

Viewpoint-aware Video Summarization

Atsushi Kanehira¹, Luc Van Gool^{3,4}, Yoshitaka Ushiku¹, and Tatsuya Harada^{1,2}

¹The University of Tokyo, ²RIKEN, ³ETH Zürich, ⁴KU Leuven

Abstract

This paper introduces a novel variant of video summarization, namely building a summarization that depends on the particular aspect of a video the viewer focuses on. We refer to this as *viewpoint*. In order to infer what the desired viewpoint may be, we assume that several other videos are available, especially groups of videos, e.g. as folders on a person’s phone or laptop. The semantic similarity between videos in a group vs. the dissimilarity between groups is used to produce viewpoint-specific summaries. For considering similarity as well as avoiding redundancy, output summary should be (A) diverse, (B) representative of videos in the same group, and (C) discriminative against videos in the different groups.

To satisfy these requirements (A)-(C) simultaneously, we proposed a novel video summarization method from multiple groups of videos. Inspired by Fisher’s discriminant criteria, it selects summary by optimizing the combination of three terms (a) inner-summary, (b) inner-group, and (c) between-group variances defined on the feature representation of summary, which can simply represent (A)-(C). Moreover, we developed a novel dataset in order to investigate how well the generated summary reflects the underlying viewpoint. Quantitative and qualitative experiments conducted on the dataset demonstrates the effectiveness of proposed method.

1. Introduction

An enormous number of videos have become available owing to the recent spread of Internet services (e.g., YouTube) and inexpensive cameras, which make it difficult to verify all content. Thus, video summarization, which compresses a video by extracting the *important* parts while avoiding redundancy, has attracted the attention of many researchers.

The information deemed *important* can be varied based on the particular aspect the viewer focuses on, which hereafter we will refer to as *viewpoint* in this paper. Please note

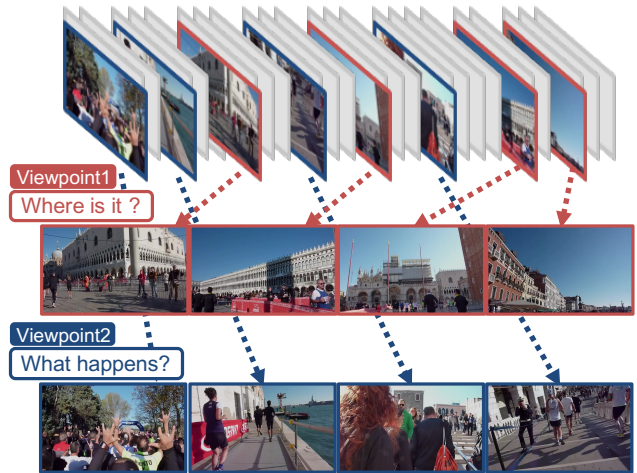


Figure 1: Many types of summaries can exist for one video. If we are given a video in which the running events take place in Venice, an optimal summary is represented based on the *viewpoint* toward it.

it does not mean physical position. For instance, given the video in which the running events take place in Venice, as shown in Fig. 1, if we watch it focusing on the “kind of activity,” the scene in which many runners come across in front of the camera is important. Alternatively, if the attention is focused on “place,” the scene that shows a beautiful building may be more important. Such *viewpoints* may not be limited to explicit ones stated in the above examples, and in this sense, the optimal summary for one video is not necessarily determined in only one way.

Most existing methods, however, assume there is only one optimal summary for one video. Even though the variance between subjects are considered by comparing multiple human-created summaries during evaluation, it is difficult to determine how well the *viewpoint* is considered.

Although several different ways may exist for interpreting a *viewpoint*, this paper takes the approach of dealing with it by considering the *similarity*, which represents what we feel that is similar or different, and has a close relationship with the *viewpoint*. For example, as shown in Fig. 2,

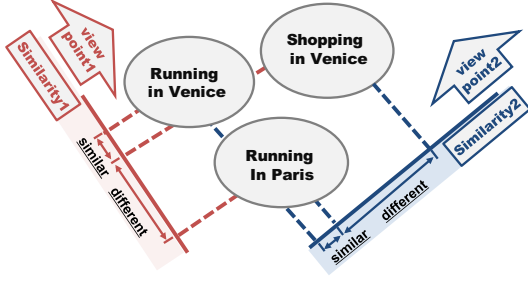


Figure 2: Conceptual relationship between a *viewpoint* and *similarity*. This paper assumes a *similarity* is derived from a corresponding *viewpoint*.

“running in Paris” is closer to “running in Venice” than “shopping in Venice” from the *viewpoint* of the “kind of activity,” but such a relationship will be reversed when the *viewpoint* changes to “place.” Here, we use the word *similarity* to indicate the one that captures semantic information rather than the appearance, and importantly, it is changeable depending on the *viewpoint*. We aim to generate a summary considering such *similarities*. A natural question here is “where does the *similarity* come from?”

We may be able to obtain it by asking someone whether two frames are similar or different for all pairs of frames (or short clips). However, a *similarity* changes depending on its *viewpoint*. Therefore, it is unrealistic to obtain frame-level similarity for all *viewpoints* in this manner.

This paper particularly focuses on video-level *similarities*. More specifically, we utilize the information of how multiple videos are divided to groups as an indicator of *similarity* because of its accessibility. For example, we have multiple video folders on our PCs or smart-phones, or we sometimes categorize videos on an Internet service. They are divided according to a reason, but in most cases, why they are grouped the way they are is unknown, or irrelevant of criteria such as preference (liked or not liked). Thus, a *viewpoint* is not generally evident, but such video-level *similarities* can be measured as a mapping of *one viewpoint*.

