applied sciences

MDPI

Article

Multimodal Summarization of User-Generated Videos

Theodoros Psallidas [1,2], Panagiotis Koromilas [1], Theodoros Giannakopoulos [1,*] and Evaggelos Spyrou [1,2,*]

[1] Institute of Informatics and Telecommunications, National Center for Scientific Research—"Demokritos", 15310 Athens, Greece; theopsall@iit.demokritos.gr (T.P.); pakoromilas@iit.demokritos.gr (P.K.)
[2] Department of Computer Science and Telecommunications, University of Thessaly, 35100 Lamia, Greece
* Correspondence: tyianak@iit.demokritos.gr (T.G.); espyrou@uth.gr (E.S.); Tel.: +30-210-650-3175 (T.G. & E.S.)

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

Keywords: video summarization; audiovisual features; benchmark dataset; machine learning

Citation: Psallidas, T.; Koromilas, P.; Giannakopoulos, T.; Spyrou, E. Multimodal Summarization of User-Generated Videos. *Appl. Sci.* **2021**, *11*, 5260. https://doi.org/10.3390/app11115260

Academic Editor: Andrea Prati

Received: 13 April 2021
Accepted: 31 May 2021
Published: 5 June 2021

1. Introduction

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

As the amount of available information grows, the amount of assistance that users need to efficiently browse huge video collections [2] and to derive useful information within large videos is rising. To fulfil the aforementioned ever-growing needs, many research efforts have shifted towards the task of *video summarization*. In brief, this task aims to create a condensed version of a given video sequence [3]; when a user watches this version, she/he should immediately capture the most important parts of the video's content. Apart from efficient browsing and retrieval of videos for entertainment purposes [4], applications of video summarization include summarization of surveillance videos [5], of medical videos [6], of large videos captured by unmanned aired vehicles (UAVs) [7] and so forth.

Approaches to video summarization may be classified into four major categories, based on the type of audiovisual visual cues produced and presented to their end user [3], that is, their output. More specifically, output of video summarization algorithms may consist of a) keyframes [8], which are extracted video frames, presented in order and are

often denoted as "static" summaries; b) (a set of) video segments [9], which are often denoted as "dynamic" summaries and consist of an obvious extension of keyframes and preserve both audio and motion of videos; c) graphical cues [10] which complement other cues by some type of graphical-based syntax to further enhance the interpretation of summaries by the end user; and d) automatically generated textual annotations [11], which aim to provide efficient summaries of video content.

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

More specifically, in this work, we propose the use of supervised knowledge from both audio and visual domains, to achieve summarizations of user-generated videos. In particular, we analyze a given video stream by splitting it into one-second segments of audio and visual representations. Segments are either classified as being "informative" (i.e., adequately "interesting" so that they should be used within the final video summary) or "uninteresting" (i.e., not containing information that should be included within the final video summary). We use a supervised binary classifier trained on the feature representations of either audio, video or fused modalities. Moreover, we present a novel dataset, comprised of user-generated videos, collected from YouTube, that have been recorded using either action cameras or smartphones. Contrary to other open datasets used for video summarization, our dataset is (a) well-defined; (b) user-generated; and (c) adequately large to train and evaluate the proposed methodology. In this paper, we also present in detail the way the collective annotation process has been carried out by a group of human annotators using an annotating tool, which was created for the purpose of this work.

The rest of this paper is organized as follows: in Section 2 we present and discuss related, state-of-the-art research in video summarization. In Section 4, we formulate the problem at hand and discuss the approach we followed to collect and process the dataset we have used for the experimental evaluation. Then, in Section 3 we describe in detail the proposed multi-modal video summarization methodology. Experimental results are presented and discussed in Section 5, while conclusions are drawn in Section 6.

2. Related Work

2.1. Approaches for Video Summarization

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

In [14], the authors formulate a video summarization as a sequential decision-making process while they develop a deep summarization network, trained with an end-to-end reinforcement learning-based framework that is able to predict for each video a probability that indicates whether the particular frame will be part of the video summary. The above model architecture consists of an encoder-decoder where the encoder is a convolutional neural network (CNN), responsible for frame feature extraction and the decoder is a long-short term memory network (LSTM), responsible for the frame probabilities. A novel supervised technique was proposed in [15] for summarizing videos based on an LSTM architecture. This approach automatically selects keyframes or keyshots, deriving compact and meaningful video summaries. In addition they report that techniques such as domain adaptation may improve the entire process of video summarizing. A generic video summarization algorithm was proposed in [16] by fusing the features from different multimodal streams. A low-level feature fusion approach using as input visual, auditory and textual

streams has been used, so as to develop a well-formed representation of the input video in order to construct a video summarization based on the informative parts from all streams.

In [17] it is pointed out that the main goal of a video summarization methodology is to make a more compact version of the initial raw video, without losing much semantic information and making quite comprehensive for the viewer. They present an innovative solution namely SASUM, which in contrast to the techniques so far that take only the diversity of the summary, extracts the most descriptive parts of the video summarizing the video. Specifically, SASUM consists of a frame selector as well as the video descriptors to compose the final video that will minimize the distance with the generated description from the description that has already been created by humans. A memory and computational efficient technique based on a hierarchical graph-based algorithm, which is able to make spatio-temporal segmentation on long video sequences, was presented in [18]. The spatio-temporal algorithm repeatedly makes segments into space–time regions clustered by their frequencies, constructing a tree consisting of such spatio-temporal segments. Moreover, the algorithm is boosted by introducing dense optical flow to describe the temporal connections on the aforementioned graph. In [19], it is emphasized that the huge number of videos that are produced on a daily basis need a summary technique to present a condensed format of the video without the unnecessary information. More specifically, their approach, namely SalSum makes use of a generative adversarial network (GAN) which has been pre-trained using the human eye fixations. The model combines colors as well as visual elements in an unsupervised model. The protrusions, along with the color information deriving from the visual flow of video through SalSum, compose a video summary.

