TEMPORAL NORMALIZATION IN ATTENTIVE KEY-FRAME EXTRACTION FOR DEEP NEURAL VIDEO SUMMARIZATION

Michail Kaseris, Ioannis Mademlis, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

ABSTRACT

Attention-based neural architectures have consistently demonstrated superior performance over Long Short-Term Memory (LSTM) Deep Neural Networks (DNNs) in tasks such as key-frame extraction for video summarization. However, existing approaches mostly rely on rather shallow Transformer DNNs. This paper revisits the issue of model depth and proposes DATS: a deep attentive architecture for supervised video summarization that meaningfully exploits skip connections. Additionally, a novel per-layer temporal normalization algorithm is proposed that yields improved test accuracy. Finally, the model's noisy output is rectified in an innovative post-processing step. Experiments conducted on two common, publicly available benchmark datasets showcase performance superior to competing state-of-the-art video summarization methods, both supervised and unsupervised.

***Index Terms*—** key-frame extraction, video summarization, Deep Neural Network, Transformer, self-attention, batch normalization

1. INTRODUCTION

Automated video summarization consists in deriving succinct *summaries* of original, full-length videos, which capture the most important segments of the full input and jointly convey its essence in a compact manner. In *static* and *dynamic summarization*, the output is a set of still key-frames [1] and a short trailer/skim [2], respectively. The goal is to select an informative, representative and temporally ordered subset of the original/full video. This paper focuses on key-frame extraction, since skims can be easily constructed by concatenating video segments centered on extracted key-frames. These should be both visually representative of the full-length video and diverse in content.

Initial unsupervised approaches to key-frame extraction involved clustering or dictionary learning [1]. Later, supervised Deep Neural Networks (DNNs) were exploited to analyze convolutional representations of video frames and to output per-frame scalar *importance scores*. These are manually post-processed to derive the final video summary. Ground-truth per-frame importance scores are exploited for learning. Long Short-Term Memory networks (LSTMs) are typically utilized to capture inter-frame dependencies in supervised settings, by modeling the problem as a sequence-to-sequence task: mapping video frame representation sequences to importance score sequences [3].

Unsupervised approaches have also been presented, based on LSTMs. The state-of-the-art adversarial reconstruction framework [4] [5][6] entails training using a composite LSTM / GAN architecture, that contains multiple interacting neural modules. However, LSTMs encode the entire video sequence into a single, fixed-length vector and process each video frame one at a time. Transformer architectures relying on self-attention [7] were adopted to address this, allowing the DNN to find correlations across the sequence's items. However, only rather shallow Transformers have been employed [8], due to the assumption that Transformers are overparameterized and prone to overfitting.

Moreover, a well-known issue with DNNs is *internal covariate shift*: during training, the input data distribution of each layer varies across iterations, because the input of each layer is the output of the previous one, whose parameters are constantly being updated. As a result, each layer has to learn to accommodate/adjust to constant variations across training iterations. *Batch normalization* [9] is a differentiable per-layer operation which addresses the issue by allowing the DNN to learn to standardize the mean and the variance of each layer's inputs. Then, during inference on test data points, the relevant statistics stored at the end of training are employed for standardizing the input to each layer, across the forward pass. Alternatively, Layer normalization is common in Transformers: statistics are computed separately for each data point across all neurons and channels.

In video analysis, the temporally ordered stack of video frame representations displays a similar, yet unique kind of distribution shift between the embeddings of different video frames belonging to the same input video sequence. This phenomenon occurs separately at each neural layer and has the potential to significantly inhibit achievable task accuracy at the inference stage. Such an effect is likely to negatively influence both LSTMs and Transformers, since they both ultimately rely on correlating different inputs across the temporal axis. Yet, there is a distinct scarcity of methods specifically targeting this issue in the literature. The typical approach is

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 951911 (AI4Media).

to simply apply unit Euclidean norm normalization to each layer's activations [10, 11, 8]. As a result, useful information for each video is potentially discarded, since the resulting per-frame embeddings vary only with respect to their vector direction for different input videos. The only alternative approach that normalizes data across the temporal axis is [12]. It is called *Batch Normalization Through Time* (BNTT) and it is used to enhance the discriminative capacity of the representations generated by Spiking Neural Networks [13]. The BNTT module captures the mean and the standard deviation for each time-step of the i-th feature of the embeddings in the sequence within a mini-batch during training. At the inference stage, these statistics are used to normalize a new, unseen data point. Despite the BNTT's ability to create more discriminative temporal representations, it requires mini-batches that contain equally lengthed data points in terms of duration.

