Revisiting Kernel Temporal Segmentation as an Adaptive Tokenizer for Long-form Video Understanding

Mohamed Afham Satya Narayan Shukla Omid Poursaeed Pengchuan Zhang
Ashish Shah Sernam Lim
Meta AI

Abstract

While most modern video understanding models operate on short-range clips, real-world videos are often several minutes long with semantically-consistent segments of variable length. A common approach to process long videos is applying a short-form video model over uniformly sampled clips of fixed temporal length and aggregating the outputs. This approach neglects the underlying nature of long videos since fixed-length clips are often redundant or uninformative. In this paper, we aim to provide a generic and adaptive sampling approach for long-form videos in lieu of the de facto uniform sampling. Viewing videos as semantically-consistent segments, we formulate a task-agnostic, unsupervised, and scalable approach based on Kernel Temporal Segmentation (KTS) for sampling and tokenizing long videos. We evaluate our method on long-form video understanding tasks such as video classification and temporal action localization, showing consistent gains over existing approaches and achieving state-of-the-art performance on long-form video modeling.

1. Introduction

The majority of video understanding models are devised to learn representations of short-form videos ranging from 5 to 10 seconds [6, 28, 14, 34, 5, 2, 17, 20]. These models usually suffer from computation and memory bottlenecks when processing videos of longer lengths. A common approach to overcome this bottleneck is to uniformly divide long videos into fixed-length clips, process each clip separately and aggregate the results. This approach is highly redundant as nearby clips often convey similar information and short clips that overlap semantically meaningful segments are often uninformative.

Several works [22, 18, 32, 8, 15] have previously investigated adaptive sampling to learn video representations in an efficient manner. These methods often devise a learnable adaptive sampler to select more representative frames of the video based on the reward or penalty provided by the final prediction score. However, these methods are often limited to the classification task and are heavily dependent on the specific tasks and datasets on which they are trained and cannot easily transfer to unseen tasks or datasets. Most of these adaptive sampling approaches are not scalable to sampling a large number of frames which is required for understanding long-form videos. In fact, all the recent approaches [13, 29] for long-form video understanding use the de facto uniform sampling for sampling fixed-length clips from long videos.

In this work, we propose a task-agnostic, adaptive, and unsupervised sampling approach for long videos. Motivated by the intuition that humans perceive videos as semantically-consistent segments of variable length, we decompose the video to semantically meaningful segments using Kernel Temporal Segmentation (KTS) [24]. KTS extracts features from sparsely sampled candidate frames, computes the matrix of frame-to-frame similarity, and outputs a set of optimal change points corresponding to the boundaries of temporal segments. We then sample frames from each segment uniformly which comprises the input to the video understanding model. Our KTS-based input tokenization achieves the following desirable attributes: (a) it is agnostic to the downstream task, (b) it yields semantically-consistent segments without relying on training data, and (c) it is scalable to an arbitrary number of segments and frames for a given long video. We validate the generalizability of KTS-based adaptive sampling on multiple downstream tasks and benchmarks. We evaluate KTS-based sampling for video classification on Breakfast [16] dataset achieving state-of-the-art performance. We also report results for temporal action localization on ActivityNet [4], showing the effectiveness of KTS-based sampling over standard uniform sampling. Furthermore, we provide a comparison with existing adaptive frame sampling methods on ActivityNet video classification and show that our approach outperforms the baselines.

The main contribution of our work can be summarized as follows:

- We propose an adaptive, unsupervised, and task-agnostic frame sampling mechanism for long videos based on Kernel Temporal Segmentation (KTS), which overcomes deficiencies of common sampling approaches.
- We extensively evaluate KTS-based adaptive sampling against existing sampling techniques on video classification and temporal action localization tasks, showing consistent improvements and achieving state-of-the-art performance on long-form video understanding.

2. Method

2.1. Kernel Temporal Segmentation

The initial motivation behind KTS is to detect change points in the input and decompose the video into semantically-consistent segments. KTS is a kernel-based algorithm that operates independently and in an unsupervised manner, hence it does not require any additional training to yield meaningful video segments. KTS has been extensively leveraged by several video summarization approaches [21, 36, 25, 33, 38] as the segmentation output provided by KTS has a significant impact on identifying highlights of the video and yielding a high-quality summarization of the video. Here we briefly describe the KTS algorithm.

