and *importance*. In the case of *interestingness*, in chapter 5, we have proposed a visual and categorical diversity method that estimates a score of change in the video in terms of its visual content and its categorical content. This method was compared against state-of-the-art works over the SumMe dataset. In the case of *importance*, treated in this work as user-personalization, we developed two approaches: **(1)** Word query similarity (see chapter 5), for which GloVe word-embedding was used in order to obtain a numeric vector for an input query q and a similarity score is computed between the query and the categories found in each video frame, also mapped with GloVe, and **(2)** A free-form text similarity (see chapter 3), for which we used the coordinated model built-in chapter 3, to compare sequences of video frames against a text query q mapped to a numeric vector using Skip-thoughts unidirectional and bidirectional models. In both cases, we present visual comparison and retrieval examples to validate the power of each approach. Although interestingness developed in this work considers visual and categorical diversities, future research can be made by considering Region-based CNN and action recognition models to measure a degree of action and interaction diversities.
- **(Objective 4) - To propose a computational framework to obtain interesting and representative segments**: In this chapter (chapter 6), we have developed an optimization approach to integrate one or all VSUMM objectives, which allows weighting the relevance of each objective in the final summary. We consider future research to focus on the use of fuzzy logic in the video summary integration process. For example, if the β parameters associated with each objective of the summary are considered probabilities, it would be possible to construct more versatile combinations of objectives using fuzzy inference.

It is important to mention some advantages and limitations of the framework proposed in this thesis, which can be used later as future work for further research:

Advantages

- Our framework does not require feature engineering as it relies on pre-trained image classification models (InceptionV3) and pre-trained linguistic models (GloVe, Skip-thoughts), except the case of the ad-hoc coordination model. This allows for rapid deployment and adjustment for industrial applications.
- Our framework is mainly unsupervised, in the sense that it does not rely on a supervised training approach ($V \rightarrow S$), which makes its result independent of the nature of the input video.

- The final summary can be personalized by both words and text queries which allows us to express the user intentions in the VSUMM task.
- The final summary can be single-objective, multi-objective, or weighted using β parameters and a knapsack problem optimization approach.
- Our framework allows us to specify each VSUMM objective separately, by using its related parameters, such as hierarchical clustering sensitivity σ (see chapter 4) or similarity thresholds τ for query-based importance.

Limitations

- Rely on pre-trained image classification models has the disadvantage of a high computational cost due to the size of the model, which makes mandatory GPU-computing use in order to process input video in a reasonable time. As future work, it is important to explore the performance of our framework using image classification models trained specifically for a limited set of image categories for a given task, which we believe will: **1)** improve the quality of the summary for task-related videos, and **2)** decrease the computational cost of computing deep features.
- The ad-hoc coordination model's retrieval performance will depend on the quality of the VTT dataset employed to train it. In our case, the dataset TRECVID is mainly composed of short videos commonly related to funny situations, which penalizes the model's generalization power. As future work, we propose the use of general VTT datasets and task-related VTT datasets, which allows to improve the generalization performance of the coordination model, or improve the specificity of the model in a given task, respectively.

Finally, our intention from this result is to apply our framework in specific projects useful in local problems, which can be benefited from video analysis, such as surveillance and security video analysis, traffic video analysis, and sports video analysis.

A. Annex: Grid-search for coordination model architecture

In table **A-1** it is presented the grid search performed to optimize parameters from coordination model.

Table A-1.: Grid search for averaged and recurrent coordination models. Best area under the curve (auc) from averaged and recurrent architectures are shown in bold.