The work proposed in [20] focuses on the computational model development based on the visual attention in order to summarize videos, mostly from television archives. Their computational model is using several techniques in order to ensemble a static the video summary, such as face detection, motion estimation and saliency map computation. The final video summary from the above computational model consists of a collection of key frames or saliency images extracted from the raw video. A novel video summarization approach, namely VISCOM was proposed in [21] and was based on the color occurrence matrices from the video, used to describe each video frame. Then, a synopsis of the most informative frames of the original video was composed. VISCOM was tested on a large amount of videos from a variety of categories, in order to make the aforementioned video summarization model robust. In [22] authors focused on the importance of the video summary on tasks such as video search, retrieval and so forth.. In relation to the approaches based on recurrent neural networks, they tested a fully convolutional sequence neural network on semantic segmentation as the solution of the sequence labeling problem for the video summarization task.

A deep video feature extraction process was proposed in [23], aiming to find the most informative parts of the video which are required so as to analyze video content. They included various levels of content to train their deep feature extraction technique. Their deep neural network also combined the description of the video, in order to extract the video features and then constructed the video summary by applying clustering based techniques also mentioned by the authors in [24]. The evaluation followed on their work is based on their own video summaries constructed by humans. The main goal of [24] was to remove redundant frames of an input video by clustering informative frames, which appeared to be the most effective way to construct a static video summary, built from all cluster centers. The frame representation that has been used within the clustering process was based on the Bag-of-Visual Words model. KVS is a novel video summarization approach, proposed in [25], specified from the video category provided, mainly from the title or the description of the video. A temporal segmentation is initially applied on a given video; its result is used as input on the KVS supervised algorithm, in order to build a higher quality video summaries compared to the unsupervised blind video category approaches.

Ma et al. [26] proposed an approach for keyframe extraction and video skimming that was based on a user attention model. To build a motion model, they extracted video,

audio, and linguistic features and built an attention model based on the motion vector field. They created three types of maps based on intensity, spatial and temporal coherence which were then fused to form a saliency map. They also incorporated a static model to select salient background regions and extracted faces as well as camera attention features and finally, the created audio, speech and music models. The aforementioned attention components were linearly fused to create an "attention" curve. Local maxima of this curve within shots were used for keyframe extraction, while skim segments were selected using several criteria. Mahaseni et al. [27] trained a deep adversarial LSTM network consisting of a "summarizer" and a "discriminator" so as to minimize distance between ground truth videos and their summarizations, based on deep features extracted by a CNN. More specifically, the former consists of a selector and an encoder that selects interesting frames from the input video and encode them to a deep feature. The latter is a decoder that classifies a given frame as "original" or "summary". The deep neural network proposed here tries to fool the discriminator by providing the video summary as the original input video, assuming that both representations are the same.

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

2.2. Related Data Sets

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

The "MED Summaries" [25] is a new dataset for evaluation of dynamic video summaries, containing annotations of 160 videos in total, with ten event categories in the test set. Indicative categories are "birthday party", "changing a vehicle tire", "flash mob gathering", "getting a vehicle unstuck", "grooming an animal", and so forth. The "TVSum" (Title-based Video Summarization) dataset [28] aims to solve the challenging task of prior knowledge in the main topic of the video. The entire dataset consists of 50 videos of various genres (e.g., "news", "how-to", "documentary", "vlog", "egocentric") and 1000 annotations of shot-level importance scores obtained via crowd-sourcing (20 per video), while video duration ranges between 2 and 10 min. The video and annotation data permit an automatic evaluation of various video summarization techniques, without having to conduct an (expensive) user study. The "SumMe" [29] is a video summarization dataset consisting of 25 videos, covering holidays, events and sports, downloaded from the popular platform of YouTube, each annotated with at least 15 human-created summaries (390 in total), while the length of the videos ranges from 1 to 6 min. The "UT Ego" (Univ. of Texas at Austin Egocentric) Dataset [30] contains 10 (4 out of 10 are available due to privacy reasons) videos captured from head-mounted cameras on a variety of activities such as "eating", "shopping", "attending a lecture", "driving", and "cooking". Each video is about 3–5 h long, captured at 15 fps and at 320 $\times$ 480 resolution uncontrolled setting. Therefore, videos contain shots with fast motion. Finally, the VSUMM dataset [31] has been initially used to produce a static video summary, by a novel evaluation method, able to remove the subjectivity of the summary quality by allowing objective comparisons of methodology between different approaches. This dataset, also known as "YouTube Dataset" consists of 50 videos from the Open Video Project (http://www.open-video.org/). The duration of the videos varies from 1 to 4 min while the approximately duration of the videos in total is approx. 75 min. The videos originate from a variety of genres such as documentary, educational, ephemeral, historical and lecture. There exist 250 user summaries created

manually by 50 individuals, each one annotating five videos, that is, each video has five video summaries created by five different users.

