CSTA: CNN-based Spatiotemporal Attention for Video Summarization

Jaewon Son, Jaehun Park, Kwangsu Kim*
Sungkyunkwan University
{31z522x4,pk9403,kim.kwangsu}@skku.edu

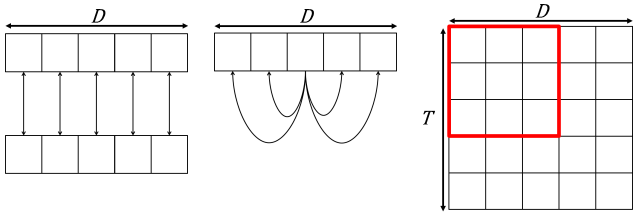
Abstract

*Video summarization aims to generate a concise representation of a video, capturing its essential content and key moments while reducing its overall length. Although several methods employ attention mechanisms to handle long-term dependencies, they often fail to capture the visual significance inherent in frames. To address this limitation, we propose a **C**NN-based **S**patio**T**emporal **A**ttention (CSTA) method that stacks each feature of frames from a single video to form image-like frame representations and applies 2D CNN to these frame features. Our methodology relies on CNN to comprehend the inter and intra-frame relations and to find crucial attributes in videos by exploiting its ability to learn absolute positions within images. In contrast to previous work compromising efficiency by designing additional modules to focus on spatial importance, CSTA requires minimal computational overhead as it uses CNN as a sliding window. Extensive experiments on two benchmark datasets (SumMe and TVSum) demonstrate that our proposed approach achieves state-of-the-art performance with fewer MACs compared to previous methods. Codes are available at* https://github.com/thswodnjs3/CSTA.

1. Introduction

The rise of social media platforms has resulted in a tremendous surge in daily video data production. Due to the high volume, diversity, or redundancy, it is time-consuming and equally difficult to retrieve the desired content or edit multiple videos. Video summarization is a powerful time-saving technique to condense long videos by retaining the most relevant information, making it easier for users to quickly grasp the main points of the video without having to watch the entire footage.

One of the challenges that occur during video summarization is the long-term dependency problem, where the initial information is often lost due to large data intervals [18, 26, 43, 44]. The decay of initial data prevents deep learning models from capturing the relation between frames essential for determining key moments in videos. Attention [38], in which entire frames are reflected through pairwise operations, has gained popularity as a widely adopted technique for solving this problem [1, 7, 15, 17, 46]. Attention-based models distinguish important parts from unimportant ones by determining the mutual reliance between frames. However, attention cannot consider spatial contexts within images [15, 27, 39, 43, 48]. For instance, current attention calculates temporal attention based on correlations of visual attributes from other frames (See Figure 1a), but the importance of visual elements within the frame remains unequal to the temporal significance. Including spatial dependency leads to different weighted values of features, causing changes in temporal importance. Therefore, attention can be calculated more precisely by including visual associations, as shown in Figure 1b.



(a) Temporal attention (b) Spatial attention (c) CNN attention

Figure 1. Approaches for calculating attention. Each row is the feature vector of a frame. *T* is the number of frames, and *D* is the dimension of the feature.

Prior studies mixed spatial importance and performed better than solely relying on sequential connections [15, 27, 39, 43, 48]. Nevertheless, acquiring spatial and temporal importance requires the design of additional modules and, thus, incurs excessive costs. Some studies used additional structures to embrace visual relativities in individual frames, such as self-attention [15, 39], multi-head attention [43], and graph convolutional neural networks [48]. Processing too many frames of lengthy videos to capture the temporal and visual importance can be expensive. Thus,