This paper proposes Deep Attentive Transformer Summarizer, or DATS: a new neural architecture for supervised video summarization, that attempts to address the above issues by integrating the following novel contributions:

1. A *deep* attentive neural network relying on self-attention and the bottleneck layer structure [14]. The output dimensionality of each bottleneck block's is half its original embedding dimensionality. Past video summarization methods have only employed shallow Transformers, whereas this paper meaningfully and effectively exploits architectural depth to increase performance, using skip connections.

2. A novel, temporal normalization module is proposed, implemented and integrated into the presented neural architecture. It can be used as a per-layer normalization algorithm for any timeseries modeling task, acting on the temporal axis of the input sequence. This paper evaluates it on supervised video summarization.

3. A denoising operator that smooths the predicted output is introduced and employed to video summarization for the first time. This is a non-neural post-processing step that adds no additional learnable parameters.

2. DEEP ATTENTIVE TRANSFORMER SUMMARIZER

Initially, DATS is fed an input sequence $\mathbf{X} \in \mathbb{R}^{T \times M}$, where T and M represent the video sequence length and the video frame representation dimensionality, respectively. The input sequence is the result of feeding the raw RGB video frames to a pretrained image recognition CNN and extracting an intermediate representation per-frame, typically from its penultimate layer. DATS consists of several consecutive attention layers [7]. Attention blocks mitigate the issue of encoding the entire sequence into a single fixed-length representation. Self-attention blocks are used for all layers and are defined as:

$$\mathbf{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}. \tag{1}$$

Eq. (1) computes the attention weights matrix $\mathbf{A} \in \mathbb{R}^{T \times T}$ that represents a measure of correlation between a video frame $\mathbf{x}_t$ and all other video frames. The matrices $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ denote the *Query*, *Key* and *Value* matrices, respectively, and they all take the form of a fully-connected layer. The input sequence is fed to the three aforementioned linear transformations forming the following set of equations:

$$\mathbf{Q} = \mathbf{W}_Q \cdot \mathbf{X} \tag{2}$$

$$\mathbf{K} = \mathbf{W}_K \cdot \mathbf{X} \tag{3}$$

$$\mathbf{V} = \mathbf{W}_V \cdot \mathbf{X}, \tag{4}$$

where $\mathbf{W}_{\{Q,K,V\}}$ denote the parameters that correspond to the Query, Key and Value subnetworks. DATS employs the multi-headed version of the attention mechanism, with an attention block consisting of H heads, hence having H sets of Query, Key and Value transformations. The H independent attention outputs are then concatenated and linearly transformed into the expected dimension. Intuitively, multiple attention heads allow for attending to parts of the sequence differently (e.g., longer-term dependencies versus shorter-term dependencies).

The multi-headed self-attention module is the fundamental element of DATS. The self-attention layers are stacked to form a deep attentive network. Each layer downsamples the sequence's dimensionality by a factor of 2 by using a fully-connected feed-forward layer, followed by a ReLU activation function. The self-attention block's output $\mathcal{A}(\mathbf{X})$ is added to the layer's input sequence $\mathbf{X}$, thus creating an identity skip connection [14]. As shown in Section 3, the output of the skip connection $\mathcal{H} = \mathcal{A}(\mathbf{X}) + \mathbf{X}$ contributes significantly to increased evaluation accuracy. For the inference head, the transformed sequence of the final self-attention block is fed to a cascade of two fully-connected layers that predict the per-frame importance scores $\mathbf{s} = \{s_t\}_{t=1}^{T}$.

DATS employs a novel, task-agnostic sequential data normalization method. In principle, it can be utilized for any sequence modeling task (e.g., machine translation, speech-to-text conversion, video captioning, etc.), although this paper evaluates the proposed neural architecture only on key-frame extraction/video summarization. In this context, a neural layer's input is a tensor $\mathcal{T} \in \mathbb{R}^{N \times T \times 1 \times M}$, where N denotes the batch size, T the number of video frames and M the video frame representation dimensionality, while the redundant unit dimension is a dummy one for conforming to tensor standards. The layer input consists in a temporally stacked collection of video frame representations.