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

3. Multimodal Video Summarization

3.1. Problem Formulation

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

3.2. Feature Extraction

Two types of information have been utilized for summarizing the videos: the auditory and the visual modality. To achieve feature representation in both modalities, we extracted hand-crafted features that are frequently used in audio and visual classification and clustering tasks such as music information retrieval, auditory scene analysis, video classification and image retrieval. Our goal was to include as many informative audio and visual features as possible. Figure 1 shows the conceptual diagram of the process followed to extract features for both the audio and visual modalities. More details on feature extraction are presented in the following.

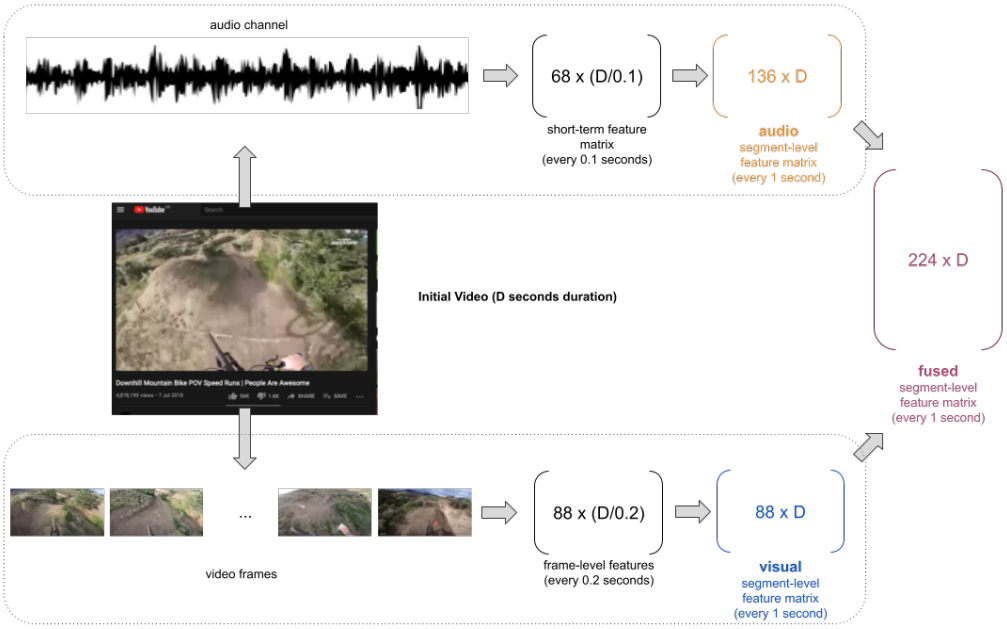

Figure 1. Conceptual diagram of the feature extraction process for both audio and visual modalities.

3.2.1. Audio

Low-level hand-crafted features have been shown that they may capture both perceived information, as well as the harmonic information of sound signals and have been

widely used in several application domains. Regarding the low-level description of the whole event, it has been shown that feature vectors constructed using statistics, such as mean and standard deviation of features can be efficiently used in event recognition-related tasks [32].

Therefore, for each audio clip, extracted from the respective video file using `ffmpeg`, we calculate segment-level audio features, using the `pyAudioAnalysis` library (https://github.com/tyiannak/pyaudioanalysis) [33]. According to this procedure, audio feature extraction is firstly carried out at a short-term basis. At a second level, segment-level feature statistics are computed and compose the final segment representation. In particular, the audio signal is divided into segment-level windows (either overlapping or non-overlapping) and for each segment a short-term processing is taken place, according to which 68 short-term features are computed (34 features and 34 deltas) for each short-term window. Short-term windows usually vary from 10 to 200 ms, while segment windows can be from 0.5 s to several seconds, depending on what is considered a homogeneous segment in the individual application domain. The short-term features extracted by the particular library are of three categories: time-domain, frequency domain and cepstral domain. The adopted short-term features are implemented in the pyAudioAnalysis library [33] and are shown in Table 1.

Table 1. Adopted short-term audio features.

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

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

3.2.2. Video

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

- Color—related features (45 features):
  - 8-bin histogram of the red values
  - 8-bin histogram of the green values
  - 8-bin histogram of the blue values
  - 8-bin histogram of the grayscale values
  - 5-bin histogram of the max-by-mean-ratio for each RGB triplet
  - 8-bin histogram of the saturation values
- Average absolute difference between two successive frames in grey scale (1 feature)
- Facial features (2 features): The Viola-Jones [34] OpenCV implementation is used to detect frontal faces and the following features are extracted per frame:
  - number of faces detected
  - average ratio of the faces' bounding boxes areas divided by the total area of the frame
- Optical-flow related features (3 features): The optical flow is estimated using the Lucas-Kanade method [35] and the following 3 features are extracted:
  - average magnitude of the flow vectors
  - standard deviation of the angles of the flow vectors
  - a hand-crafted feature that measures the possibility that there is a camera tilt movement—this is achieved by measuring a ratio of the magnitude of the flow vectors by the deviation of the angles of the flow vectors.
- Current shot duration (1 feature): a basic shot detection method is implemented in this library. The length of the shot (in seconds) in which each frame belongs to, is used as a feature.
- Object-related features (36 features): We use the Single Shot Multibox Detector [36] method for detecting 12 categories of objects. For each frame, as soon as the object(s) of each category are detected, three statistics are extracted: number of objects detected, average detection confidence and average ratio of the objects' area to the area of the frame. So in total, $3 \times 12 = 36$ object-related features are extracted. The 12 object categories we detect are the following: person, vehicle, outdoor, animal, accessory, sports, kitchen, food, furniture, electronic, appliance and indoor.