In this paper, we assume the situation that multiple groups of videos that are divided based on *one similarity* are given, and we investigate how to introduce unknown underlying *viewpoints* to the summary. It is worth noting that, as we assume there are multiple possible ways to divide videos into groups depending on a *viewpoint* given the same set of videos, some overlap of content can exist between videos belonging to different groups, leading to technical difficulties, as we will state in Section 2.

In order to consider *similarities*, summaries extracted from similar videos should be similar, and ones extracted from different videos should be different from each other in addition to avoiding the redundancy derived from the original motivation of video summarization. In other words, given multiple groups of videos, the output summary of the video summarization algorithm should be: (A) diverse, (B)

representative of videos in the same group, and (C) discriminative against videos in the different groups.

To satisfy the requirements (A)-(C) simultaneously, we proposed a novel video summarization method from multiple groups of videos. Inspired by Fisher’s discriminant criteria, it selects a summary by optimizing the combination of three terms the (a) inner-summary, (b) inner-group, and (c) between-group variance defined based on the feature representation of the summary, which can simply represent (A)-(C). In addition, we developed a novel optimization algorithm, which can be easily combined with feature learning, such as using convolutional neural network (CNN).

Moreover, we developed a novel dataset in order to investigate how well the generated summary reflects an underlying *viewpoint*. Since knowing individual *viewpoints* is generally impossible, we fixed it to two types of topics for each video. We also collected multiple videos that can be divided to groups based on these *viewpoints*. Quantitative and qualitative experiments were conducted on the dataset to demonstrate the effectiveness of proposed method.

The contributions of this paper are as follows:

- Propose a novel video summarization method from multiple groups of videos where their *similarities* are taken into consideration,
- Develop a novel dataset for quantitative evaluation
- Demonstrate the effectiveness of proposed method by quantitative and qualitative experiments on the dataset.

The remainder of this paper is organized as follows. In Section 2, we discuss the related work of video summarization. Further, we explain the formulation and optimization of our video summarization method in Section 3. We state the detail of the dataset we created in Section 4, and describe and discuss the experiments that we performed on it in Section 5. Finally, we conclude our paper in Section 6.

2. Related work

Many recent studies have tackled the video summarization problem, and most of them can be classified into an either unsupervised or supervised approach. Unsupervised summarization [23, 20, 19, 21, 1, 5, 35, 11, 13, 14, 28, 22] that creates a summary using specific selection criteria, has been conventionally studied. However, owing to the subjective property of this task, a supervised approach [17, 30, 26, 18, 9, 10, 15, 6, 34], that trains a summarization model which takes human-created summaries as the supervision, became standard because of its better performance. Most of their methods aim to extract one optimal summary and do not consider the *viewpoint*, which we focus on in this study.

The exception is query extractive summarization [27] whose model takes a keyword as input and generates a summary based on it. It is similar to our work in that it assumes

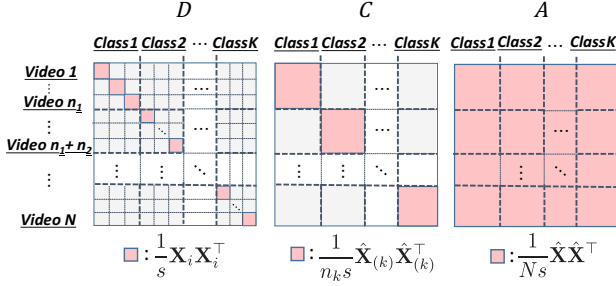


Figure 3: Overview of matrices D, C, and A, which are similarity matrices of inner-video, inner-group, and all videos. Non-zero elements of each matrix are colored pink and zero elements are colored gray.

there can be multiple kinds of summaries for one video. However, our work is different in that we estimate the *view-point* from the data instead of taking it as input. In addition, this requires frame-level importance annotation for the training model, which is unrealistic for real applications.

Some of the previous research on the subject worked on video summarization utilizing only other videos to alleviate the difficulty of building a dataset [3, 24, 25]. [3, 25] utilized other similar videos and aims to generate a summary that is (A) diverse, and (B) representative of videos in a similar group, but it is not considered to be (C) discriminative against videos in different groups. Given that not only *what is similar* but also *what is different* is important to consider in terms of *similarity*, we attempt to generate a summary that meets all of the conditions, (A)-(C).

The research most relevant to ours is [24], which attempted to introduce discriminative information by utilizing a trained video classification model. It generates a summary with two steps. In the first step, it trains a spatio-temporal CNN that classifies the category of each video. In the second step, it calculates importance scores by spatially and temporally aggregating the gradients of the network’s output with regard to the input over clips.

The success of this method has a strong dependence on the training in the first step. In this step, training is performed clip-by-clip by assigning the same label as that the video belongs to, to all clips of the video. Thus, it implicitly assumes all clips can be classified to the same group, and if there are some clips that are difficult to classify, it suffers from over-fitting caused by trying to classify it correctly. Such a strong assumption does not apply in general, because generic videos (such as ones on YouTube) include various types of content. This assumption does not also apply in our case because we are interested in the situation where there are multiple possible ways to divide videos into groups given the same set of videos, as stated in Section 1, where some parts of videos can overlap with ones belonging to different groups for some *viewpoints*.

Unlike this method, we do not assume that all clips in

the video should be able to be classified correctly. Instead, our method considers the discrimination for only parts of videos. This makes it easy to find discriminative information even when there are visually similar clips across different groups.

