



Cluster-Based Video Summarization with Temporal Context Awareness

Hai-Dang Huynh-Lam^{1,2}, Ngoc-Phuong Ho-Thi^{1,2(✉)}, Minh-Triet Tran^{1,2} ,
and Trung-Nghia Le^{1,2} 

¹ University of Science, VNU-HCM, Vietnam
htnphuong19@apcs.fitus.edu.vn

² Vietnam National University, Ho Chi Minh, Vietnam

Abstract. In this paper, we present TAC-SUM, a novel and efficient training-free approach for video summarization that addresses the limitations of existing cluster-based models by incorporating temporal context. Our method partitions the input video into temporally consecutive segments with clustering information, enabling the injection of temporal awareness into the clustering process, setting it apart from prior cluster-based summarization methods. The resulting temporal-aware clusters are then utilized to compute the final summary, using simple rules for keyframe selection and frame importance scoring. Experimental results on the SumMe dataset demonstrate the effectiveness of our proposed approach, outperforming existing unsupervised methods and achieving comparable performance to state-of-the-art supervised summarization techniques. Our source code is available for reference at <https://github.com/hcmus-thesis-gulu/TAC-SUM>.

Keywords: video summarization · clustering · unsupervised learning

1 Introduction

Video summarization is a crucial research area that aims to generate concise and informative summaries of videos, capturing their temporal and semantic aspects while preserving essential content. This task poses several challenges, including identifying important frames or shots, detecting significant events, and maintaining overall coherence. Video summarization finds applications in diverse fields, enhancing video browsing, retrieval, and user experience [4].

The current state-of-the-art methods in summarizing videos are SMN [24] and PGL-SUM [5]. SMN stacks LSTM and memory layers hierarchically to capture long-term temporal context and estimate frame importance based on this information. Its training, however, relies on LSTMs and is not fully parallelizable. PGL-SUM uses self-attention mechanisms to estimate the importance and

H.-D. Huynh-Lam and N.-P. Ho-Thi—Both authors contributed equally to this research.

dependencies of video frames. It combines global and local multi-head attention with positional encoding to create concise and representative video summaries. Both SMN and PGL-SUM heavily rely on human-generated summaries as ground truth, introducing biases and inconsistencies during training.

To eliminate the need for labeled data required by supervised approaches, unsupervised algorithms have been explored, such as Generative Adversarial Networks [3] and Reinforcement Learning [30]. While achieving remarkable results without annotations, their performance gains have been minor compared to supervised methods, and the computational requirements can be high with GPU usage.

A line of research focusing on the use of clustering algorithms for video summarization has been pioneered by De *et al.* [9] and followed by Mahmoud *et al.* [18] to create interpretable summaries without labels and training. Such methods demonstrate acceptable performance in low-resource environments, but their effectiveness has yet to be competitive with learnable approaches.

In this paper, we propose a training-free approach called Temporal-Aware Cluster-based SUMmarization (TAC-SUM) to address the challenges encountered by previous studies. This method leverages temporal relations between frames inside a video to convert clusters of frames into temporally aware segments. Specifically, frame similarities available from these clusters are used to divide the video into non-overlapping and consecutive segments. The proposed algorithm then applies simple and naive rules to select keyframes from these segments as well as assign importance scores to each frame based on its segment’s information. Our approach is expected to outperform existing cluster-based methods by injecting temporal awareness after the clustering step. It eliminates the need for expensive annotation, increases efficiency, and offers high interpretability due to its visualizability and transparent rules. An important distinction from some previous unsupervised studies is that TAC-SUM currently relies on naive rules, leaving ample room for future improvement, including the integration of learnable components, which have been successful in learning-free algorithms [18].

We conduct quantitative and qualitative experiments on the SumMe dataset [11] to evaluate our method’s performance in video summarization. The quantitative experiment shows that our approach significantly outperforms existing unsupervised methods and is comparable to current state-of-the-art supervised algorithms. The qualitative study demonstrates that our approach produces effective visual summaries and exhibits high interpretability with the use of naive rules.