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

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

3.3. Segment-Level Classification

According to the process presented above, the content of each video has been described by an audio and a visual feature vector that represents each 1-s segment of the video. In addition, as described in Section 4, each 1-s segment of the video has been characterized either as "informative" or as "uninteresting". Informative segments are the ones that belong to the summary according to the aggregation process of the annotation data described in Section 4, while "uninteresting" are all other 1-s segments that the annotators have agreed that do not belong in the video summary. Based on this separation, a binary classification task can be formulated.

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

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

For the training procedure, a training set, which consisted of a proportion of 80% of the dataset's videos, has been created. The rest of the data has been used for validation purposes. The following classifier types have been evaluated for all three feature modality setups: Naive Bayes, k-Nearest Neighbors, Logistic Regression, Decision Tree, Random Forest, XGBoost and a Fully Connected Neural Network (FNN). For the first four, the implementation of [37] with the appropriate parameter tuning was used. For the k-Nearest Neighbors classifier the k parameter, which represents the number of neighbors, was optimized. As for the Logistic Regression, the inverse regularization parameter (i.e., C), was tuned. Decision Tree classifier was optimized with respect to the split quality measure criterion (i.e., Gini impurity, entropy etc.) and the the maximum tree depth. The Random Forest was based on the balanced classifier provided by [38], while for the XGBoost the classifier of [39] was used. Both of these classifiers were optimized in regard to the split quality measure criterion and the number of tree estimators. Finally, the Fully Connected Neural Network consists of eight layers (one input layer, six hidden layers and one output layer); the "ReLU" activation function has been used for all layers. Each layer is followed by a batch-normalization layer. Moreover the first five layers are also followed by a dropout layer. We should herein note that we have also experimented with Perceptron classifiers and Support Vector Machines (SVMs). The former showed poor performance, that is, not significantly better than random. Moreover, the latter, although they consist a very powerful and robust classifier, are not of practical use in terms of memory and training time, when the training dataset becomes significantly large [40,41].

3.4. Post-Processing

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

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

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

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

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

4. Dataset Compilation

4.1. Video Data

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

4.2. Annotation Procedure

At the end of the video collection process, the annotation process on the selected videos took place. The purpose of the video annotation process was to create the video summaries as ground truth for training and testing the proposed video summarization technique. More specifically, 22 humans were asked to watch and annotate some videos in order to construct the ground truth video summaries. This process was executed through a web application, specifically designed for this particular annotation pipeline. The application was capable of randomly serving all the videos, one by one, to the end user, while the user was able to watch the whole video, go back and forth in time, and note the time intervals she/he found interesting (informative). The user was able to freely label the timestamps of each informative time interval while the number of interesting intervals was arbitrary without restricting the user, making the whole process completely subjective. The users only had to provide the endpoints (starting and ending timestamps) for each informative time interval. This web application tool (Figures 2 and 3), which includes a quick user registration process, is called Video Annotator Tool (VAT) and is publicly available (https://github.com/theopsall/video_annotator).

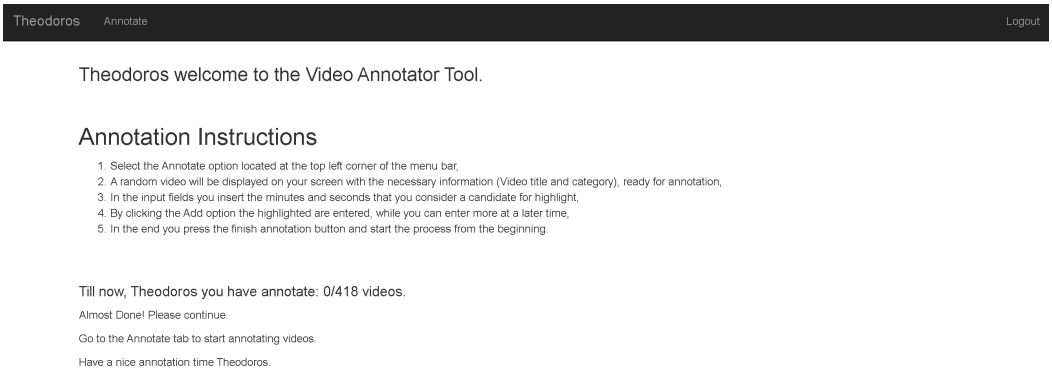

Figure 2. Video Annotator Tool Home page.