3. Method

First, we introduce three quantities, that is, the (a) inner-summary, (b) within-group, and (c) between-group variances in subsection 3.1. Subsequently, we formulate our method by defining a loss function to meet the requirements discussed in Section 1. The optimization algorithm is described in subsection 3.2, and how to combine it with CNN feature learning is mentioned in subsection 3.3. Also, the detailed derivation of the formula can be found in the supplemental material.

3.1. Formulation

Let $\mathbf{X}_i = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T_i}]^\top \in \mathbb{R}^{T_i \times d}$ be a feature matrix for a video i with T_i segment (or frame) features \mathbf{x} . Our goal is to select s segments from the video. We start by defining the feature representation of the summary for video i as $\mathbf{v}_i = \frac{1}{s} \mathbf{X}_i^\top \mathbf{z}_i$, where $\mathbf{z}_i \in \{0, 1\}^{T_i}$ is the indicator variable and $z_{it} = 1$ if the t -th segment is selected, and otherwise 0. It also has a constraint $\|\mathbf{z}_i\|_0 = s$ indicating that just s segments are selected as a summary. We can define a variance S_i^V for the summary of a video i as

$$S_i^V = \sum_{t=1}^{T_i} z_t (\mathbf{x}_t - \mathbf{v}_i) (\mathbf{x}_t - \mathbf{v}_i)^\top. \quad (1)$$

Thus, its trace can be written as:

$$Tr(S_i^V) = \sum_{t=1}^{T_i} z_t \mathbf{x}_t^\top \mathbf{x}_t - \frac{1}{s} \mathbf{z}_i^\top \mathbf{X}_i \mathbf{X}_i^\top \mathbf{z}_i. \quad (2)$$

Placing all N videos together by using a stacked variable $\hat{\mathbf{z}} = [\mathbf{z}_1^\top, \mathbf{z}_2^\top, \dots, \mathbf{z}_N^\top]^\top \in \{0, 1\}^{\sum_{i=1}^N T_i}$, we can rewrite

$$Tr(S^V) = \sum_{i=1}^N Tr(S_i^V) = \hat{\mathbf{z}}^\top (F - D) \hat{\mathbf{z}}. \quad (3)$$

where F is a diagonal matrix whose element corresponds to $\mathbf{x}_t^\top \mathbf{x}_t$, and $D = \frac{1}{s} \oplus \sum_{i=1}^N \mathbf{X}_i \mathbf{X}_i^\top$ is a block diagonal matrix containing a similarity matrix of segments in the video i as i -th block elements.

By exploiting categorical information, we can also compute within-group variance S^W and between-group variance S^B . To compute them, we define the mean vector $\boldsymbol{\mu}_k$

for group $k \in \{1 : K\}$ and global mean vector $\bar{\mu}$ as:

$$\mu_k = \frac{1}{n_k} \sum_{i \in L(k)} \mathbf{v}_i = \frac{1}{n_k s} \hat{\mathbf{X}}_{(k)}^\top \hat{\mathbf{z}}_{(k)}, \quad (4)$$

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i = \frac{1}{N s} \hat{\mathbf{X}}^\top \hat{\mathbf{z}}, \quad (5)$$

respectively. In these equations, $L(k)$ is the set of indices of videos belonging to group k and $n_k = |L(k)|$ (i.e., $N = \sum_k n_k$). In addition, $\hat{\mathbf{X}} = [\mathbf{X}_1^\top | \mathbf{X}_2^\top | \dots | \mathbf{X}_N^\top]^\top \in \mathbb{R}^{(\sum_{i=1}^N T_i) \times d}$ is the matrix stacking all segment features of all videos. $\hat{\mathbf{X}}_{(k)}$ and $\hat{\mathbf{z}}_{(k)}$ are parts of $\hat{\mathbf{X}}$ and $\hat{\mathbf{z}}$, respectively, corresponding to videos contained by group k . We assume that a video index is ordered to satisfy $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_{(1)}^\top | \hat{\mathbf{X}}_{(2)}^\top | \dots | \hat{\mathbf{X}}_{(K)}^\top]^\top$. Here, the trace of within-group variance for group k can be written as:

$$\begin{aligned} Tr(S_{(k)}^W) &= Tr\left(\sum_{i \in L(k)} s(\mathbf{v}_i - \mu_k)(\mathbf{v}_i - \mu_k)^\top\right) \\ &= \frac{1}{s} \sum_{i \in L(k)} \mathbf{z}_i^\top \mathbf{X}_i \mathbf{X}_i^\top \mathbf{z}_i - \frac{1}{n_k s} \hat{\mathbf{z}}_{(k)}^\top \hat{\mathbf{X}}_{(k)} \hat{\mathbf{X}}_{(k)}^\top \hat{\mathbf{z}}_{(k)}. \end{aligned} \quad (6)$$

Aggregating them over all groups, the trace of within-group variance takes the following form:

$$Tr(S^W) = \sum_{k=1}^K Tr(S_{(k)}^W) = \hat{\mathbf{z}}^\top (D - C) \hat{\mathbf{z}}. \quad (7)$$

$C = \frac{1}{s} \oplus \sum_{k=1}^K \frac{1}{n_k} \hat{\mathbf{X}}_{(k)} \hat{\mathbf{X}}_{(k)}^\top$ is a block diagonal matrix containing a similarity matrix of segments in the video belonging to group k as a k -th block element. Similarly, the trace of between-group variance is:

$$\begin{aligned} Tr(S^B) &= Tr\left(\sum_{k=1}^K n_k s (\mu_k - \bar{\mu})(\mu_k - \bar{\mu})^\top\right) \\ &= \hat{\mathbf{z}}^\top (C - A) \hat{\mathbf{z}}. \end{aligned} \quad (8)$$

In addition, matrix A is defined by $A = \frac{1}{N s} \hat{\mathbf{X}} \hat{\mathbf{X}}^\top$. We show the overview of matrices D , C , and A in Fig. 3.