index	activation	units	alpha	drop_rate	auc_avg	auc_rnn
0	tanh	512	0.8	0.2	0.924942	0.904378
1	tanh	512	0.8	0.3	0.922835	0.900993
2	tanh	512	0.9	0.2	0.927860	0.908676
3	tanh	512	0.9	0.3	0.926024	0.905796
4	tanh	512	1.0	0.2	0.926421	0.909608
5	tanh	512	1.0	0.3	0.926172	0.908004
6	tanh	1024	0.8	0.2	0.917809	0.914399
7	tanh	1024	0.8	0.3	0.916686	0.912122
8	tanh	1024	0.9	0.2	0.923608	0.914632
9	tanh	1024	0.9	0.3	0.922577	0.911936
10	tanh	1024	1.0	0.2	0.924254	0.914348
11	tanh	1024	1.0	0.3	0.923791	0.912855
12	relu	512	0.8	0.2	0.928338	0.602629
13	relu	512	0.8	0.3	0.926376	0.596629
14	relu	512	0.9	0.2	0.926718	0.603571
15	relu	512	0.9	0.3	0.925429	0.596465
16	relu	512	1.0	0.2	0.924304	0.898307
17	relu	512	1.0	0.3	0.923535	0.895910
18	relu	1024	0.8	0.2	0.929189	0.652779
19	**relu**	**1024**	**0.8**	**0.3**	**0.930429**	0.647643
20	relu	1024	0.9	0.2	0.927276	0.909161
21	relu	1024	0.9	0.3	0.928020	0.833548
22	relu	1024	1.0	0.2	0.924323	0.912850
23	relu	1024	1.0	0.3	0.925683	0.910773
24	mish	512	0.8	0.2	0.911142	0.613876
25	mish	512	0.8	0.3	0.907558	0.607409
26	mish	512	0.9	0.2	0.914496	0.911838
27	mish	512	0.9	0.3	0.912623	0.607397
28	mish	512	1.0	0.2	0.914715	0.910061
29	mish	512	1.0	0.3	0.913756	0.908454
30	mish	1024	0.8	0.2	0.913968	0.605957
31	mish	1024	0.8	0.3	0.909277	0.597742
32	**mish**	**1024**	**0.9**	**0.2**	0.918503	**0.915619**
33	mish	1024	0.9	0.3	0.916725	0.597599
34	mish	1024	1.0	0.2	0.918153	0.913205
35	mish	1024	1.0	0.3	0.917491	0.912279

References

[1] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, and R. Klette, "{STFCN:} Spatio-Temporal {FCN} for Semantic Video Segmentation," *CoRR*, vol. abs/1608.0, 2016.

[2] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8695 LNCS, pp. 505–520, 2014.

[3] A. G. del Molino, C. Tan, J. H. Lim, and A. H. Tan, "Summarization of Egocentric Videos: A Comprehensive Survey," *IEEE Transactions on Human-Machine Systems*, vol. 47, pp. 65–76, 2 2017.

[4] A. Sharghi, B. Gong, and M. Shah, "Query-focused extractive video summarization," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9912 LNCS, pp. 3–19, 2016.

[5] H. Oosterhuis, S. Ravi, and M. Bendersky, "Semantic Video Trailers," *CoRR - ICML 2016 Workshop on Multi-View Representation Learning*, vol. abs/1609.0, 2016.

[6] S. Yeung, A. Fathi, and L. Fei-Fei, "VideoSET: Video Summary Evaluation through Text," *arXiv preprint arXiv:1406.5824*, 2014.

[7] 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*, pp. 1346–1353, 2012.

[8] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, D. Joy, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, G. Quénot, J. Magalhaes, D. Semedo, and S. Blasi, "TRECVID 2018: Benchmarking Video Activity Detection, Video Captioning and Matching, Video Storytelling Linking and Video Search," in *Proceedings of TRECVID 2018*, NIST, USA, 2018.

[9] J. Xu, T. Mei, T. Yao, and Y. Rui, "MSR-VTT: A Large Video Description Dataset for Bridging Video and Language," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5288–5296, 2016.

[10] M. Gygli, H. Grabner, and L. V. Gool, "Video summarization by learning submodular mixtures of objectives," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3090–3098, 2015.

[11] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Learning Joint Representations of Videos and Sentences with Web Image Search," *CoRR*, vol. abs/1608.0, 2016.

[12] K. Zhang, K. Grauman, and F. Sha, "Retrospective Encoders for Video Summarization," 2018.

[13] K. Zhou, Y. Qiao, and T. Xiang, "Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward," *AAAI 2018*, 2018.

[14] A. Sharghi, J. S. Laurel, and B. Gong, "Query-Focused Video Summarization: Dataset, Evaluation, and A Memory Network Based Approach," 7 2017.

[15] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video Summarization with Long Short-term Memory," *ECCV*, pp. 1–24, 2016.

[16] A. Oliva and A. Torralba, "The role of context in object recognition," *Trends in Cognitive Sciences*, vol. 11, no. 12, 2007.

[17] T. Brosch, K. R. Scherer, D. Grandjean, and D. Sander, "The impact of emotion on perception, attention, memory, and decision-making.," *Swiss medical weekly*, vol. 143, p. w13786, 2013.

[18] M. Greene, A. Botros, D. Beck, and F.-F. Li, "What you see is what you expect: rapid scene understanding benefits from prior experience.," *Attention, Perception, & Psychophysics*, vol. 77, no. 4, 2015.

[19] M. Bar and E. Aminoff, "Cortical Analysis of Visual Context," *Neuron*, vol. 38, pp. 347–358, 2003.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, vol. 1301.3781, 2013.

[21] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2121–2129, Curran Associates, Inc., 2013.

[22] Z. Lu and K. Grauman, "Story-driven summarization for egocentric video," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition, pp. 2714–2721, 2013.