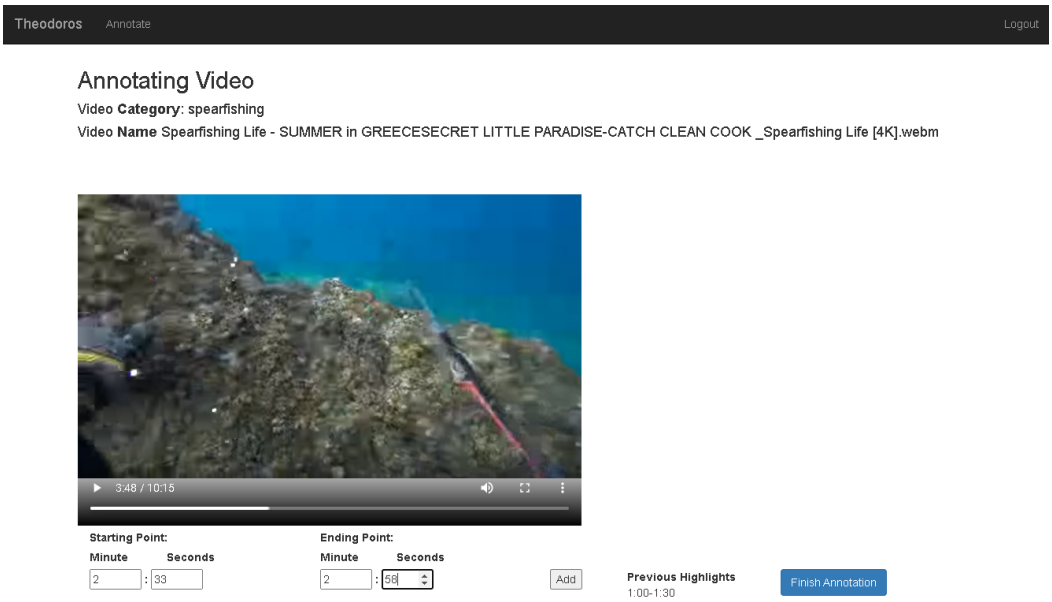

Figure 3. The annotating page of the web application.

A dataset of 1430 videos was voluntarily annotated by 22 annotators. In most of the cases, the videos were annotated by 3 to 4 annotators, while the maximum number of annotators for a given video was eight. The complete distribution of the number of human annotations per video are presented in Figure 4.

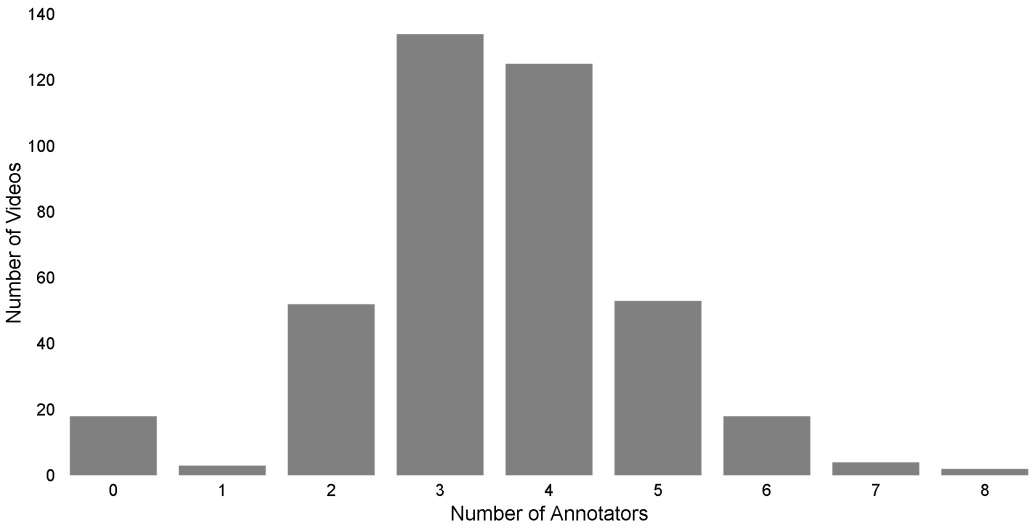

Figure 4. Distribution of the number of users annotated each video. Only 73 videos have been annotated by only 2 or less humans.

4.3. Annotation Data Aggregation

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

The resulting video summaries that were annotated, as mentioned in the previous section, did not necessarily have the same number of annotators per video, making the construction of a robust dataset more difficult. For that reason, the original dataset has been reduced by deleting the videos that have been annotated by less than three annotators.

During that process, 61 videos have been excluded from the dataset as not having sufficient annotation data. For 12 videos the aggregated annotation resulted in no informative segments at all, therefore these videos were also removed from the dataset. This procedure led to the final dataset comprising of 336 videos that have been annotated by at least three different annotators. For these videos that consist the final dataset, the respective ground truth was generated by a simple majority-based aggregation rule. In particular, each segment of 1-s was considered to be included in the summary (i.e., characterized as "informative") if at least by 60% of the annotators agreed on that decision.