Loss function: We designed an optimization problem to meet the requirements discussed in Section 1: (A) diverse, (B) representative of videos in the same group, and (C) discriminative against videos in different groups. To simultaneously meet these requirements, we minimized the within-group variance while maximizing the between-group and inner-video variances inspired by the concept of linear discriminant analysis. Thus, we maximized the following function, which is the weighted sum of the aforementioned three variances:

$$\begin{aligned} &\lambda_1 Tr(S^V) - \lambda_2 Tr(S^W) + \lambda_3 Tr(S^B) \\ \text{s.t. } &\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0 \end{aligned} \quad (9)$$

Algorithm 1 Optimization algorithm of (11)

- 1: **INPUT:** data matrix $Q = Q_1 - Q_2$, the number of selected clips s .
 - 2: **INITIALIZE:** $\mathbf{z}_i = (1/s) \mathbf{1}_{T_i}$ for all video index i .
 - 3: **repeat**
 - 4: Calculate upper bound $\hat{L}_{(t)} = \hat{\mathbf{z}}^\top Q_1 \hat{\mathbf{z}} - 2 \hat{\mathbf{z}}_{(t)}^\top Q_2 \hat{\mathbf{z}}$
 - 5: Replace loss with $\hat{L}_{(t)}$ and solve QP problem.
 - 6: **until** convergence
 - 7: **RETURN** $\hat{\mathbf{z}}$
-

where $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters that control the importance of each term. We empirically fixed $\lambda_1 = 0.05$ in our experiments.

By substituting (3), (7), and (8) into (9), the optimization problem can be solved as:

$$\begin{aligned} \min \quad & \hat{\mathbf{z}}^\top Q \hat{\mathbf{z}} \\ Q \triangleq \quad & -\lambda_1 F + (\lambda_1 + \lambda_2) D - (\lambda_2 + \lambda_3) C + \lambda_3 A \\ \text{s.t.} \quad & \|\mathbf{z}_i\|_0 = s, \mathbf{z}_i \in \{0, 1\}^{T_i}, \forall i \in \{1 : N\} \end{aligned} \quad (10)$$

3.2. Optimization

Given that minimizing (10) directly is infeasible, we relaxed it to a continuous problem as follows:

$$\begin{aligned} \min \quad & \hat{\mathbf{z}}^\top Q \hat{\mathbf{z}} \\ \text{s.t.} \quad & P \hat{\mathbf{z}} = s \mathbf{1}_N, \hat{\mathbf{z}} \in [0, 1]^{\sum_{i=1}^N T_i} \end{aligned}$$

where $P^\top = \begin{bmatrix} \mathbf{1}_{T_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{T_2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1}_{T_N} \end{bmatrix} \quad (11)$

$\mathbf{1}_a$ indicates a vector whose elements are all ones and whose size is a , and the size of matrix P is $N \times \sum_{i=1}^N T_i$. The designed optimization problem is the difference of convex (DC) programming problem because all matrices that compose Q in (11) are positive semi-definite. We utilized a well-known CCCP (concave convex procedure) algorithm [32, 33] to solve it. Given the loss function represented by $L(\mathbf{x}) = f(\mathbf{x}) - g(\mathbf{x})$ where $f(\cdot)$ and $g(\cdot)$ are convex functions, the algorithm iteratively minimizes the upper bound of loss calculated by the linear approximation of $g(\mathbf{x})$. Formally, in the iteration t , it minimizes: $\hat{L}(\mathbf{x}) = f(\mathbf{x}) - \partial_{\mathbf{x}} g(\mathbf{x}_{(t)})^\top \mathbf{x} \geq L(\mathbf{x})$. In our problem, the loss function can be decomposed into the difference of two convex functions: $\hat{\mathbf{z}}^\top Q \hat{\mathbf{z}} = \hat{\mathbf{z}}^\top Q_1 \hat{\mathbf{z}} - \hat{\mathbf{z}}^\top Q_2 \hat{\mathbf{z}}$, where $Q_1 \triangleq (\lambda_1 + \lambda_2) D + \lambda_3 A$ and $Q_2 \triangleq \lambda_1 F + (\lambda_2 + \lambda_3) C$. We optimized the following quadratic programming (QP) problem in t -th iteration,

$$\begin{aligned} \min \quad & \hat{\mathbf{z}}^\top Q_1 \hat{\mathbf{z}} - 2 \hat{\mathbf{z}}_{(t)}^\top Q_2 \hat{\mathbf{z}} \\ \text{s.t.} \quad & P \hat{\mathbf{z}} = s \mathbf{1}_N, \hat{\mathbf{z}} \in [0, 1]^{\sum_{i=1}^N T_i} \end{aligned} \quad (12)$$

where $\hat{\mathbf{z}}_{(t)}$ is the estimation of $\hat{\mathbf{z}}$ in the t -th iteration. In our implementation, we used a CVX package [8, 7] to solve the QP problem (12). An overview of our algorithm is shown in Algorithm 1. Please refer [16] for the convergence property of CCCP.