Essentially, DATS incorporates a modification of batch normalization. However, instead of normalizing the k-th feature channel along the data point axis of the mini-batch, we normalize the k-th channel along the temporal axis, i.e., along the video frame sequence. It holds that $1 \leq k \leq M$. Formally, consider a collection of features $\mathcal{B} = \{\mathbf{X}_{t,k}\}_{t=1}^{T}$ that serves as the input to the current DNN layer. Then, the k-th channel is shifted and scaled in order to compute the normalized feature channel $\mathbf{y}^{(k)}$:

$$\mu_{\mathcal{B}} = \frac{1}{T}\sum_{t=1}^{T} x_t^{(k)} \tag{5}$$

$$\sigma_{\mathcal{B}}^2 = \frac{1}{T}\sum_{t=1}^{T}(x_t^{(k)} - \mu_{\mathcal{B}})^2 \tag{6}$$

$$\hat{\mathbf{x}}^{(k)} = \frac{\mathbf{x}^{(k)} - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \tag{7}$$

$$\mathbf{y}^{(k)} = \alpha\hat{\mathbf{x}}^{(k)} + \beta \tag{8}$$

In Eqs. (5)-(8), the mean and the standard deviation of the k-th feature dimension across the entire video sequence (i.e., across the temporal axis) is first computed. Then, $\mathcal{B}$ is standardized using the computed statistics $(\mu_{\mathcal{B}}, \sigma_{\mathcal{B}}^2)$. Similarly to [9], the Temporal Normalization module is a function parameterized by α and β. These scalar parameters are learnt during training along with the model's parameters.

Finally, in order to deal with the noisy outputs produced by the inference head, a simple operator $\phi(\cdot)$ is defined. It is appended to the tail of DATS, in order to smooth out the generated video frame importance scores in a post-processing step occurring immediately after inference. $\phi(\cdot)$ acts on a small temporal region $\mathcal{R}^{(t)}$ in a neighbourhood around a video frame indexed by t:

$$\mathcal{R}^{(t)} = \{s_t | \max(0, t-s) \leq t < \min(t+s, T), s_t \in \mathbf{s}\}. \tag{9}$$

Parameter s is a user-defined hyperparameter that dictates the region size. In general, supervised key-frame extraction ground-truth labels provided by human annotators do not present abrupt changes in the per-frame importance scores. However, the raw output generated from typical summarization DNNs during inference commonly incorporates high-frequency noise components, when viewed as an 1D signal defined over the temporal domain. This implies that a factor limiting the accuracy of existing summarization DNNs is their inability to learn the concept of smoothness.

One way to overcome this issue is to *learn* a parametric operator that shifts and scales the importance scores based on their value within a small temporal region. This paper argues that a simpler non-parametric operator $s'_t = \max(\mathcal{R}^{(t)})$ can provide satisfactory performance without inducing any additional computational cost.

The proposed deep attentive neural network architecture is trained by minimizing the Mean Square Error (MSE) loss function, computed between the predicted video frame importance scores $\mathbf{s} \in [0,1]^T$ and the respective ground-truth importance scores $\mathbf{y} \in [0,1]^T$. As a preprocessing step, the minimum value of each ground-truth importance score sequence is first subtracted from every individual importance score. Then, each per-frame score is divided by the resulting maximum:

$$\mathbf{y}_{shift} := \mathbf{y}_{old} - \min(\mathbf{y}_{old}) \tag{10}$$

$$\mathbf{y} := \frac{\mathbf{y}_{shift}}{\max(\mathbf{y}_{shift})}. \tag{11}$$

By doing so, the model has to learn for all videos to predict consistently a minimum/maximum value of 0/1, respectively. This strategy prevents overfitting.

Table 1: Peformance comparisons in the canonical setting, in terms of F1-Score.