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

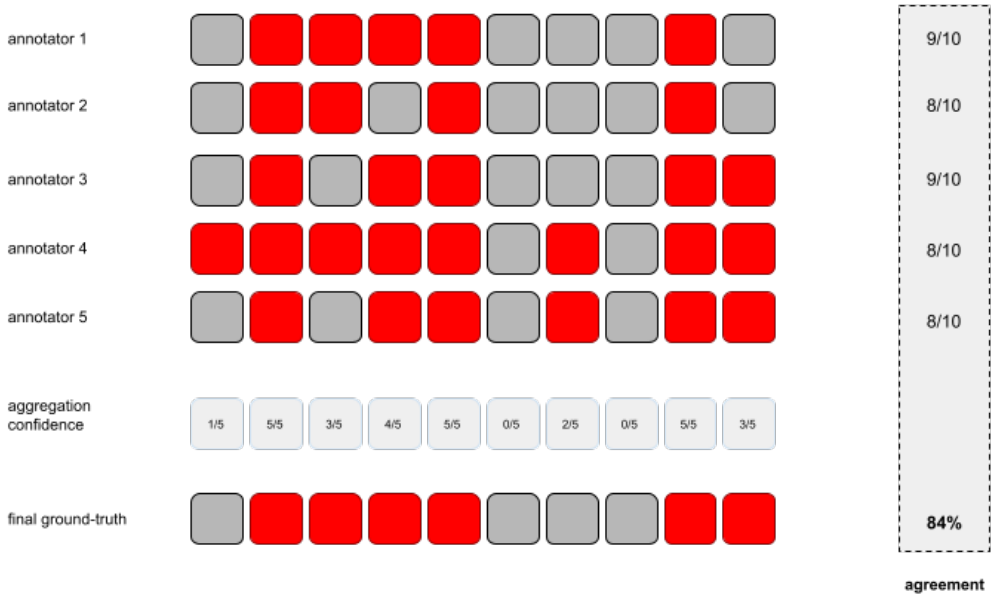

Figure 5. Example of aggregating annotation decisions from five different human annotators.

The final dataset (both raw data and respective annotations) (https://drive.google.com/drive/folders/1-nBp2zJKXsUe2xa9DtxonNdZ6frwWkMp?usp=sharing), along with the respective tools for the entire aggregation process are publicly available (https://github.com/theopsall/Video-Summarization). Some statistics for our dataset are presented in Table 2.

Table 2. Video dataset statistics.

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

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

5. Results

5.1. Evaluation Metrics

Before proceeding to the definition of the adopted evaluation metrics for the proposed classification task, let us focus on the way the data is separated to training and testing. Train/test split strategies are significantly contributing to the statistical correctness of the results of any supervised methodology. In this work, we have chosen to split the data not at segment level, but at the video level of the dataset. In that way, different 1-s segments (i.e., individual examples of the classification task) of the same video cannot belong at the same time to both training and test sets, since that would introduce significant bias in the results, as the classifiers would be "video-dependent". Under that constraint, 20% of the data have been used for test as presented in Table 3.

Table 3. Training-test samples.

Subset	Total Videos	Total Samples
Training Dataset	268	127,972
Test Dataset	68	31,113

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

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

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

5.2. Results

Table 4 shows AUC and F1 for six different classification methods and for the three modalities (audio, visual and fusion). Random Forest seems to be the best choice as it achieves the best AUC score and one of the best F1 scores. However, AUC is more important in our case as it incorporates the ability of the classifier to function at different operation points, that is, different probabilistic thresholds. It is also important to note that the classifiers based on the visual modality are always at least 4% relatively better than the audio-based classifiers, while the relative improvement is almost 3% better in fusion

compared to the visual modality. ROC curves for top three classifiers are illustrated in Figure 6.

Table 4. Segment Level Metrics. Best performance per modality/metric is indicated in bold.

Classifier	ROC AUC			F1 macro averaged		
	Audio	**Visual**	**Fused**	**Audio**	**Visual**	**Fused**
Random	49.7%			47.6%		
Naive Bayes	59.5%	64%	63.4%	51.7%	48.3%	51.6%
KNN	59.3%	60.7%	62.6%	54.6%	56.3%	57.7%
Log Reg	62.8%	67.2%	67.4%	41.4%	44.6%	49.4%
Decision Tree	60.6%	66.3%	66.5%	41.8%	45.6%	45.6%
Random Forest	66.7%	**69.8%**	**71.8%**	57.8%	60.4%	60.6%
XGBOOST	65.3%	66.8%	69.6%	59.8%	60.4%	62.3%
FNN	**67.45%**	68.6%	70.14%	**62.12%**	**64.4%**	**66.37%**

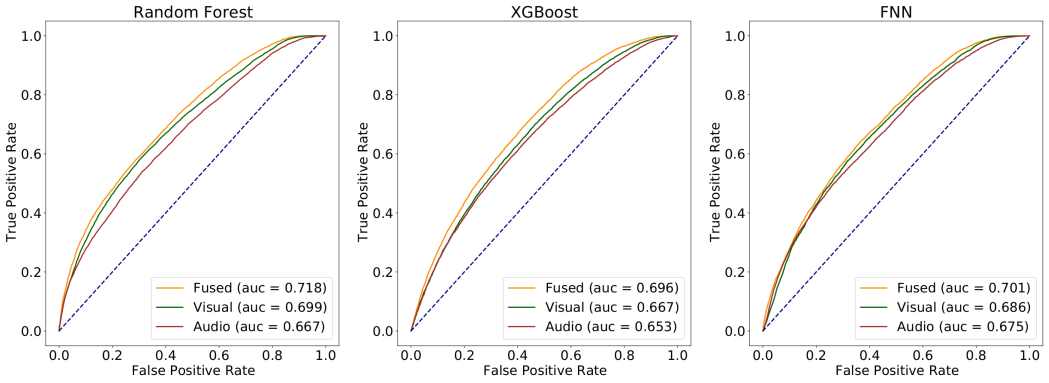

Figure 6. Receiver Operating Characteristic (ROC) curve of the Random Forest classifier (**left**), XGBoost classifier (**middle**), Fully Connected classifier (**right**).