3.3. Feature learning

In order to obtain a feature that is more suitable for video summarization, we developed a novel feature learning method. Firstly, we replace the visual feature \mathbf{x} in subsection 3.1 to $f(\mathbf{x}; \mathbf{w})$ where $f(\cdot)$ is a feature extractor function that is differentiable with regard to the parameter \mathbf{w} and the input \mathbf{x} is a sequence of raw frames in the RGB space. Specifically, we exploited the C3D network [31] as a feature extractor. Fixing $\hat{\mathbf{z}}$, the loss function in (11) can be written as:

$$L = \sum_{i,j} \hat{z}_i \hat{z}_j m_{ij} f(\mathbf{x}_i)^\top f(\mathbf{x}_j) \quad (13)$$

where \hat{z}_i is i -th element of $\hat{\mathbf{z}}$. Also, m_{ij} is the ij -th element of matrix M written as follows:

$$M = -\lambda_1 \mathbf{1}_F + (\lambda_1 + \lambda_2) \mathbf{1}_D - (\lambda_2 + \lambda_3) \mathbf{1}_C + \lambda_3 \mathbf{1}_A.$$

Here, $\mathbf{1}_X$ represents an indicator matrix whose element takes 1 where the corresponding element of X is not 0, and takes 0 otherwise. We optimized the loss function with regard to the parameter by stochastic gradient decent (SGD). Because many of \hat{z}_i are small values or zeros, direct minimization (13) is not efficient. We avoid the inefficiency by sampling samples \mathbf{x}_i based on their weight \hat{z}_i . Given $\sum \hat{z}_i = Ns$, we sampled \mathbf{x}_i from the distribution $p(\mathbf{x}_i) = \hat{z}_i / Ns$ (≥ 0) and stochastically minimized the expectation:

$$\mathbb{E}_{\mathbf{x}_i, \mathbf{x}_j \sim p(\mathbf{x})} [m_{ij} f(\mathbf{x}_i)^\top f(\mathbf{x}_j)] \quad (14)$$

In an iteration when updating parameters, the model fetches pairs of $(\mathbf{x}_i, \mathbf{x}_j)$ and computes the dot product of the feature representations. The loss for this batch is calculated by summing up the dot product weighted by m_{ij} . We repeatedly and alternately compute the summary via the Algorithm 1 and optimize the parameter of the feature extractor.

4. Dataset

The motivation of this study is the claim that an optimal summary should be varied depending on a *viewpoint*, and this paper deals with this by considering the *similarities*. To investigate how well the underlying *viewpoints* are taken into consideration, given multiple groups of videos that are divided based on the *similarity*, we developed a novel video summarization dataset¹. Its quantitative evaluation is challenging because the *viewpoint* is generally unknown. Thus,

¹Dataset will be available soon.

for the purpose of quantitative evaluation, we collected a set of videos that can have two interpretable ways of separation assuming they have corresponding *viewpoints*. In addition, we collected human-created summaries fixing the importance criteria to two concepts based on each *viewpoint*. The procedure of building the dataset is as follows:

First, we collected five videos that match the topics written in **target group (TG)**, **other group1 (OG1)**, **other group2 (OG2)** of Table 1 by retrieving them in YouTube² using a keyword. Each of **TG**, **OG1**, **OG2** has two explicit concepts chosen, such as location, activity, object, and scene such that they can be visually confirmed. The concepts of **TG** are written in **concept1** and **concept2** columns in the table, and both **OG1** and **OG2** are chosen to share either one of them. There are two interpretable ways to divide these sets of videos, i.e., **(TG + OG1)** vs. **(OG2)** and **(TG + OG2)** vs. **(OG1)** because **OG1** and **OG2** share one topic with **TG**. Assuming these divisions are based on one *viewpoint*, we collect the summary based on it using two concepts for videos belonging to **TG**. For example, if we are given two groups, one of which contains “running in Venice” and “running in Paris” videos, and the other group includes “shopping in Venice” videos, the underlying “viewpoint” is expected to be “kind of activity.” For such a division, we collect summaries based on “running” for the videos of “running in Venice.”

For annotating the importance of each frame of the video belonging to **TG**, we used Amazon Mechanical Turk (AMT). Firstly, videos were evenly divided into clips beforehand so that the length of each clip was two seconds long following the setting of [28]. Subsequently, after workers watched a whole video, they were asked to assign importance scores to each clip of the video, assuming that they created a summary based on a pre-determined topic, which corresponds to the concept written in **concept1** or **concept2** columns in the Table 1. Importance scores are chosen from 1 (not important) to 3 (very important), and workers were asked to guarantee the number of clips having a score of 3 falls in the range between 10% and 20% of the total number of clips in the video. For each video and each *concept*, five workers were assigned.

We display the statistics of the dataset and some example of the human-created summary in Table 2 and Fig. 4, respectively. Also, in order to investigate how similar and far the assigned score between subjects is, we calculate the similarity of the score vector. After subtracting the mean value from each score, the mean cosine similarity for the pair of scores that are assigned for the same concepts (e.g., **concept1** and **concept1**) and different concepts (**concept1** and **concept2**) are separately computed, and the result is shown in Fig. 5. As we can see in the table, the similarity of scores that comes from the inner-concept is higher than

²<https://www.youtube.com/>

Table 1: List of the name of video groups (**target group**, **other group1**, **other group2**), and individual concepts of **target group** (**concept1**, **concept2**) that are used to base the summary on. We omit the article (e.g., the) before nouns due to the lack of space. We use the abbreviation of target group as [RV, RB, BS, DS, RD, SR, CC, RN, SC, RS] from top to bottom.