The main contributions presented in the paper are as follows:

- We introduce the integration of temporal context into the clustering mechanism for video summarization, addressing the shortcomings of traditional cluster-based methods.
- We propose a novel architecture that effectively embeds temporal context into the clustering step, leading to improved video summarization results.
- Our approach demonstrates superior performance compared to existing cluster-based methods and remains competitive with state-of-the-art deep learning summarization approaches.

2 Related Work

Video summarization techniques can be broadly classified into two categories: supervised methods and unsupervised methods. While supervised methods demonstrate superior performance in domain-specific applications, they rely heavily on labeled data, making them less practical for general video summarization tasks where labeled data may be scarce or costly to obtain. As a result, unsupervised methods remain popular for their versatility and ability to generate summaries without the need for labeled data. Within unsupervised approaches, clustering algorithms have emerged as a popular choice.

Cluster-based video summarization methods utilize the concept of grouping similar frames or shots into clusters and selecting representative keyframes from each cluster to form the final summary. These approaches have shown promise in generating meaningful summaries, as they can capture content diversity and reduce redundancy effectively. Prior works have explored various clustering techniques for video summarization. Mundu *et al.* [20] employed Delaunay triangulation clustering using color feature space, but high computational overhead limited its practicality. De *et al.* [9] utilized K-means clustering with hue histogram representation for keyframe extraction. Shroff *et al.* [23] introduced a modified version of K-means that considers inter-cluster center variance and intra-cluster distance for improved representativeness and diversity. Asadi *et al.* [6] applied fuzzy C-means clustering with color component histograms. Mahmoud *et al.* [19] used DBSCAN clustering with Bhattacharya distance as a similarity metric within the VSCAN algorithm. Cluster-based methods offer simplicity and interpretability, often relying on distance metrics like Euclidean or cosine similarity to group similar frames. Their computational efficiency allows for scalability to large video datasets. However, traditional cluster-based approaches have limitations. Notably, they may overlook temporal coherence, leading to fragmented and incoherent summaries. Additionally, handling complex video content with multiple events or dynamic scenes can pose challenges, as these methods primarily rely on visual similarity for clustering.

With the rise of deep learning, video summarization has seen significant advancements. In supervised approaches, temporal coherence is addressed by modeling variable-range temporal dependencies among frames and learning their importance based on ground-truth annotations. This has been achieved using various architectures, such as LSTM-based key-frame selectors [25, 27–29], Fully Convolutional Sequence Networks [22], and attention-based architectures [10, 15, 16].

However, achieving temporal coherence in unsupervised learning poses challenges. One promising direction is the utilization of Generative Adversarial Networks (GANs). Mahasseni *et al.* [17] combined an LSTM-based key-frame selector, a Variational Auto-Encoder (VAE), and a trainable Discriminator in an adversarial learning framework to reconstruct the original video from the summary. Other works extended this core VAE-GAN architecture with tailored attention mechanisms to capture frame dependencies at various temporal granularities during keyframe selection [12–14]. These methods focus on important

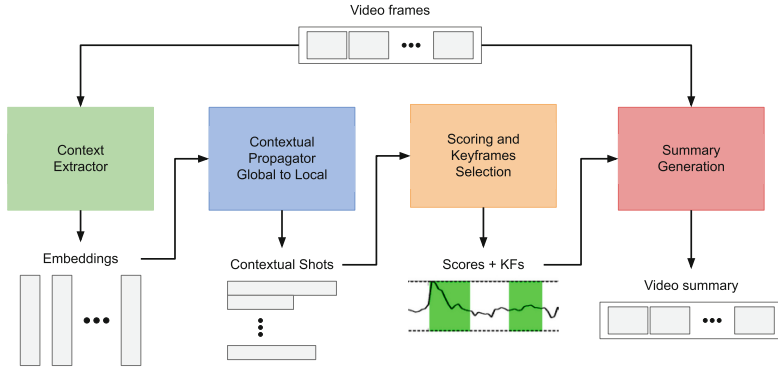


Fig. 1. Pipeline of the proposed approach showcasing four modules and information flow across main stages.

temporal regions and model long-range dependencies in the video sequence. Although GAN-based models have shown promise in generating coherent summaries, they face challenges of unstable training and limited evaluation criteria.