Method	Datasets	
	TVSum	**SumMe**
CLIP-It!$_{supervised}$ [15]	66.3 %	54.2%
iPTNet [16]	63.4%	54.5%
PGL-SUM [11]	61.0%	55.6%
M-AVS [17]	61.0%	44.4%
SUM-GAN-AAE [18]	58.3%	48.9%
CA-SUM [8]	61.4%	51.1%
DSNet$_{anchor-based}$ [10]	62.1%	50.2%
DSNet$_{anchor-free}$ [10]	61.9%	51.2%
AC-SUM-GAN [19]	60.6%	50.8%
RR-STG [20]	63.0%	53.4%
DATS (proposed)	**73.3%**	**61.8%**

Table 2: DATS ablation study.

Model	Datasets	
Variant	**TVSum**	**SumMe**
w/o Output Smoothing	73.1%	56.7%
w/o Temporal Normalization	70.5%	57.4%
w/o Skip Connections	62.0%	47.9%
Complete DATS	**73.3%**	**61.8%**

3. EXPERIMENTAL EVALUATION

All aspects of the evaluation process follow established common protocols, utilizing 2 public datasets: TVSum [21] and SumMe [22]. TVSum contains 50 videos with their visual content ranging from documentaries, news and television shows to video tutorials. SumMe is comprised of 25 videos characterized by a relatively shorter duration and covering more diverse content, such as holidays and sports. According to established protocol, two additional datasets were employed to augment TVSum/SumMe during training: OVP and YouTube [23]. However, test/evaluation does not involve these two auxiliary datasets. YouTube and OVP consist of 39 and 50 video sequences, respectively. Following common convention, all videos are temporally subsampled at a rate of 2 Frames Per Second (FPS), starting from an original framerate of 30 FPS for all videos/datasets.

Three evaluation settings were adopted, separately for SumMe and for TVSum: *canonical*, *augmented* and *transfer*.

Table 3: Comparison of best performers in the 3 evaluation settings.

Method	**TVSum**			**SumMe**		
	Canonical	Augmented	Transfer	Canonical	Augmented	Transfer
iPTNet [16]	64.4%	64.2%	59.8%	54.5%	**56.9**%	49.2%
DSNet$_{a-b}$ [10]	62.1%	63.9%	59.4%	50.2%	50.7%	46.5%
DSNet$_{a-f}$ [10]	61.9%	62.2%	58.0%	51.2%	53.3%	47.6%
DATS (proposed)	**73.3**%	**73.5**%	**67.4**%	**61.8**%	56.2%	**62.7**%

In the first two settings, 80% of TVSum/SumMe was used for training and the remaining videos were employed only for test/evaluation. The validation set was randomly divided into 5 splits. Moreover, in the augmented setting, the current dataset (either TVSum or SumMe) is augmented with the remaining three datasets during training (YouTube, OVP and either SumMe or TVSum). The transfer setting dictates that the DNN is trained on the union of 3 datasets and evaluated on the remaining one. The canonical setting was the default comparison protocol.

Per-frame semantic representations were extracted for each video frame from the pool5 layer of a pretrained GoogLeNet, according to common conventions. DATS incorporates 3 self-attention layers and the embeddings dimensionality is halved on each layer's output. Every self-attention layer has 8 attention heads. For each experiment, the model was trained for 500 epochs with the Adam optimizer, using a learning rate equal to 10^{-4} and a weight decay equal to 10^{-5}. In order to generate the final key-frame summary *after* neural inference has been completed for a test video, the shot cut/scene change points must be detected by employing the Kernel Temporal Segmentation (KTS) algorithm. This step is essential so as to then execute the 1/0 Knapsack algorithm, which requires both the predicted video frame scores and video frame weights. The weights are are trivially derived by evaluating the video frame count of the shots defined by the detected shot cuts.

The F1-Score metric is utilized evaluating DATS and its competitors. For a given predicted summary A and a ground-truth summary U, the precision and recall metrics, P and R are defined as:

$$P = \frac{||A \cap U||}{||U||}, \qquad R = \frac{||A \cap U||}{||A||}, \tag{12}$$

where the $||\cdot||$ denotes the cardinality of a set. The F1-Score metric F is computed as follows:

$$F = \frac{P \cdot R}{P + R}. \tag{13}$$

For TVSum and SumMe, the predicted summary's quality is evaluated by comparing it to the ground-truth summary, derived from the ground-truth per-frame importance scores. During validation, both the average F1-Score (across all 5 test dataset splits) and the maximum F1-Score (across all 5 splits and all training epochs) is computed for each test video sequence, as dictated in [21] and [22], respectively.