Table 5 presents the final Precision, Recall, F1 and Accuracy for the best classification method (Random Forest), when also using the post-processing approach described in Section 3.4, for different values of the N_1 and N_2 parameters of the filtering process. It can be seen that for $N_1 = 3$ and $N_2 = 5$, F1 is relatively boosted by almost 4%. In terms of the positive class recall and precision rates: the first is increased and the latter is decreased as expected, since the filtering process removes positive predictions that do not match the aforementioned criteria. Overall, the 63% F1 macro-averaged score achieved by the method is reasonable, since the human performance on that metric, as measured from the agreement between the annotators, is 72.8%.

Table 5. Random Forest performance metrics for different parameters of the post-processing technique described in Section 3.4. Best result is indicated in bold.

Thresholds (med (N_1)-hard (N_2))	Precision	Recall	f1 Macro	Accuracy
no	42.2%	69.9%	60.6%	62%
3-3	43.7%	69.7%	62%	63.8%
3-5	**44.9%**	**66.9%**	**63%**	**65.3%**
5-3	43.4%	70.8%	61.8%	63.4%
5-5	44.2%	69.9%	62.5%	64.3%

In addition to the methodology for distilling the summary of a video, emphasis was placed on recognizing the features that contributed the most to the final result of the random forest classifier. By using the Recursive Feature Elimination (RFE), a feature selection algorithm able to rank features with recursive feature elimination, we were able to find the ten most crucial out of the 224 audiovisual features. In Table 6 the ten features with rank 1 by RFE are presented. The purpose of the RFE feature selection procedure, was to find out the key features on which our proposed video summarization classifier algorithm is based.

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

Table 6. The ten most valuable features of prediction.

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

To summarize, the basic conclusions from the previously described experimental procedures are the following:

- Random forest achieves the best classification performance in terms of AUC for the binary classification task in all three modalities (visual, audio and multimodal).
- Visual-based classifier is always almost 4% relatively better than audio.
- Fusion-based classifier is always almost 3% relatively better than visual, which indicates that the two modalities both contain useful information for the summarization task.
- The final performance of the binary classifier after applying the proposed post-processing technique reaches almost 45% precision and 67% recall rate at a 1-second segment level.
- Motion-related features seem to be among the most important with regards to the classifiers' decision, along with some spectral domain audio features and color intensity and saturation features.

6. Conclusions & Future Work

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

As future work, it would be quite interesting to train the proposed method on a robust training data set with a large number of videos from categories annotated from an extensive number of users. An exploration in further modalities, such as text, may accompany a video in places likes the description, title, subtitles or even in the comments. The possible absence of such a huge data set would be interesting to study and start as a data collection process in our methodology. Extension to the machine learning models and deep learning algorithms, with different pre-processing techniques, such as time-series models, can be tested, or a deep neural network can be used as a video feature extractor. Lastly, A/B testing by measuring the final result of the method based on the point of view of users who did not participate in the annotation process, could be explored as a totally different and novel evaluation method, evaluating the videos at the summary level.

Author Contributions: Conceptualization, T.G. and E.S.; methodology, T.G., E.S., T.P. and P.K.; software, T.P. and P.K.; validation, T.G. and E.S.; data curation, T.P. and P.K.; writing—original draft preparation, T.P., P.K., T.G. and E.S.; writing—review and editing, T.P., P.K., T.G. and E.S.; funding acquisition, E.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (project code: 1EDK-02070).

Acknowledgments: The authors would like to thank all anonymous annotators, for their participation in the video annotation process.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAVs	Unnamed Aired Vehicles
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
GAN	Generative Adversarial Network
MFCCs	Mel Frequency Cepstral Coefficients
RGB	Red Green Blue
VAT	Video Annotator Tool
Log Reg	Logistic Regression
KNN	k-Nearest Neighbors
XGBoost	eXtreme Gradient Boosting
FNN	Fully Connected Neural Network
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
RFE	Recursive Feature Elimination