The proposed method leverages cluster-based models by utilizing visual representations generated by unsupervised deep learning approaches such as DINO [7]. Addressing the problem of temporal coherence, our developed TAC-SUM introduces the temporal context into the process. This integration of temporal context enhances the summarization performance, as demonstrated by experimental results.

3 Proposed Approach

Our approach selects an ordered subset $\mathbf{S} = \{I_{t_1}, I_{t_2}, \dots, I_{t_L}\}$ of L frames from a video $\mathbf{I} = \{I_1, I_2, \dots, I_T\}$, where T is the total number of frames and the summarized subset \mathbf{S} is obtained by selecting frames indexed at t_i positions. The timestamp vector \mathbf{t} comprises such positions $\{t_1, t_2, \dots, t_L\}$. In Fig. 1, we illustrate the four stages of our method as distinct modules. Each stage comprises several steps tailored to the specific role and algorithm implemented. We provide a detailed explanation of each stage in the remaining text of this section. In addition, Sect. 3.4 is dedicated to clarifying several technical details related to the implementation of our approach.

3.1 Generating Contextual Embeddings

This stage extracts the context of an input video \mathbf{I} from its frames I_t . It involves two steps: sampling the video and constructing embeddings for each sampled frame.

Sampling Step. To reduce computational complexity, we employ a sampling technique to extract frames from \mathbf{I} into a sequence of samples $\hat{\mathbf{I}}$. The frame rate

of $\hat{\mathbf{I}}$ is matched to a pre-specified frame rate R . This method ensures representativeness and serves as normalization for different inputs. The sampling process involves dividing the original frames within a one-second period into equal-length snippets and selecting the middle frame of each snippet as the final sample.

Embedding Step. For each sampled frame \hat{I}_i , we utilize a pre-trained model to extract its visual embedding \mathbf{e}_i . The pre-trained model is denoted as a function $g : \mathbb{R}^{W \times H \times C} \rightarrow \mathbb{R}^D$ that converts \hat{I}_i into an embedding vector of size D . All embeddings are concatenated to form the contextual embedding of the sampled video $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{\hat{T}}\}$. Figure 2 gives two examples of contextual embeddings.

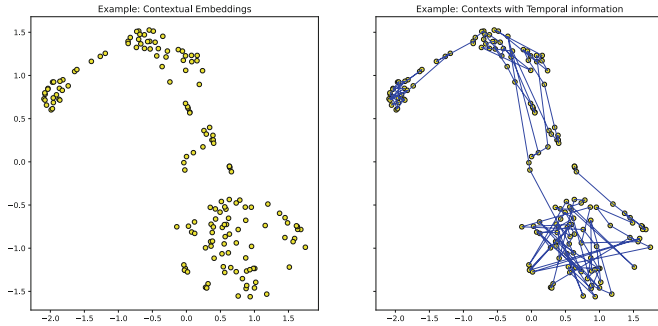


Fig. 2. Visual illustration of contextual information.

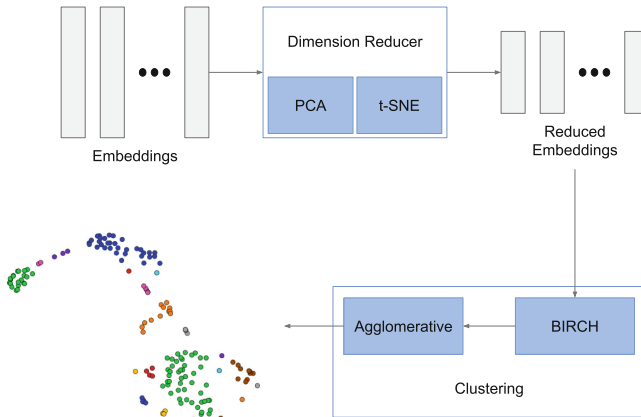


Fig. 3. Overall pipeline for the Contextual Clustering step.

3.2 From Global Context to Local Semantics

This stage distills global information from the contextual embedding \mathbf{E} into finer, local levels. Our method comprises two steps: using traditional clustering

to propagate contextual information into partition-level clusters, and further distilling partition-level information into sample-level.