*Corresponding author

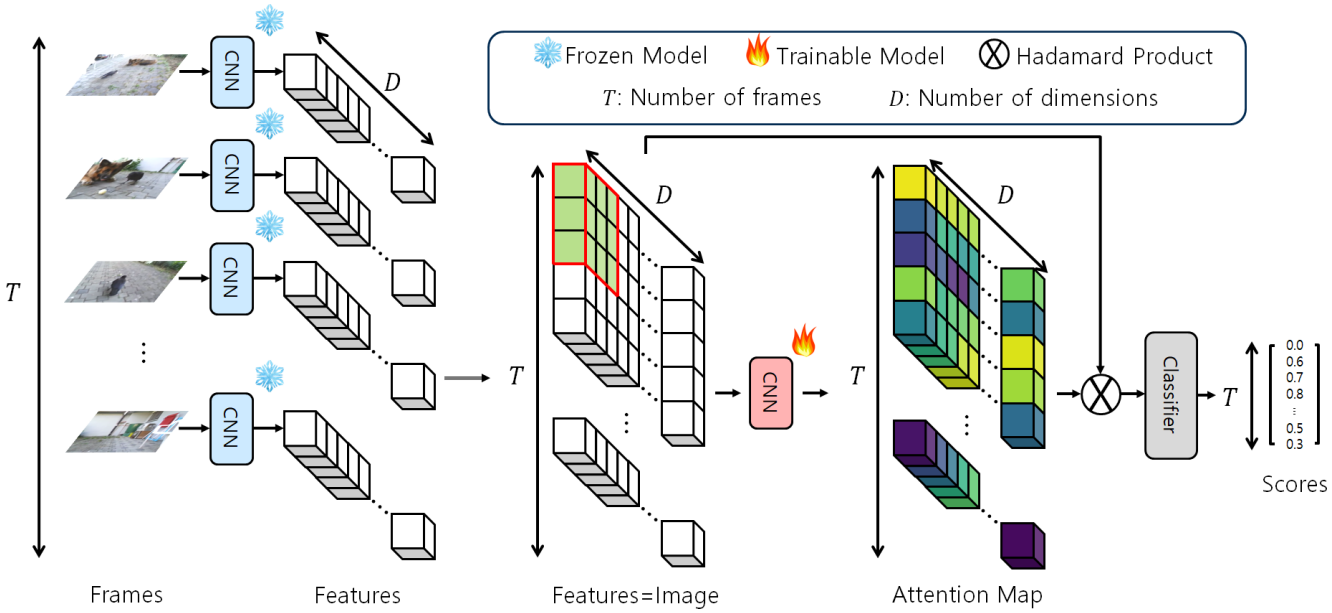


Figure 2. Workflow of CSTA

obtaining both inter and intra-frame relationships with few computation resources becomes a non-trivial problem.

This paper introduces **C**NN-based **S**patio**T**emporal **A**ttention (CSTA) to simultaneously capture the visual and ordering reliance in video frames, as shown in Figure 2. CSTA works as follows: Firstly, it extracts features of frames from a video and then concatenates them. Secondly, it treats the assembled frame representations as an image and applies a 2D convolutional neural network (CNN) model to them for producing attention maps. Finally, it combines the attention maps with frame features to predict the importance scores of frames. CSTA derives spatial and temporal relationships in the same manner as CNN derives patterns from images, as shown in Figure 1c. Further, it searches for vital components in frame representations with the capacity of a CNN to infer absolute positions from images [16, 21]. Unlike previous methods, CSTA is efficient as a one-way spatiotemporal processing algorithm because it uses a CNN as a sliding window.

We test the efficacy of CSTA on two benchmark datasets - SumMe [12] and TVSum [34]. Our experiment validates that a CNN produces attention maps from frame features. Further, CSTA needs fewer multiply-accumulate operations (MACs) than previous methods for considering the visual and sequential dependency. Our contributions are summarized below:

- To the best of our knowledge, the proposed model appears to be the first to apply 2D CNN to frame representations in video summarization.
- The CSTA design reflects spatial and temporal associations in videos without requiring considerable computational resources.
- CSTA demonstrates state-of-the-art based on the overall results of two benchmark datasets, SumMe and TVSum.

2. Related Work

2.1. Attention-based Video Summarization

Many video summarization models use attention to deduce the correct relations between frames and find crucial frames in videos. A-AVS and M-AVS [17] are encoder-decoder structures in which attention is used to find essential frames. VASNet [7] is based on plain self-attention for better efficiency than encoder-decoder-based ones. SUM-GDA [26] also employs attention for efficiency and supplements diversity into the attention mechanism for generated summaries. CA-SUM [2] further enhances SUM-GDA by introducing uniqueness into the attention algorithm in unsupervised ways. Attention in DSNet [47] helps predict scores and precise localization of shots in videos. PGL-SUM [1] has a mechanism to alleviate long-term dependency problems by discovering local and global relationships by applying multi-head attention to segments and the entire video. GL-RPE [20] approaches similarly in unsupervised ways by local and global sampling in addition to relative position and attention. VJMHT [24] uses transformers and improves summarization by learning similarities between analogous videos. CLIP-It [29] also relies on the transformers to predict scores by cross-attention between frames and captions of the video. Attention helps models recognize the relations between frames, however, it does not focus on visual rela-

tions.