target group (TG)	concept1	concept2	other group1 (OG1)	other group2 (OG2)
running in Venice	Venice	running	running in Paris	shopping in Venice
riding bike on beach	beach	riding bike	riding bike in city	surfing on beach
boarding on snow mountain	snow mountain	boarding	boarding on dry sloop	hike in snow mountain
dog chasing sheep	sheep	dog	dog playing with kids	sheep grazing grass
racing in desert	desert	racing	racing in circuit	riding camel in desert
swimming and riding bike	swimming	riding bike	riding bike and tricking	diving and swimming
catching and cooking fish	catching fish	cooking fish	cooking fish in village	catching fish at river
riding helicopter in NewYork	NewYork	helicopter	riding helicopter in Hawaii	riding ship in NewYork
slackline and rock climbing	slackline	rock climbing	rock climbing and camping	slcakline and jaggling
riding horse in safari	safari	riding horse	riding horse in mountain	riding vehicle in safari

Table 2: statistics of dataset

group	# of videos	# of frames	duration
TG	50	243,873	8,832(s)
OG1 + OG2	100	440,330	15,683(s)

inter-concept, which indicates that the importance depends on the viewpoint of the videos.

5. Experiment

5.1. Preprocessing

To compute the segment that is used as the smallest element for video summarization, we followed a simple method proposed in [3]. After counting the difference of two consecutive frames in the RGB and HSV space, the points on which the total amount of change exceeds 75% of all pixels were regarded as change points. Subsequently, we combined short clips into a subsequent clip and evenly divided the long clips in order such that the number of frames in each clip is more than 32 and less than 112.

5.2. Visual features

In order to obtain frame-level visual features, we exploited the intermediate state of the C3D [31] network, which is known to be so generic that it can be used for other tasks, including video summarization [25]. We extracted the features from an fc6 layer of a network pre-trained on a Sports1M [12] dataset. The length of the input is 16 frames, and features are extracted every 16 frames. The dimension of the output feature vector is 4,096.

Clip-level representations were calculated by performing an average pooling over all frame-level features in each clip followed by a l_2 normalization.

5.3. Evaluation

For a quantitative evaluation, we compared automatically generated summaries with human made ones.

First, we explain the grouping setting of videos. There are two possible ways of grouping that include each target group as stated in Section 4:

- regarding **other group2 (OG2)** as the same group as **target group (TG)** and **other group1 (OG1)** as the different group (setting1).
- regarding **other group1 (OG1)** as the same group as **target group (TG)** and **other group2 (OG2)** as the different group (setting2).

In the case that the grouping setting1 is used, we evaluated it with the summary annotated for **concept1**. Alternatively, when videos are divided like setting2, the summary for **concept2** was used for the evaluation. Please note we treat each **TG** independently in throughout this experiment.

We set the ground-truth summary in the following procedure. The mean of the importance scores were calculated over all frames in each clip, which is determined by the method described in the previous subsection. The top-30% of the number of all clips whose importance scores are highest were extracted from each video and regarded as ground-truth. As an evaluation metric, we computed the mean Average Precision (MAP) from a pair of summaries, and reported the mean value. Formally, for each **TG**, $1/(CIJ) \sum_{c=1}^C \sum_{j=1}^J \sum_{i=1}^I AP(l_{(c)}^{ij}, \hat{l}_{(c)}^i)$ is calculated where l and \hat{l} are ground-truth summaries and the predicted summary, respectively. C indicates the number of concepts on which the summary created by the annotators is based on. I, J are the number of subjects and the number of videos in the group respectively. In particular, (C, I, J) are (2, 5, 5) as written in Section 4 in this study.

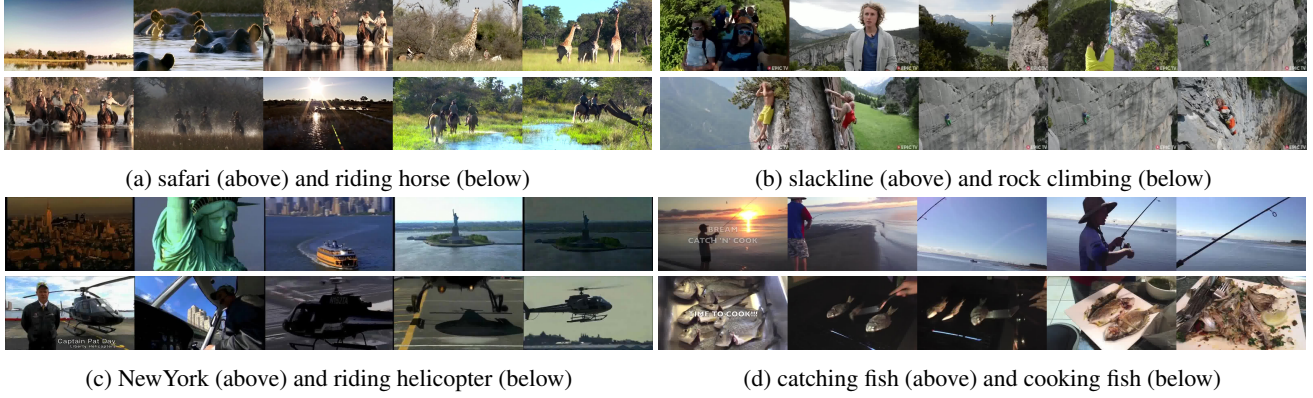


Figure 4: Example human-created summary of video whose **target group** are “riding horse in safari” (upper left), “slackline and rock climbing” (upper right), “riding helicopter in NewYork” (lower left), and “catching and cooking fish” (lower right) based on the concept written in each figure.