Contextual Clustering. Clustering the contextual embeddings \mathbf{E} captures global and local relationships between visual elements in the video. We first reduce the dimension of \mathbf{E} to a reduced embedding $\hat{\mathbf{E}}$. A coarse-to-fine clustering approach is then applied to divide the sampled frames into K clusters, creating a label vector $\mathbf{c} \in \mathbf{N}^{\hat{T}}$. More details can be found in Fig. 3. Starting with the contextual embedding $\mathbf{E} \in \mathbb{R}^{\hat{T} \times D}$, a reduced embedding $\hat{\mathbf{E}} \in \mathbb{R}^{\hat{T} \times \hat{D}}$ is computed using PCA and t-SNE. A traditional clustering method called BIRCH algorithm [26] is applied to compute coarse clusters of sampled frames, creating a sample-level notation for coarse clusters $\hat{\mathbf{c}} = \{\hat{c}_1, \hat{c}_2, \dots, \hat{c}_{\hat{T}}\}$. Then, a hierarchical clustering algorithm is employed to combine coarse clusters into finer clusters with the number of eventual clusters is pre-determined based on a sigmoidal function and a maximum threshold. The fine cluster is formed as the union of at least one coarse cluster. Clusters are progressively merged based on affinity between them. This approach achieves a hierarchical clustering that effectively propagates information from the global level $\hat{\mathbf{E}}$ to the local level \mathbf{c} , enabling us to extract semantically meaningful clusters.

Semantic Partitioning. Following the contextual clustering step, each sampled frame \hat{I}_i is assigned a label c_i corresponding to its cluster index. An outlier elimination removes possible outliers and a refinement step consolidates smaller partitions into larger ones with a threshold ϵ . A smoothing operation is applied to labels by assigning the final label \hat{c}_i of each frame by taking a majority vote among its consecutive neighboring frames. Once frames have been assigned their final labels \mathcal{C} , they are partitioned into sections \mathcal{P} based on these labels. The semantic partitioning $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_{\hat{N}}\}$ obtained from the above process contains \hat{N} sections which are then progressively refined with length condition. Algorithm for this refinement is delineated as follows with a parameter ϵ denoting the minimum partition’s length allowed in the result. Initially, the number

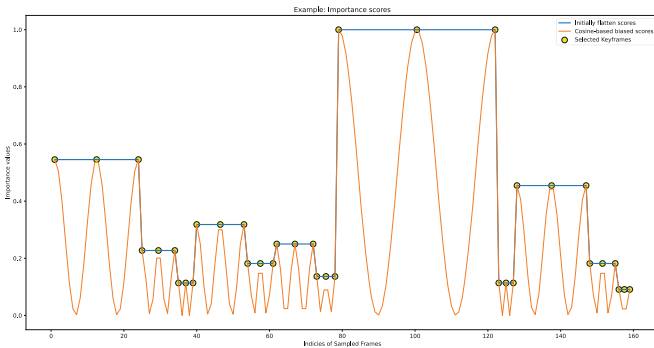


Fig. 4. Comparison between cosine-interpolated scores and flat scores are demonstrated for two examples.

of partitions N is set to \hat{N} . Subsequently, while the minimum length of the partitions is less than ϵ , the index of the shortest partition \hat{i} is determined. The left and right sides of partition \hat{i} are merged with their respective neighboring partitions and their lengths are updated accordingly. The indexes of \mathcal{P} are then updated and the number of partitions N is reduced by 1. This process continues until all partitions have a length of at least ϵ . This partitioning result allows us to focus on individual semantic parts within the video and analyze their characteristics independently, enabling more detailed analysis and summary generation in subsequent stages.

3.3 Keyframes and Importance Scores

After the partitioning step, the resulted partitions \mathcal{P} are used to generate keyframes \mathbf{k} which carry important information of the original input. An importance score v_i is calculated for every sampled frame \hat{I}_i .

Keyframes Selection. The set of keyframes \mathbf{k} is a subset of the indexes of sampled frames $\mathbf{k} \subset \mathbf{t}$, and is a union of partition-wise keyframes $\mathbf{k}^{(i)}$, that is $\mathbf{k} = \bigcup_{i=1}^N \mathbf{k}^{(i)}$. There are three options for extracting the partition-wise keyframes $\mathbf{k}^{(i)}$ from its associated partition \mathcal{P}_i which are respectively *Mean*, *Middle*, and *Ends*. These options can be further combined into more advanced settings such as the rule *Middle + Ends* demonstrated in Fig. 4.