Visual relevance is vital to understanding video content as it influences the expression of temporal dependency. Some studies have proposed additional networks to find frame-wise visual relationships [15, 27, 39, 43]. The models process the temporal dependency and exploit self-attention or multi-head attention for visual relations of every frame. RR-STG [48] uses graph CNNs to draw spatial associations using graphs. RR-STG creates graphs based on elements from object detection models [32] to capture the spatial relevance. These methods offer increased performance but incur a high computational cost owing to the separate module handling many frames. This paper adopts CNN as a one-way mechanism for more efficient reflecton of the spatiotemporal importance of multiple frames in long videos.

2.2. CNN for Efficiency and Absolute Positions

CNN is usually employed to resolve computation problems in attention. CvT [40] uses CNN for token embedding and projection in vision transformers (ViT) [6] and requires a few FLOPs. CeiT [41] uses both CNN and transformers and shows better results with fewer parameters and FLOPs. CmT [11] applies depth-wise convolutional operations to obtain a better trade-off between accuracy and efficiency for ViT. We exploit CNN to enhance the efficiency of dealing with multiple frames in video summarization.

CNN can be used for attention by learning absolute positions from images. Islam *et al*. [16] proved that features extracted using a CNN contain position signals. They attributed it to padding, and Kayhan and Germert [21] verified the same under various paddings. CPVT [4] uses this ability to reflect the position information of tokens and to tackle problems in previous positional encodings for ViT. Based on this behavior of CNNs, our proposed method is designed to seek only the necessary elements for video summarization from frame representations by considering frame features as images.

3. Method

3.1. Overview

This study approaches video summarization as a subset selection problem. We show the proposed CSTA framework in Figure 3. During the *Embedding Process*, the model converts the frames into feature representations. The *Prediction Process* involves using these representations to predict importance scores. In the *Prediction Process*, the *Attention Module* generates attention for videos, and the *Mixing Module* fuses this attention with input frame features. Finally, the CSTA predicts every frame's importance score, representing the probability of whether the frame should be included in the summary videos. The model is trained by comparing estimated scores and human-annotated scores. During inference, it selects frames based on the knapsack algorithm and creates summary videos using them.

3.2. Embedding Process

CSTA converts frames to features for input into the model, as depicted in the *Embedding Process* (Figure 3). Let the frames be $X = \{x_i\}_{t=1}^{T}$ when there are T frames in a video, with H as the height and W as the width. Following [7, 9, 25, 39, 42, 47] for a fair comparison, the frozen pre-trained CNN model (GoogleNet [35]) modifies $X \in \mathbb{R}^{T\times 3\times H\times W}$ into $X' \in \mathbb{R}^{T\times D}$ where D is the dimension of frame features.

To fully utilize the CNN, we replicate the frame representations to match the number of channels (*i.e*., three). A CNN is usually trained using RGB images [14, 33, 35, 36]; therefore, pre-trained models are well-optimized on images with three channels. Additionally, we concatenate the classification token (CLS token) [6, 15] into frame features:

$$X'' = Concat_{axis=0}(X', X', X') \tag{1}$$

$$E = Concat_{axis=1}(X_{CLS}, X'') \tag{2}$$

where $X'' \in \mathbb{R}^{3\times T\times D}$ and $X_{CLS} \in \mathbb{R}^{3\times 1\times D}$ are the appended feature and the CLS token, respectively. $E \in \mathbb{R}^{3\times (T+1)\times D}$ is the embedded feature. $Concat_{axis=0}$ and $Concat_{axis=1}$ concatenate features in the channel axis and T axis, respectively. Motivated by STVT [15], we append the CLS token with input frame features. The CLS token is the learnable parameters fed into the models with inputs and trained with models jointly. STVT obtains correlations of frames using the CLS token and aggregates the CLS token with input frames to capture global contexts. We follow the same method in prepending and combining the CLS token with frame features. The fusing process is completed later in the *Mixing Module*.

3.3. Prediction Process

CSTA calculates importance scores for T frames, as shown in the *Prediction Process* (Figure 3). The classifier assigns scores to frames after the *Attention Module* and *Mixing Module*. The *Attention Module* makes attention maps from E, and the *Mixing Module* aggregates this attention with E. A detailed explanation is given in Algorithm 1.

We generate the key and value from E by using two linear layers based on the original attention [38]. The metrics W^K and $W^V \in \mathbb{R}^{D\times D}$ are weights of linear layers projecting E into the key and value (Line 2-Line 3). Unlike E^K, CSTA uses a single channel of frame features in E to produce features by value embedding (Line 3) because we only need one X' except for duplicated ones, which are simply used for reproducing image-like features. We select the first index as a representative, which is $E[0] \in \mathbb{R}^{(T+1)\times D}$.