[23] P. Wang, Y. Cao, C. Shen, L. Liu, and H. T. Shen, "Temporal Pyramid Pooling Based Convolutional Neural Networks for Action Recognition," *CoRR*, vol. abs/1503.0, 2015.

[24] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 4507–4515, 2015.

[25] Y. Xu, Y. Han, R. Hong, and Q. Tian, "Sequential Video VLAD: Training the Aggregation Locally and Temporally," *IEEE Transactions on Image Processing*, 2018.

[26] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel, "Visual Question Answering: A Survey of Methods and Datasets," p. 25, 7 2016.

[27] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014.

[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, (USA), pp. 3111–3119, Curran Associates Inc., 2013.

[29] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," *Advances in Neural Information Processing Systems*, vol. 2015-Janua, pp. 3294–3302, 2015.

[30] Z. Lu and K. Grauman, "Story-Driven Summarization for Egocentric Video," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2714–2721, 2013.

[31] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5179–5187, 2015.

[32] Y. Gong and X. Liu, "Video Summarization and Retrieval Using Singular Value Decomposition," *Multimedia Syst.*, vol. 9, no. 2, pp. 157–168, 2003.

[33] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse Sequential Subset Selection for Supervised Video Summarization," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, (Cambridge, MA, USA), pp. 2069–2077, MIT Press, 2014.

[34] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[35] 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, pp. 886–893, 2005.

[36] W. Wolf, "Key frame selection by motion analysis," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, pp. 1228–1231, 5 1996.

[37] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognition*, vol. 30, pp. 643–658, 4 1997.

[38] A. Kapoor, K. K. Biswas, and M. Hanmandlu, "Fuzzy video summarization using key frame extraction," in *2013 4th National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2013*, pp. 1–5, 2013.

[39] N. Zlatintsi, P. Maragos, A. Potamianos, and G. Evangelopoulos, "A saliency-based approach to audio event detection and summarization," 2012.

[40] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 787–802, Springer Verlag, part 1 ed., 2014.

[41] T. Joachims, "Optimizing Search Engines Using Clickthrough Data," in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, (New York, NY, USA), pp. 133–142, ACM, 2002.

[42] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li, "A User Attention Model for Video Summarization," in *Proceedings of the Tenth ACM International Conference on Multimedia*, MULTIMEDIA '02, (New York, NY, USA), pp. 533–542, ACM, 2002.

[43] G. Kim, L. Sigal, and E. P. Xing, "Joint summarization of large-scale collections of web images and videos for storyline reconstruction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4225–4232, 2014.

[44] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, (Miami Beach, FL.), 2009.

[45] A. Khosla, R. Hamid, C. J. Lin, and N. Sundaresan, "Large-Scale Video Summarization Using Web-Image Priors," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2698–2705, 2013.

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

[47] B. A. Plummer, M. Brown, and S. Lazebnik, "Enhancing video summarization via vision-language embedding," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1052–1060, 2017.

[48] K. Aizawa, K. Ishijima, and M. Shiina, "Summarizing wearable video," in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 3, pp. 398–401, 2001.

[49] P. Varini, G. Serra, and R. Cucchiara, "Personalized Egocentric Video Summarization for Cultural Experience," *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval - ICMR '15*, vol. PP, no. 99, pp. 539–542, 2015.

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

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

[52] D. Borth, T. Chen, R. Ji, and S.-F. Chang, "SentiBank: Large-scale Ontology and Classifiers for Detecting Sentiment and Emotions in Visual Content," in *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, (New York, NY, USA), pp. 459–460, ACM, 2013.

[53] P. Varini, G. Serra, and R. Cucchiara, "Personalized Egocentric Video Summarization of Cultural Tour on User Preferences Input," *IEEE Transactions on Multimedia*, vol. PP, no. 99, p. 1, 2017.

[54] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual semantic search: Retrieving videos via complex textual queries," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2657–2664, 2014.

[55] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *CVPR*, 2014.

[56] T. Sebastian and J. J. Puthiyidam, "Article: A Survey on Video Summarization Techniques," *International Journal of Computer Applications*, vol. 132, no. 13, pp. 30–32, 2015.

[57] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Category-specific video summarization," in *ECCV 2014 - European Conference on Computer Vision* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), vol. 8694 of *Lecture Notes in Computer Science*, (Zurich, Switzerland), pp. 540–555, Springer, 2014.

[58] S. E. F. de Avila, A. P. B. Lopes, A. da Luz Jr., and A. de Albuquerque AraË™jo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[59] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya, "Video Summarization using Deep Semantic Features," *CoRR*, vol. abs/1609.0, 2016.