Importance Scores. The individual importance scores v_i of all sampled frames \hat{I}_i form a vector of importances $\mathbf{v} \in \mathbb{R}^{\hat{T}}$. We initialize the importance score $\hat{\mathbf{v}}$ to be the length of the section it belongs to. The final importance score of each sample v_i is computed by scaling the initialized value \hat{v}_i using a keyframe-biasing method. Several biasing options are given to either increase the importance of frames closer to keyframes or decrease the scores of others. Different interpolating methods are used to fill the importance scores of samples between key positions. Two options for interpolation are *cosine* and *linear*. An example illustrating the difference between cosine-interpolated importances and flat scores is given in Fig. 4. By determining the importance scores of frames, we can prioritize and select the most significant frames for inclusion in the video summary.

3.4 Implementation Details

Before the feature extraction step, the video is sampled with a target frame rate of $R = 4$ frames per second. We experiment with 2 pre-trained models to generate embeddings for each frame: DINO [7] and CLIP [21]. The input frame is processed using the pre-trained image processor associated with the pre-trained model. The output is an image, which is fed into the pre-trained model to obtain embeddings. For DINO, we select the first vector (cls token) in its output embedding as the semantic embedding of the sample. We concatenate the vector from all frames in to obtain the contextual embedding.

For dimension reduction, we utilize models from `scikit-learn`, including PCA and t-SNE. The number of clusters K in contextual clustering is then computed by the equation provided in Sect. 3.2. In the semantic partition step, we set the window size W for mode convolution to 5, and the minimum length ϵ for each segment to 4. For keyframe selection, we employ the setting *Middle + End*. In the importance scoring step, we use `cosine` interpolation and set keyframe biasing scheme to *Increase the importances of keyframes* with $B = 0.5$.

4 Experiments

4.1 Dataset

For evaluating the performance of our TAC-SUM model, we utilize the SumMe dataset [11]. This benchmarking dataset consists of 25 videos ranging from 1 to 6 min in duration, covering various events captured from both first-person and third-person perspectives. Each video is annotated with multiple (15–18) key-fragments representing important segments. Additionally, a ground-truth summary in the form of frame-level importance scores (computed by averaging the key-fragment user summaries per frame) is provided for each video to support supervised training.

4.2 Evaluation Measures

The summary selected by our summarizer is then compared with those generated by humans to determine its correctness, in other words, whether that summary is good or not depends on its similarity with regard to the annotated ones. A widely established metric for this comparison is f-measure, which is adopted in prior works [1, 5, 8]. This metric requires an automatic summarizer to generate a proxy summary $\hat{\mathcal{S}}$ from pre-computed consecutive segmentations \mathcal{S} associated with each video in the dataset. The f-measure metric is computed as f1-score between the segments chosen by automatic method against ground-truth selected by human evaluators. Previous studies [5, 25] have formulated the conversion from importances to choice of segments as a Knapsack problem so that a simple dynamic programming method can be implemented to recover the proxy summary from outputted scores. The formulation includes lengths of segments as weighting condition while individual segment’s value is computed using importance scores. More detailed information can be found in prior research [25].

4.3 Comparison with State-of-the-art Methods

The performance of our proposed TAC-SUM approach is compared with various summarization methods from the literature in Table 1. These referenced approaches include both supervised and unsupervised algorithms that have been previously published, and the evaluation metric used is established under Sect. 4.2. As a general baseline, we include a random summarizer, which assigns

Table 1. Comparison of performance in f-measure (%) among previous approaches and our method together with rankings on unsupervised only as well as in general.

Methods		F-Score	Rank (Unsupervised)	Rank (General)
Random summary		40.2	7	13
Supervised	SMN [24]	58.3	-	1
	VASNet [10]	49.7	-	10
	PGL-SUM [5]	57.1	-	2
	H-MAN [16]	51.8	-	5
	SUM-GDA [15]	52.8	-	4
	SUM-DeepLab [22]	48.8	-	12
Unsupervised	CSNet [13]	51.3	2	6
	AC-SUM-GAN [1]	50.8	3	7
	CSNet+GL+RPE [14]	50.2	4	8
	SUM-GAN-AAE [3]	48.9	6	11
	SUM-GDA _{unsup} [15]	50.0	5	9
	TAC-SUM (ours)	54.48	1	3

importance scores to each frame based on a uniform distribution. The final performance is averaged over 100 sampling runs for each video [2].