DATS is compared against recent, state-of-the-art deep neural key-frame extraction methods, both supervised and unsupervised, using the canonical setting. As it can be seen in Table 1, DATS achieves top performance and a gain of 7% in F1-Score against the latest state-of-the-art method, i.e., CLIP-It! [15]. With 6.5 million parameters, the proposed approach also attains a favorable balance between complexity (number of parameters) and accuracy.

DATS was also evaluated on the Augmented and the Transfer settings, by introducing videos from the OVP and YouTube datasets during training. As shown in Table 3, DATS surpasses competing methods in these settings as well. Moreover, it actually performs slightly better than in the canonical setting, most likely due to training on a larger/augmented dataset. The challenging transfer setting task reveals that DATS can generalize well on unknown samples, provided that it has been trained on a sufficiently large dataset. Finally, Table 2 indicates that the utilized skip connections are instrumental for architectural depth to be effective.

4. CONCLUSIONS

This paper presented DATS: a novel deep attentive architecture for key-frame extraction in video summarization tasks. Based on the overall experimental evaluation, there are three important findings we can highlight. First, it seems that a greater number of training data points contributes to increased summarization accuracy, even under a domain/data distribution shift between the training and the test stage. Second, all three main components of DATS are essential to its high accuracy: i) the deep structure with the skip connections, ii) the temporal normalization algorithm, and iii) the output smoothness operator. Third, deep attentive architecture such as the proposed one hold the promise of alleviating model complexity, without sacrificing accuracy. In fact, DATS performs astonishingly well at a fraction of the model complexity its competitors showcase.

5. REFERENCES

[1] I. Mademlis, A. Tefas, and I. Pitas, "A salient dictionary learning framework for activity video summarization via key-frame extraction," *Information Sciences*, vol. 432, pp. 319–331, 2018.

[2] I. Mademlis, A. Tefas, N. Nikolaidis, and I. Pitas,

"Movie shot selection preserving narrative properties," in *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2016.

[3] M. Rochan, L. Ye, and Y. Wang, "Video summarization using Fully Convolutional sequence Networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[4] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] M. Kaseris, I. Mademlis, and I. Pitas, "Exploiting caption diversity for unsupervised video summarization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[6] M. Kaseris, I. Mademlis, and I. Pitas, "Adversarial unsupervised video summarization augmented with dictionary loss," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2021.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[8] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames," in *Proceedings of the ACM International Conference on Multimedia Retrieval (ICMR)*, 2022.

[9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2015, pp. 448–456.

[10] W. Zhu, J. Lu, J. Li, and J. Zhou, "DSNet: A flexible detect-to-summarize network for video summarization," *IEEE Transactions on Image Processing*, vol. 30, pp. 948–962, 2020.

[11] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Combining global and local attention with positional encoding for video summarization," in *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 2021.

[12] Y. Kim and P. Panda, "Revisiting Batch Normalization for training low-latency Deep Spiking Neural Networks from scratch," *Frontiers in Neuroscience*, p. 1638, 2020.

[13] J.H. Lee, T. Delbruck, and M. Pfeiffer, "Training Deep Spiking Neural Networks using backpropagation," *Frontiers in Neuroscience*, vol. 10, pp. 508, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] M. Narasimhan, A. Rohrbach, and T. Darrell, "CLIP-It! language-guided video summarization," *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, vol. 34, 2021.

[16] H. Jiang and Y. Mu, "Joint video summarization and moment localization by cross-task sample transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder-decoder networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1709–1717, 2019.

[18] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Unsupervised video summarization via attention-driven adversarial learning," in *Proceedings of the International Conference on Multimedia Modeling (MMM)*. Springer, 2020.

[19] E. Apostolidis, E. Adamantidou, AI Metsai, V. Mezaris, and I. Patras, "AC-SUM-GAN: connecting Actor-Critic and Generative Adversarial Networks for unsupervised video summarization," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 3278–3292, 2021.

[20] W. Zhu, Y. Han, J. Lu, and J. Zhou, "Relational reasoning over spatial-temporal graphs for video summarization," *IEEE Transactions on Image Processing*, vol. 31, pp. 3017–3031, 2022.

[21] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "TVSum: Summarizing web videos using titles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2014.

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