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

[61] V. Chasanis, A. Kalogeratos, and A. Likas, "Movie segmentation into scenes and chapters using locally weighted bag of visual words," in *CIVR 2009 - Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 264–271, 2009.

[62] T.-y. Lin, M. Maire, S. Belongie, and L. Bourdev, "Microsoft: COCO: Common Objects in Context," *Computer Vision*, 2015.

[63] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," tech. rep., 9 2014.

[64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[65] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[66] R. Girshick, "[Fast R-CNN] Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 1440–1448, 2015.

[67] H. Fang, S. Gupta, F. Landola, and R. Srivastava, "From Captions to Visual Concepts and Back," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Computer Society Conference on*, 2015.

[68] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention." 2015.

[69] K. Zhang, W. L. Chao, F. Sha, and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," *Proceedings of the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, pp. 1059–1067, 2016.

[70] T. Baltrusaitis, C. Ahuja, and L. P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," 2018.

[71] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (K. Daniilidis, P. Maragos, and N. Paragios, eds.), vol. 6314 LNCS, pp. 15–29, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[72] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. D. Manning, and A. Y. Ng, "Zero-Shot Learning Through Cross-Modal Transfer," *CoRR*, vol. abs/1301.3, 2013.

[73] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 19–27, 2015.

[74] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *CoRR*, vol. abs/1409.4, 2014.

[75] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly Modeling Deep Video and Compositional Text to Bridge Vision and Language in a Unified Framework," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2346–2352, AAAI Press, 2015.

[76] R. Xu, C. Xiong, W. Chen, and J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework," in *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 2346–2352, 2015.

[77] P. Atencio, S. T. German, J. W. Branch, and C. Delrieux, "Video summarisation by deep visual and categorical diversity," *IET Computer Vision*, vol. 13, no. 6, pp. 569–577, 2019.

[78] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *CoRR*, vol. abs/1512.0, 2015.

[79] F. Chollet, "Xception: Deep Learning with Separable Convolutions," *arXiv preprint arXiv:1610.02357*, pp. 1–14, 2016.

[80] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.0, 2015.

[81] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *CoRR*, vol. abs/1602.0, 2016.

[82] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *CoRR*, vol. abs/1608.0, 2016.

[83] D. Lahat, T. Adali, and C. Jutten, "Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects," 2015.

[84] A. Karpathy, *Connecting Images and Natural Language*. PhD thesis, Stanford University, 2016.

[85] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[86] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality Reduction by Learning an Invariant Mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, pp. 1735–1742, 8 2006.

[87] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations," *Transactions of the Association of Computational Linguistics – Volume 2, Issue 1*, 2014.

[88] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models," *CoRR*, vol. abs/1411.2, 2014.

[89] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image Captioning with Semantic Attention," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*, vol. preprint, 2016.

[90] A. B. Vasudevan, M. Gygli, A. Volokitin, and L. Van Gool, "Query-adaptive Video Summarization via Quality-aware Relevance Estimation," in *Proceedings of the 25th ACM international conference on Multimedia*, (Mountain View, California, USA), pp. 582–590, ACM, 2017.

[91] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain Images with Multimodal Recurrent Neural Networks," *CoRR*, vol. abs/1410.1, 2014.

[92] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded Compositional Semantics for Finding and Describing Images with Sentences," *Transactions of the Association for Computational Linguistics*, 2014.

[93] S. E. F. De Avila, A. P. B. Lopes, A. Da Luz, and A. De Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel

evaluation method," in *Pattern Recognition Letters*, 2011.

[94] J. Iparraguirre and C. Delrieux, "Online Video Summarization Based on Local Features," *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 5, pp. 41–53, 2014.

[95] S. E. F. De Avila, A. P. B. Lopes, A. Da Luz, and A. De Albuquerque Araújo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," in *Pattern Recognition Letters*, 2011.

[96] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," *International Journal on Digital Libraries*, 2006.

[97] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: STIll and MOving video storyboard for the web scenario," *Multimedia Tools and Applications*, 2010.

[98] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *IEEE International Conference on Image Processing*, 1998.

[99] H. C. Shih, "A Survey on Content-aware Video Analysis for Sports," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, p. 1, 2017.

[100] J. Sang and C. Xu, "Character-based Movie Summarization," in *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, (New York, NY, USA), pp. 855–858, ACM, 2010.

[101] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[102] M. Sun, A. Farhadi, B. Taskar, and S. Seitz, "Summarizing Unconstrained Videos Using Salient Montages," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2256–2269, 2017.

[103] N. Ejaz, I. Mehmood, and S. W. Baik, "Efficient visual attention based framework for extracting key frames from videos," *Signal Processing: Image Communication*, vol. 28, no. 1, pp. 34–44, 2013.