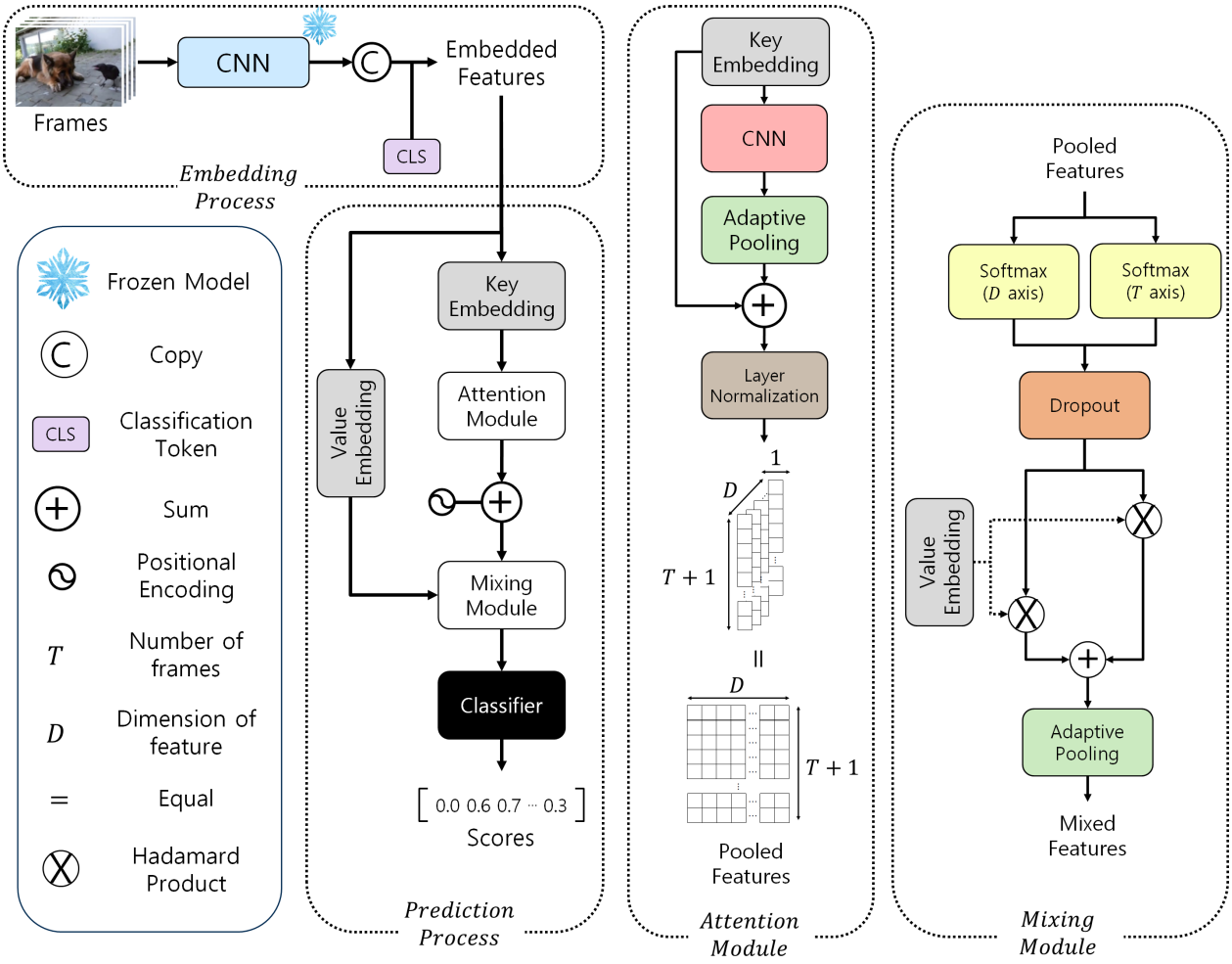


Figure 3. Architecture of CSTA

Algorithm 1: *Prediction Process*

input : $E \in \mathbb{R}^{3\times(T+1)\times D}$
output: $S \in \mathbb{R}^{T}$

1 **begin**
2 $E^K = W^K E$
3 $E^V = W^V E[0]$
4
5 $P = Attention\ Module(E^K)$ Section 3.4
6 $P_{pos} = P + Positional\ Encoding$
7 $M = Mixing\ Module(P_{pos}, E^V)$ Section 3.5
8 $S = Classifier(M)$
9 **return** S
10 **end**