References

1. YouTube in Numbers. Available online: https://www.youtube.com/intl/en-GB/about/press/ (accessed on 20 February 2021).
2. Furini, M.; Ghini, V. An audio-video summarization scheme based on audio and video analysis. In Proceedings of the IEEE CCNC, Las Vegas, Nevada, USA, 8–10 January 2006.
3. Money, A.G.; Agius, H. Video summarisation: A conceptual framework and survey of the state of the art. *J. Vis. Commun. Image Represent.* **2008**, *19*, 121–143. [CrossRef]
4. Xiong, Z.; Radhakrishnan, R.; Divakaran, A.; Yong-Rui, Z.; Huang, T.S. *A Unified Framework for Video Summarization, Browsing & Retrieval: With Applications to Consumer and Surveillance Video*; Elsevier: Amsterdam, The Netherlands, 2006.
5. Lai, P.K.; Décombas, M.; Moutet, K.; Laganière, R. Video summarization of surveillance cameras. In Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 23–26 August 2016; pp. 286–294.
6. Priya, G.L.; Domnic, S. Medical Video Summarization using Central Tendency-Based Shot Boundary Detection. *Int. J. Comput. Vis. Image Process.* **2013**, *3*, 5565. [CrossRef]
7. Trinh, H.; Li, J.; Miyazawa, S.; Moreno, J.; Pankanti, S. Efficient UAV video event summarization. In Proceedings of the 21st IEEE International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 2226–2229.
8. Spyrou, E.; Tolias, G.; Mylonas, P.; Avrithis, Y. Concept detection and keyframe extraction using a visual thesaurus. *Multimed. Tools Appl.* **2009**, *41*, 337–373. [CrossRef]
9. Li, Y.; Merialdo, B.; Rouvier, M.; Linares, G. Static and dynamic video summaries. In Proceedings of the 19th ACM International Conference on Multimedia, Scottsdale, AZ, USA, 28 November–1 December 2011; pp. 1573–1576.
10. Lienhart, R. Pfeiffer, S. Effelsberg, W. The MoCA workbench: Support for creativity in movie content analysis. In Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 17–23 June 1996; pp. 314–321.
11. Chen, B.C.; Chen, Y.Y.; Chen, F. Video to Text Summary: Joint Video Summarization and Captioning with Recurrent Neural Networks. In Proceedings of the BMVC, London, UK, 4–7 September 2017.
12. Smith, M.A.; Kanade, T. *Video Skimming for Quick Browsing Based on Audio and Image Characterization*; School of Computer Science, Carnegie Mellon University: Pittsburgh, PA, USA, 1995.
13. Sen, D.; Raman, B. Video skimming: Taxonomy and comprehensive survey. *arXiv* **2019**, arXiv:1909.12948.
14. Zhou, K.; Qiao, Y.; Xiang, T. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2018; Volume 32.
15. Zhang, K.; Chao, W.L.; Sha, F.; Grauman, K. Video summarization with long short-term memory. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 766–782.
16. Evangelopoulos, G.; Zlatintsi, A.; Potamianos, A.; Maragos, P.; Rapantzikos, K.; Skoumas, G.; Avrithis, Y. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Trans. Multimed.* **2013**, *15*, 1553–1568. [CrossRef]
17. Wei, H.; Ni, B.; Yan, Y.; Yu, H.; Yang, X.; Yao, C. Video summarization via semantic attended networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 2–9 February 2018; Volume 32.
18. Grundmann, M.; Kwatra, V.; Han, M.; Essa, I. Efficient hierarchical graph-based video segmentation. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2141–2148.
19. Pantazis, G.; Dimas, G.; Iakovidis, D.K. SalSum: Saliency-based Video Summarization using Generative Adversarial Networks. *arXiv* **2020**, arXiv:2011.10432.
20. Jacob, H.; Pádua, F.L.; Lacerda, A. Pereira, A.C. A video summarization approach based on the emulation of bottom-up mechanisms of visual attention. *J. Intell. Inf. Syst.* **2017**, *49*, 193–211. [CrossRef]
21. Cirne, M.V.M.; Pedrini, H. VISCOM: A robust video summarization approach using color co-occurrence matrices. *Multimed. Tools Appl.* **2018**, *77*, 857–875. [CrossRef]
22. Rochan, M.; Ye, L.; Wang, Y. Video summarization using fully convolutional sequence networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 347–363.
23. Otani, M.; Nakashima, Y.; Rahtu, E.; Heikkilä, J.; Yokoya, N. Video summarization using deep semantic features. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; p. 361377.
24. Wu, J.; Zhong, S.H.; Jiang, J.; Yang, Y. A novel clustering method for static video summarization. *Multimed. Tools Appl.* **2017**, *76*, 9625–9641. [CrossRef]
25. Potapov, D.; Douze, M.; Harchaoui, Z.; Schmid, C. Category-specific video summarization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 540–555.
26. Ma, Y.F.; Lu, L.; Zhang, H.J.; Li, M. A user attention model for video summarization. In Proceedings of the Tenth ACM International Conference on Multimedia, Juan les Pins, France, 1–6 December 2002; pp. 533–542.
27. Mahasseni, B. Lam, M. Todorovic, S. Unsupervised video summarization with adversarial lstm networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 202–211.

28. Song, Y.; Vallmitjana, J.; Stent, A.; Jaimes, A. Tvsum: Summarizing web videos using titles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5179–5187.
29. Gygli, M.; Grabner, H.; Riemenschneider, H.; Van Gool, L. Creating summaries from user videos. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 505–520.
30. Lee, Y.J.; Ghosh, J.; Grauman, K. Discovering important people and objects for egocentric video summarization. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1346–1353.
31. De Avila, S.E.F.; Lopes, A.P.B.; da Luz, A., Jr.; de Albuquerque Araújo, A. VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognit. Lett.* **2011**, *32*, 56–68. [CrossRef]
32. Alías, F.; Socoró, J.C.; Sevillano, X. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Appl. Sci.* **2016**, *6*, 143. [CrossRef]
33. Giannakopoulos, T. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS ONE* **2015**, *10*, e0144610. [CrossRef] [PubMed]
34. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001; Volume 1, p. I.
35. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), Vancouver, BC, Canada, 24–28 August 1981.
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
37. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
38. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **2017**, *18*, 559–563.
39. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; ACM: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
40. Bottou, L.; Lin, C.J. Support vector machine solvers. *Large Scale Kernel Mach.* **2007**, *3*, 301–320.
41. List, N.; Simon, H.U. SVM-optimization and steepest-descent line search. In Proceedings of the 22nd Annual Conference on Computational Learning Theory, Montreal, QC, Canada, 18–21 June 2009.