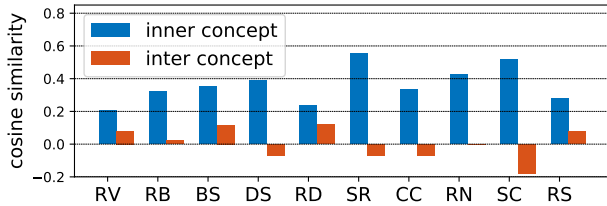


Figure 5: Mean cosine similarity of human assigned scores for each target group. We denote the value computed from the score pairs that are assigned to the same concept and different concepts as inner concepts (blue) and inter concepts (orange), respectively. When referring to the abbreviated names of groups, please refer to the Table 1.

5.4. Implementation detail

As stated in Section 3, we used a C3D network [31] pre-trained on a Sports1M dataset [12], which has eight convolution layers followed by three fully connected layers. During fine-tuning, the initial learning rate was 10^{-5} . Weight decay and momentum were set to 10^{-4} and 0.9 respectively. The number of repetitions of the feature learning and summary estimation was set to 5. The number of epochs for each repetition was 10, and the learning rate was multiplied by 0.9 for every epoch. Here, we define *epoch* as $\{\# \text{ of all clips}\} / \{\text{batch size}\}$ iteration even though clips are not uniformly sampled. We implemented it with the keras library [2].

5.5. Comparison with other methods

To investigate the effectiveness of the proposed method, we compared it with other baseline methods as follows:

Sparse Modeling Representative Selection (SMRS) [4]: SMRS computes a representation of video clips such that a small number of clips can represent an entire video by group sparse regularization. We selected clips whose l_2 norm of representation was the largest.

kmeans (CK) and spectral clustering (CS): One simple solution to extract representative information between multiple videos is applying clustering algorithm. We applied two clustering algorithms, namely kmeans (CK) and spectral clustering (CS), for all clips of video which was regarded as the same groups. RBF kernel was used to build an affinity matrix necessary for computation of spectral clustering. The number of clusters was set to 20 as in [24]. Summaries were generated by selecting clips that are closest to the cluster center of the largest clusters.

Maximum Bi-Clique Finding (MBF) [3]: The MBF is a video co-summarization algorithm that extracts a bi-clique from a bi-partite graph with a maximum inner weight. MBF algorithms are applied to each pair of videos within a video group, and the quality scores are computed by aggregating the results of all pairs. The hyper-parameters are the same as the ones suggested in the original paper [3].

Collaborative Video Summarization (CVS) [25]: CVS is the method that computes a representation of a video clip based on sparse modelling, similar to SMRS. The main difference is that CVS aims to extract a summary that is representative of other videos belonging to the same group as well as the video. We select the clips whose l_2 norm of representation is the largest. The decision of hyper-parameters follows the original paper [25].

Weakly Supervised Video Summarization (WSVS) [24]: Similar to our method, WSVS creates a summary using multiple groups. It computes the importance score by calculating the gradient of the classification network with regard to the input space, and aggregating it over a clip. To train the classification network, the learning setting, network structure, and data augmentation, the technique followed the original paper [24]. For a fair comparison, we leveraged the same network as the one we used as well as the one proposed in the original paper pre-trained on split-1 of the UCF101 [29] dataset (denoted as WSVS (large) and WSVS respectively). Moreover, all clips were used for

Table 3: top-5 Mean AP computed from human-created summary and predicted summary for each method. Results are shown for each **target group**. For referring to the abbreviated names of groups, please see the Table 1.

	RV	RB	BS	DS	RD	SR	CC	RN	SC	RS	mean
SMRS [4]	0.318	0.371	0.338	0.314	0.283	0.317	0.294	0.348	0.348	0.286	0.322
CK	0.329	0.321	0.291	0.269	0.318	0.271	0.275	0.295	0.305	0.268	0.294
CS	0.318	0.330	0.309	0.317	0.278	0.293	0.302	0.355	0.350	0.271	0.312
MBF [3]	0.387	0.332	0.345	0.316	0.319	0.324	0.375	0.317	0.324	0.288	0.333
CVS [25]	0.339	0.365	0.388	0.334	0.359	0.386	0.362	0.303	0.337	0.356	0.353
WSVS [24]	0.333	0.339	0.310	0.331	0.272	0.335	0.336	0.303	0.329	0.330	0.322
WSVS (large) [24]	0.331	0.350	0.322	0.294	0.304	0.306	0.308	0.322	0.342	0.310	0.319
ours	0.373	0.382	0.367	0.396	0.327	0.497	0.374	0.340	0.368	0.368	0.379
ours (feature learning)	0.372	0.376	0.299	0.403	0.373	0.518	0.388	0.338	0.408	0.378	0.385

Table 4: User study results for the quality evaluation.

method	MBF [3]	CVS [25]	ours
score	1.07	1.22	1.32

training, and gradients were calculated for them.

The top-5 MAP are shown in Table 3. First, our method performed better than the other methods, when considering only the representativeness from a single group in most of the **target groups**, and demonstrated competitive performance in the other. It implies that discriminative information is essentially the key to estimating the *viewpoint*.

Secondly, the performance of our methods with feature learning is better than that without it as a whole. We found it works well even though we exploited a large network with many parameters and the number of samples was relatively small in many cases, except in a few categories. When considering “riding bike on beach (RB)” or “boarding on a snow mountain (BS)”, we noticed a drop in the performance. Our feature learning algorithm works in a kind of self-supervised manner. It trains the feature extractor to better explain the current summary, and therefore, it is dependent on the initial summary selection. If outliers have a high importance score in that step, no matter whether it is discriminative, the parameter update may be strongly affected by such outliers, which likely causes a performance drop.