Given a long-form video, we initially downsample it, e.g. to one frame per second, and extract frame-level features using a pre-trained feature extractor f_θ. Let $(x_i)_{i=1}^n \in \mathbf{X}$ represent the sampled frames, $\mathbf{K} : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ represent a kernel function (Gram matrix) between descriptors $f_\theta(x_i)$ and $\phi : \mathbf{X} \to \mathcal{H}$ be the associated feature map with norm $\|.\|_{\mathcal{H}}$. Suppose we want to choose $m-1$ change points $x_{t_1}, \cdots, x_{t_{m-1}}$, which correspond to m segments $[x_{t_0}, x_{t_1}], [x_{t_1}, x_{t_2}], \cdots, [x_{t_{m-1}}, x_{t_m}]$ with $x_{t_0} = 0$ and $x_{t_m} = T$ being length of the video.

The KTS algorithm minimizes the sum of the within-segment variances:

$$\min_{m, t_1, \cdots, t_{m-1}} \sum_{i=1}^{m} var(t_{i-1}, t_i) \tag{1}$$

where:

$$var(t_{i-1}, t_i) = \sum\nolimits_{t=t_{i-1}}^{t_i - 1} \|\phi(x_t) - \mu_i\|^2 \tag{2}$$

and μ_i is the within-segment mean:

$$\mu_i = \frac{\sum_{t=t_{i-1}}^{t_i-1} \phi(x_t)}{t_i - t_{i-1}} \tag{3}$$

We can also make KTS adaptive to each video by making the number of segments m variable. To avoid over-segmentation we add a penalty term $g(m, n)$ to the objective function. A common choice for $g(m, n)$ is $m \log(\frac{m}{n} + 1)$. In this case, our final objective is:

$$\min_{m, t_1, \cdots, t_{m-1}} \sum_{i=1}^{m} var(t_{i-1}, t_i) + g(m, n) \tag{4}$$

In order to solve Equation 1 and 4, we first compute the kernel for each pair of descriptors. We use a dot-product kernel in practice. Then the segment variances are computed for each possible starting point and segment duration. Finally, we use dynamic programming to minimize the objective and find the change points. Refer to [24] for more details.

2.2. Adaptive sampling with KTS

KTS algorithm yields a set of change points $x_{t_1}, \cdots, x_{t_{m-1}}$ which decompose the video into m segments. Note that unlike shot boundary detection methods which focus on local differences between consecutive frames, KTS takes into account the differences between all pairs of frames. Therefore it provides semantically-consistent and general segments. To represent each segment we uniformly sample k frames from it. Long-form video models often consist of a backbone to process short-range clips and an aggregation mechanism (e.g. via a transformer or simple averaging). We feed sampled frames from each segment to the clip-level model which learns the representation for each segment/scene. The aggregation mechanism then combines scene-level information to obtain a global video-level representation. This is in line with how humans perceive videos. Despite its simplicity, we show that our sampling approach achieves state-of-the-art performance on long-form video modeling and outperforms existing samplers on several tasks and benchmarks.

3. Experiments

3.1. Datasets

Breakfast [16] is a human activity dataset focused on cooking-oriented actions. It comprises 10 categories of cooking breakfast. It contains 1712 videos in total with 1357 for training and 335 for testing. The average length of a video is 2.3 minutes. ActivityNet [4] dataset contains around 20,000 untrimmed videos spanning 200 action classes of daily activities. The average length of a video is 117 seconds, and the average length of action segments is 48 seconds. Thus it can be considered as a long-form video dataset. We report average $mAP@[0.5 : 0.05 : 0.95]$ similar to Actionformer [35] for a fair comparison.