The results in Table 1 highlight the effectiveness of our training-free approach, which achieves remarkable performance without any learning aspect. It outperforms existing unsupervised models by at least 3.18%, demonstrating its ability to generate high-quality summaries. Moreover, our model ranks third when compared to state-of-the-art supervised methods, showing competitive performance and even surpassing several existing approaches.

It is worth noting that the SMN method has been evaluated using only one randomly created split of the used data [24]. Apostolidis *et al.* [2] suggest that these random data splits show significantly varying levels of difficulty that affect the evaluation outcomes.

We acknowledge that the pre-trained models used in our architecture were originally trained on general image datasets, which may not perfectly align with the distribution of the specific dataset used in this evaluation. Despite this potential distribution mismatch, our proposed method exhibits strong performance on the evaluated dataset, showcasing the generalizability and adaptability of this training-free framework.

4.4 Ablation Study

To assess the contribution of each core component in our model, we conduct an ablation study, evaluating the following variants of the proposed architecture: variant **TAC-SUM w/o TC**: which is not aware of temporal context by skipping the semantic partitioning stage (Sect. 3.2), and the full algorithm

Table 2. Ablation study based on the performance (F-Score(%)) of two variants of the proposed approach on SumMe

Settings	F-Score
TAC-SUM w/o TC	46.00
TAC-SUM (ours)	54.48

Table 3. Comparison of performance (F-Score(%)) with different embedding pre-trained models

Setting		F-Score
Embedding Model	Best Config	
dino-b16	Euclidean PCA (34) + t-SNE (2)	54.48
clip-base-16	Euclidean PCA (44) + t-SNE (3)	52.33

TAC-SUM (ours). The results presented in Table 2 demonstrate that removing the temporal context significantly impacts the summarization performance, thus confirming the effectiveness of our proposed techniques. The inclusion of temporal context enhances the quality of the generated summaries, supporting the superiority of our proposed TAC-SUM model.

As mentioned in Sect. 3.4, we conducted experiments using different pre-trained models for visual embedding extraction. Table 3 compares the result of the framework using different pre-trained models: dino-b16 and clip-base-16. Both models are base models with a patch size of 16. The “Best Config” column shows the configuration that achieved the best result, including the distance used in the clustering step (Euclidean), the algorithms used for embedding size reduction (PCA and t-SNE), and the dimension of the reduced embeddings represented by the number next to the reducer. The results presented in Table 3 demonstrate that our proposed framework performs relatively well with various pre-trained models, showcasing its flexibility and efficiency. The ability to work effectively with different pre-trained models indicates that our approach can leverage a wide range of visual embeddings, making it adaptable to various video summarization scenarios. This flexibility allows practitioners to choose the most suitable pre-trained model based on their specific requirements and available resources.

4.5 Qualitative Assessment

To evaluate the interpretability of the proposed approach, we compared the automatically generated importance scores with those assigned by human annotators. Figure 5 displays the importance scores obtained through averaging human annotations as well as the scores generated by the proposed method. The flat result shows that each computed partition may be associated with one or several peaks in the user summaries, located at different positions within the partition.

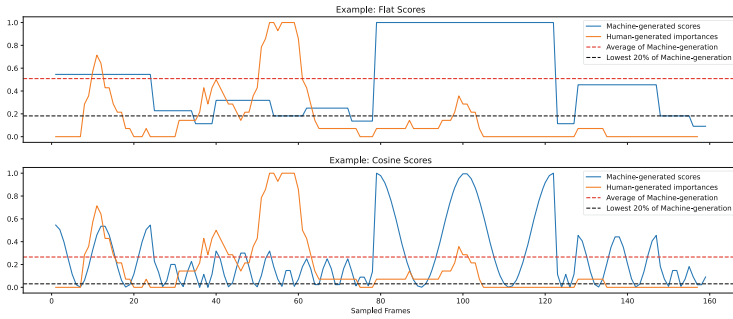


Fig. 5. Comparison of importance scores between user-annotated scores and scores generated by the proposed method under the unbiased **flat** rule as well as the biased **cosine** rule.