Thirdly, we found the performance of WSVS and WSVS (large) were worse than our method and even than CSV, which uses only one group. We assume that the reason is that it fails to train the classification model. In this method, the classification model is trained clip-by-clip by assigning the same label to all video clips. It implicitly assumes all clips can be classified into the same group, which is unrealistic when using generic videos such as ones on YouTube as stated in Section 2. If there are some clips that are difficult or impossible to classify, it suffers from over-fitting caused by attempting to correctly classify them. In our case, we assume there are multiple possible ways to divide videos into groups given the same set of videos, as stated earlier. Therefore, parameters cannot be appropriately learned because some clips in videos belonging to different groups can appear to be similar. Given that our method considers

Table 5: User study results for topic selection task. The accuracy takes the value in the range [0, 1].

method	MBF [3]	CVS [25]	ours
accuracy	0.47	0.60	0.76

the discrimination of the generated summary, not all clips, it works better even when using CNN with large parameters.

5.6. User study

Because video summarization is relatively subjective task, we also evaluated the performance with user study. We asked crowd-workers to assign quality score to summaries generated from MBF, CVS, and proposed method. They chose the score from -2 (bad) to 2 (good), and for each video and concept, 10 workers were assigned. The mean results are shown in Table 4. It indicates that the quality of summaries of our method is the best among three methods.

5.7. Visualizing the reason of group division

One possible application of this method is visualizing the reasons driving group divisions. Given multiple groups of videos, why they are grouped in such way is generally unknown. Our algorithm works to visualize an underlying visual concept that is a criterion of division. To determine how well our algorithm has ability to solve this problem, we performed a qualitative evaluation using AMT. We asked crowd-workers to select the topic out of either **concept1** or **concept2** for summaries created in the group setting1 and setting2. We evaluated the performance of how well workers can answer questions about a topic correctly. We set the ground-truth topic as **concept1** when setting1 is used and **concept2** for setting2. We assigned 10 workers for each summary and each group setting. We reported the score in the Table 5. As shown in this table, our method performed better than other methods, which indicates the ability to explain the reasons behind grouping.

6. Conclusion

In this study, we introduced a *viewpoint* for video summarization motivated by the claim that multiple optimal

summaries should exist for one video. We developed a general video summarization method that aims to estimate underlying *viewpoint* by considering video-level *similarity* which is assumed to be derived from corresponding *viewpoint*. For the evaluation, we developed a dataset and we demonstrated the effectiveness of proposed method by performing the qualitative and quantitative experiments on it.

References

- [1] F. Chen and C. De Vleeschouwer. Formulating team-sport video summarization as a resource allocation problem. *TCSVT*, 21(2):193–205, 2011. 2
- [2] F. Chollet et al. Keras. <https://github.com/fchollet/keras>, 2015. 7
- [3] W.-S. Chu, Y. Song, and A. Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 2015. 3, 6, 7, 8
- [4] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *CVPR*, 2012. 7, 8
- [5] M. Fleischman, B. Roy, and D. Roy. Temporal feature induction for baseball highlight classification. In *ACMMM*, 2007. 2
- [6] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014. 2
- [7] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In *Recent Advances in Learning and Control*. 2008. http://stanford.edu/~boyd/graph_dcp.html. 5
- [8] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>, Mar. 2014. 5
- [9] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *ECCV*, 2014. 2
- [10] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 2015. 2
- [11] R. Hong, J. Tang, H.-K. Tan, S. Yan, C. Ngo, and T.-S. Chua. Event driven summarization for web videos. In *SIGMM workshop*, 2009. 2
- [12] 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. 6, 7
- [13] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013. 2
- [14] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014. 2
- [15] A. Kulesza, B. Taskar, et al. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, 2012. 2
- [16] G. R. Lanckriet and B. K. Sriperumbudur. On the convergence of the concave-convex procedure. In *NIPS*, 2009. 5
- [17] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2
- [18] D. Liu, G. Hua, and T. Chen. A hierarchical visual model for video object summarization. *TPAMI*, 32(12):2178–2190, 2010. 2
- [19] T. Liu and J. R. Kender. Optimization algorithms for the selection of key frame sequences of variable length. In *ECCV*, 2002. 2
- [20] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013. 2
- [21] Y.-F. Ma, L. Lu, H.-J. Zhang, and M. Li. A user attention model for video summarization. In *ACMMM*, 2002. 2
- [22] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, 2017. 2
- [23] C.-W. Ngo, Y.-F. Ma, and H.-J. Zhang. Automatic video summarization by graph modeling. In *ICCV*, 2003. 2
- [24] R. Panda, A. Das, Z. Wu, J. Ernst, and A. K. Roy-Chowdhury. Weakly supervised summarization of web videos. In *ICCV*, 2017. 3, 7, 8
- [25] R. Panda and A. K. Roy-Chowdhury. Collaborative summarization of topic-related videos. In *CVPR*, 2017. 3, 6, 7, 8
- [26] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *ECCV*, 2014. 2
- [27] A. Sharghi, B. Gong, and M. Shah. Query-focused extractive video summarization. In *ECCV*, 2016. 2
- [28] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015. 2, 5
- [29] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 7
- [30] M. Sun, A. Farhadi, and S. Seitz. Ranking domain-specific highlights by analyzing edited videos. In *ECCV*, 2014. 2
- [31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 5, 6, 7
- [32] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). In *NIPS*, 2002. 4
- [33] A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural computation*, 15(4):915–936, 2003. 4
- [34] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *CVPR*, 2016. 2
- [35] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *ACMMM*, 2007. 2