3.2. Comparison with Existing Adaptive Sampling Methods

Table. 1 shows the comparison of KTS-based adaptive tokenization with existing efficient frame sampling meth-

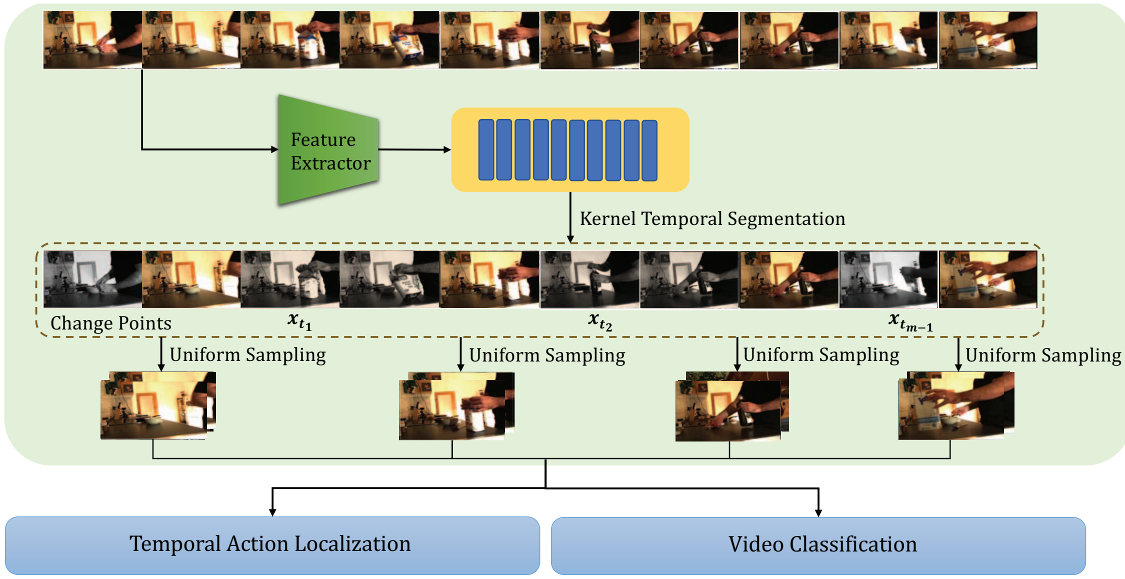

Figure 1: An overview of KTS-based adaptive sampling for Video Classification and Temporal Action Localization. The input video is initially downsampled and $m - 1$ change points are computed using the KTS algorithm. k frames are then uniformly sampled from each of the m segments and are processed for the downstream task.

Table 1: Comparison of our approach with existing adaptive sampling strategies on ActivityNet video classification.

Method	Backbone	mAP (%)	GFLOPs
NSNet [32]	ResNet-101	74.9	73.2
AdaFrame [31]	ResNet-101	71.5	78.7
LiteEval [30]	ResNet-101	72.7	95.1
KTS (Ours) (84×84) [8 frames]	ResNet-101	**80.9**	**67.1**
Uniform	ResNet-50	72.5	65.8
Random	ResNet-50	71.2	65.8
SCSampler [15]	ResNet-50	72.9	41.9
AdaMML [23]	ResNet-50	73.9	94.0
AR-Net [22]	ResNet-50	73.8	33.5
ListenToLook [7]	ResNet-50	72.3	81.4
OCSampler [18]	ResNet-50	79.8	67.2
KTS (Ours) (84×84) [6 frames]	ResNet-50	74.8	**29.7**
KTS (Ours) (84×84) [8 frames]	ResNet-50	**80.0**	32.1
KTS (Ours) (112×112) [8 frames]	ResNet-50	**80.3**	37.4

ods for video classification on the ActivityNet dataset. We use MobileNetv2 [26] pre-trained on ImageNet-1K to extract the features. For a fair comparison with previous methods in terms of efficiency, we initially uniformly sample 16 frames resized to a smaller resolution (e.g., 112×112) in a given video as the change point candidates and estimate change points. We sample one frame within each segment and train the ResNet50 classifier (pre-trained on Imagenet-1K) for video classification on ActivityNet. Our results show that KTS-based sampling yields a competitive performance when compared to existing adaptive sampling approaches. In particular, KTS-based sampling improves the classification accuracy by 1.03% over AR-Net [22] while minimizing the computational cost by 3.8 GFLOPS. KTS algorithm incurs only around 0.004 GFLOPS in our experiments which is comparatively negligible to the computational cost incurred by ResNet50 and MobileNetV2. KTS-based sampling method also outperforms OCSampler [18] while incurring significantly less computation cost.