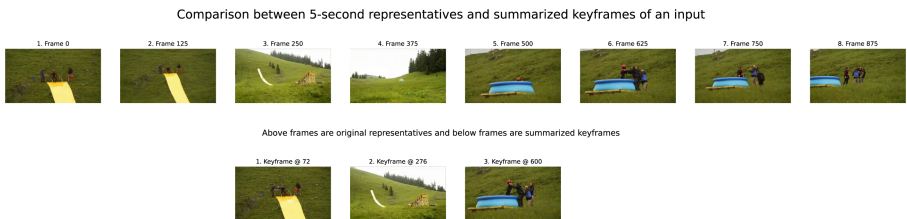


Fig. 6. Comparison between the representatives sampled from the original video with its summarization as a set of keyframes.

Longer partitions, which have higher flat scores according to the definition, tend to provide a more stable estimation of users’ peaks. The experimental result also provides insights into the keyframe-biasing method employed in the proposed method, wherein higher importance is assigned to frames that are closer to keyframes. This figure reveals that the majority of the peaks in the **cosine** scores align with the peaks of the annotated importance. However, there are some peaks in the users’ scores that are not captured by the **cosine** interpolation.

A visual inspection of our method’s summarization results is conducted in which a reference video is analyzed against its summary generated through the approach. We present the inspection’s result in Fig. 6 with the original frames of the reference video and selected keyframes. The original frames are sampled every 5s from the video, which shows a man playing a game of sliding down a slope and jumping into a pool of water. The keyframes are selected based on their importance scores, which are higher than the average on the video level. Our method preserves the main content and events of the video and selects diverse and representative keyframes that show different aspects of the video. Our method generates informative and expressive keyframes that convey the main theme, message, or story of the video.

4.6 Limitations

While the proposed method offers several advantages which have been already illustrated in the experimental results, it also has certain limitations that should be acknowledged. **Naive rules** for scoring and selection of keyframes are being used. Therefore, our current approach may not always accurately predict frame importance. Incorporating more sophisticated scoring mechanisms can enhance the summarization process. **Limited learnability** is demonstrated by our method as it lacks the ability to improve in a data-driven way due to its current reliance on predefined rules. Future research could explore integrating data-driven approaches like machine learning algorithms or attention mechanisms to enhance adaptability.

5 Conclusion

In this paper, we introduced TAC-SUM, an unsupervised video summarization approach that incorporates temporal context for generating concise and coherent summaries. The contextual clustering algorithm has successfully partitioned frames into meaningful segments, ensuring temporal coherence. Experimental results show that our method significantly outperforms traditional cluster-based approaches and even is competitive with state-of-the-art supervised methods on the SumMe dataset.

Despite its success, TAC-SUM has limitations related to pre-trained models and data-driven improvement. To address these limitations, future work will focus on integrating learnable components into the model to enhance the summarization process and improve adaptability to various video domains. This includes replacing the current algorithm for contextual clustering with a deep neural network having trainable parameters, enabling the model to capture more complex patterns and adapt to diverse video datasets. Additionally, various architectures and training techniques will be explored to transform the naive rules of importance into a data-driven scoring process, allowing complicated scores to be predicted.

Acknowledgement. This research is supported by research funding from Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.

References

1. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: AC-SUM-GAN: connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Trans. Circuits Syst. Video Technol.* **31**(8), 3278–3292 (2020)
2. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: Performance over random: a robust evaluation protocol for video summarization methods. In: *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1056–1064 (2020)

3. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: Unsupervised video summarization via attention-driven adversarial learning. In: Ro, Y.M., et al. (eds.) MMM 2020. LNCS, vol. 11961, pp. 492–504. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-37731-1_40
4. Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., Patras, I.: Video summarization using deep neural networks: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021)
5. Apostolidis, E., Balaouras, G., Mezaris, V., Patras, I.: Combining global and local attention with positional encoding for video summarization. In: 2021 IEEE International Symposium on Multimedia (ISM), pp. 226–234 (2021)
6. Asadi, E., Charkari, N.M.: Video summarization using fuzzy C-means clustering. In: 20th Iranian Conference on Electrical Engineering (ICEE2012), pp. 690–694. IEEE (2012)
7. Caron, M., et al.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
8. Chu, W.T., Liu, Y.H.: Spatiotemporal modeling and label distribution learning for video summarization. In: 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6. IEEE (2019)
9. De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., de Albuquerque Araújo, A.: VSUMM: a mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recogn. Lett.* **32**(1), 56–68 (2011)
10. Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., Remagnino, P.: Summarizing videos with attention. In: Carneiro, G., You, S. (eds.) ACCV 2018. LNCS, vol. 11367, pp. 39–54. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21074-8_4
11. Gygli, M., Grabner, H., Riemenschneider, H., Van Gool, L.: Creating summaries from user videos. In: ECCV (2014)
12. He, X., et al.: Unsupervised video summarization with attentive conditional generative adversarial networks. In: Proceedings of the 27th ACM International Conference on multimedia, pp. 2296–2304 (2019)
13. Jung, Y., Cho, D., Kim, D., Woo, S., Kweon, I.S.: Discriminative feature learning for unsupervised video summarization. In: Proceedings of the AAAI Conference on artificial intelligence, vol. 33, pp. 8537–8544 (2019)
14. Jung, Y., Cho, D., Woo, S., Kweon, I.S.: Global-and-Local relative position embedding for unsupervised video summarization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 167–183. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_11
15. Li, P., Ye, Q., Zhang, L., Yuan, L., Xu, X., Shao, L.: Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recogn.* **111**, 107677 (2021)
16. Liu, Y.T., Li, Y.J., Yang, F.E., Chen, S.F., Wang, Y.C.F.: Learning hierarchical self-attention for video summarization. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 3377–3381. IEEE (2019)
17. Mahasseni, B., Lam, M., Todorovic, S.: Unsupervised video summarization with adversarial LSTM networks. In: CVPR (2017)
18. Mahmoud, K.M., Ghanem, N.M., Ismail, M.A.: Unsupervised video summarization via dynamic modeling-based hierarchical clustering. In: Proceedings of the 12th International Conference on Machine Learning and Applications, vol. 2, pp. 303–308 (2013)

19. Mahmoud, K.M., Ismail, M.A., Ghanem, N.M.: VSCAN: an enhanced video summarization using density-based spatial clustering. In: Petrosino, A. (ed.) ICIAP 2013. LNCS, vol. 8156, pp. 733–742. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41181-6_74
20. Mundur, P., Rao, Y., Yesha, Y.: Keyframe-based video summarization using Delaunay clustering. *Int. J. Digit. Libr.* **6**, 219–232 (2006)
21. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR (2021)
22. Rochan, M., Ye, L., Wang, Y.: Video summarization using fully convolutional sequence networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 347–363 (2018)
23. Shroff, N., Turaga, P., Chellappa, R.: Video précis: highlighting diverse aspects of videos. *IEEE Trans. Multimedia* **12**(8), 853–868 (2010)
24. Wang, J., Wang, W., Wang, Z., Wang, L., Feng, D., Tan, T.: Stacked memory network for video summarization. In: Proceedings of the 27th ACM International Conference on Multimedia, pp. 836–844 (2019)
25. Zhang, K., Chao, W.-L., Sha, F., Grauman, K.: Video summarization with long short-term memory. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 766–782. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_47
26. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. *SIGMOD Rec.* **25**(2), 103–114 (1996). <https://doi.org/10.1145/235968.233324>
27. Zhao, B., Li, X., Lu, X.: Hierarchical recurrent neural network for video summarization. In: Proceedings of the 25th ACM International Conference on Multimedia, pp. 863–871 (2017)
28. Zhao, B., Li, X., Lu, X.: HSA-RNN: hierarchical structure-adaptive RNN for video summarization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7405–7414 (2018)
29. Zhao, B., Li, X., Lu, X.: TTH-RNN: tensor-train hierarchical recurrent neural network for video summarization. *IEEE Trans. Industr. Electron.* **68**(4), 3629–3637 (2020)
30. Zhou, K., Qiao, Y., Xiang, T.: Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In: AAAI (2018)