3.3. Video Classification

Baseline: We adopt the recently introduced ViS4mer [13] as the baseline model to evaluate the performance of KTS-based adaptive sampling against the uniform sampling on video classification tasks. ViS4mer is a long-range video classification model comprised of a standard Transformer encoder [3, 20] and a multi-scale temporal S4 [9] decoder. It extracts features from input video tokens using the Transformer encoder which are then fed to the multi-scale S4 decoder that learns hierarchical spatio-temporal video representations. ViS4mer uses Video Swin Transformer [20] to extract features in experiments on the Breakfast dataset. Despite innovation in the modeling aspect, ViS4mer leverages uniform sampling to tokenize the input video. We adopt KTS-based adaptive sampling in both settings owing to its task-agnostic nature.

Implementation Details: Given a video, we downsample it to one frame per second, and use the downsampled frames as candidates for computing the change points. We use GoogleNet [27] pretrained on ImageNet-1K for extracting the feature descriptors. We sample $m \times k$ frames for each video as described in Sec. 2.2, and the sampled frames are then fed to the video classification model.

Results: Table. 2 demonstrates the video classification results on the Breakfast dataset. We observe that KTS-based adaptive sampling achieves state-of-the-art results

Table 2: Video Classification results on Breakfast. We evaluate KTS-based sampling against uniform sampling with ViS4mer [13] as the baseline. Our approach achieves state-of-the-art performance with significantly less computation.

Method	Frames	Accuracy
VideoGraph [11]	64×8	69.50
Timeception [12]	1024×8	71.30
GHRM [37]	64×8	75.49
ViS4mer [13]	32×32	85.63
ViS4mer [13]	512×32	88.17
ViS4mer + KTS (Ours)	$\mathbf{32 \times 32}$	**89.86**

on the Breakfast dataset while utilizing $16\times$ fewer number of frames per video compared to the original ViS4mer baseline which uses uniform sampling. When compared with uniform sampling using the same setting [32×32], we observe a significant gain of 4.23% in terms of accuracy with KTS-based adaptive sampling, showing its superiority over uniform sampling.

3.4. Temporal Action Localization

Temporal Action localization (TAL) aims to identify the action instances present in a video in the temporal domain and recognize the action categories. Despite the steady progress in TAL performance in the modeling aspects (*e.g.,* action proposals [19], pretraining [1], single-stage TAL [35]), uniform sampling is adopted as the de facto sampling approach in most of the action localization models. We analyze the impact of the KTS-based adaptive sampling mechanism on action localization.

Baseline: We investigate the performance of KTS-based sampling on the strong Actionformer [35] baseline, which achieves the current state-of-the-art performance on TAL for ActivityNet. It comprises of a multi-scale transformer encoder which encodes the sequence of embedded video clip features into a feature pyramid. The feature pyramid is then followed by a classification and a regression head to recognize the action instance and estimate the action boundaries respectively. TSP [1] model pre-trained on ActivityNet video classification task is used to extract non-overlapping clip-level features. Refer to [35] for a complete description of Actionformer.

Implementation Details: Given a video, we downsample it to one frame per second when computing the KTS change points and use ResNet-50 [10] pre-trained on ImageNet-1K to extract feature descriptors for KTS computation. We adopt a similar training configuration as the Actionformer to study the impact of KTS-based adaptive sampling in TAL. Actionformer employs clips of 16 frames at a frame rate of 15 fps and a stride of 16 frames (i.e., non-overlapping clips) as input to the feature extractor followed by the localization module. This gives one feature vector per $\frac{16}{15} \approx 1.067$ seconds and $M = \frac{15}{16}T$ segments where T is the video length. We can also consider $\frac{M}{2}$, $\frac{M}{4}$, $\cdots$ segments by sampling

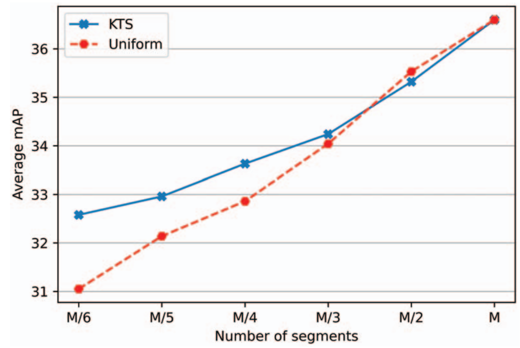

Figure 2: KTS vs Uniform sampling comparison on ActivityNet Action Localization. We report average mAP when varying the number of segments. M corresponds to the number of segments when each segment length is $\frac{16}{15}$ seconds as used in the Actionformer baseline.

every 2^{nd}, 4^{th}, $\cdots$ frame. Similarly, we can choose $\frac{M}{2}$, $\frac{M}{4}$, $\cdots$ segments in our KTS-based sampling strategy. For the baseline, all the segments have the same length while our adaptive sampling technique yields variable-length segments. Within each segment, we uniformly sample 16 frames in both cases. These frames are then fed to the action localization model. Fig. 2 provides a comparison of KTS vs uniform sampling, showing improved performance, especially for the smaller number of segments.

Results: Fig. 2 shows the empirical analysis of KTS-based sampling on TAL. Note that the performance gain of using KTS-based adaptive sampling is clearly observed for smaller number of segments (*e.g.,* $\frac{M}{3}$ and below), and the gap in performance increases when reducing the number of segments. In particular, for $\frac{M}{6}$ segments uniform sampling achieves 31.05% average mAP while KTS-based sampling attains 32.58% average mAP on ActivityNet, yielding 1.53% gain. For larger number of segments, the performance of KTS is nearly similar to uniform sampling. For M segments, KTS reduces to uniform sampling as there are M change point candidates when using one frame per second for sampling candidates. Similarly, for $\frac{M}{2}$ we select half of the candidates as change points, which makes it quite similar to uniform sampling.

4. Conclusion

In this work, we present an adaptive and task-agnostic frame sampling mechanism for long video modeling. Our approach leverages Kernel Temporal Segmentation (KTS) to generate semantically-consistent segments used for sampling frames. We perform a comprehensive set of experiments on video classification and temporal action localization on several long video understanding datasets and show the superiority of KTS-based adaptive sampling against existing sampling strategies. In spite of its simplicity, our approach achieves state-of-the-art performance on long-form video understanding benchmarks while being efficient.

References

[1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. TSP: Temporally-sensitive pretraining of video encoders for localization tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 3173–3183, 2021.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[4] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[5] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[7] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020.

[8] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15608–15618, 2021.

[9] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[10] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[11] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Videograph: Recognizing minutes-long human activities in videos. In *ICCV Workshop on Scene Graph Representation and Learning*, 2019.

[12] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Timeception for complex action recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 254–263, 2018.

[13] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 87–104, 2022.

[14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[15] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019.

[16] H. Kuehne, A. B. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2014.

[17] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[18] Jintao Lin, Haodong Duan, Kai Chen, Dahua Lin, and Limin Wang. Ocsampler: Compressing videos to one clip with single-step sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13903, 2022.

[19] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019.

[20] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022.

[21] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017.

[22] Yue Meng, Chung-Ching Lin, Rameswar Panda, Prasanna Sattigeri, Leonid Karlinsky, Aude Oliva, Kate Saenko, and Rogerio Feris. Ar-net: Adaptive frame resolution for efficient action recognition. *arXiv preprint arXiv:2007.15796*, 2020.

[23] Rameswar Panda, Chun-Fu Richard Chen, Quanfu Fan, Ximeng Sun, Kate Saenko, Aude Oliva, and Rogerio Feris. Adamml: Adaptive multi-modal learning for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7576–7585, 2021.

[24] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pages 540–555. Springer, 2014.

[25] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 347–363, 2018.

[26] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, D. Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2014.

[28] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[29] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021.

[30] Zuxuan Wu, Caiming Xiong, Yu-Gang Jiang, and Larry S Davis. Liteeval: A coarse-to-fine framework for resource efficient video recognition. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019.

[32] Boyang Xia, Wenhao Wu, Haoran Wang, Rui Su, Dongliang He, Haosen Yang, Xiaoran Fan, and Wanli Ouyang. Nsnet: Non-saliency suppression sampler for efficient video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 705–723. Springer, 2022.

[33] Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 982–990, 2016.

[34] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[35] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision (ECCV)*, pages 492–510, 2022.

[36] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 766–782. Springer, 2016.

[37] Jiaming Zhou, Kun-Yu Lin, Haoxin Li, and Wei-Shi Zheng. Graph-based high-order relation modeling for long-term action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8984–8993, 2021.

[38] Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Transactions on Image Processing*, 30:948–962, 2020.