The *Attention Module* processes spatiotemporal characteristics and focuses on critical attributes in E^K (Line 5). We add positional encodings to P to strengthen the absolute position awareness further (Line 6). Unlike the prevalent way of adding positional encoding into inputs [6, 38], this study adds positional encoding into the attention maps based on [1]. This is because adding positional encodings into input features distorts images so that models can recognize this distortion as different images. Moreover, models cannot fully recognize these absolute position encodings in images during training owing to a lack of data. Therefore, CSTA makes P_{pos} by attaching positional encodings to attentive features P.

The *Mixing Module* inputs P_{pos} and E^V and produces mixed features $M \in \mathbb{R}^{(T+1)\times D}$ (Line 7). The classifier predicts importance scores vectors $S \in \mathbb{R}^{T}$ from M (Line 8).

3.4. Attention Module

The *Attention Module* (Figure 3) produces attention maps by utilizing a trainable CNN (GoogleNet [35]) handling frame features E^K. The CNN captures the spatiotemporal dependency using kernels, similar to how a CNN

learns from images, as shown in Figure 1c. The CNN also searches for essential elements from E^K for summarization, with the ability to learn absolute positions. Based on [4, 16, 21], CNN imbues representations with positional information so that CSTA can encode the locations of significant attributes from frame features for summarization.

We make the shape of attention maps the same as that of input features to aggregate attention maps with input features. This study leverages two strategies for equal scale: deploying the adaptive pooling operation and using the same CNN model (GoogleNet [35]) in the *Embedding Process* and *Attention Module*. Pooling layers reduce the scale of features in the CNN; therefore, the size of outputs from the CNN is changed from $E^K \in \mathbb{R}^{3\times(T+1)\times D}$ to $E^K_{CNN} \in \mathbb{R}^{D\times\frac{T+1}{r}\times\frac{D}{r}}$, where r is the reduction ratio. To expand diverse lengths of frame representations, we exploit adaptive pooling layers to adjust the shape of features by bilinear interpolation. Furthermore, the number of output channels from the learnable CNN equals the dimension of frame features from the fixed CNN because of the same CNN models. The output from adaptive pooling is $E^K_{pool} \in \mathbb{R}^{D\times(T+1)\times 1}$.

As suggested in [14], this study uses a skip connection:

$$P = LayerNorm(E^K_{pool} + E^K[0]) \tag{3}$$

where the output is $P \in \mathbb{R}^{D\times(T+1)}$, followed by layer normalization [3]. A skip connection supports more precise attention and stable training in CSTA. As same with E^V, explained in Algorithm 1 (Line 3), we only use the single frame feature of E^K and ignore replications of frame features.

The size of P is equal to the size of frame features with $(T+1)\times D$; therefore, each value of P has the spatiotemporal importance of frame features. By combining P with frame features, the CSTA reflects the sequential and visual significance of frames. After supplementing the positional encodings, P_{pos} will be used as inputs for the *Mixing Module*.

3.5. Mixing Module

In the *Mixing Module* (Figure 3), we employ softmax along the time and dimension axes to compute the temporal and visual weighted values of P_{pos}:

$$Att_T : \sigma(d_i) = \left(\frac{e^{d_i}}{\sum_j e^{d_j}}\right) \; j = 1, ..., T+1 \tag{4}$$

$$Att_D : \sigma(d_i) = \left(\frac{e^{d_i}}{\sum_k e^{d_k}}\right) \; k = 1, ..., D \tag{5}$$

where Att_T is the temporal importance, and Att_D is the visual importance. Equation (4) calculates the weighted values between $T+1$ frames, including the CLS token, in the same dimension. Equation (5) computes the weighted values between different dimensions in the same frame. Att_D represents the spatial importance because each value of the dimension from features includes visual characteristics by CNN, processing image patterns, and producing informative vectors.

After acquiring weighted values, a dropout is employed for these values before integrating them with E^V. The dropout erases parts of features by setting 0 values for better generalization; it also works for attention, as shown in [1, 38]. If a dropout is applied to inputs as in the original attention [38], the CNN cannot learn contexts from 0 values, unlike self-attention, because the dropout spoils the local contexts of deleted parts. Therefore, we follow [1] by applying the dropout to the output of the softmax operations for generalization.

After dropout, the CSTA combines the spatial and temporal importance with the frame features:

$$M = Att_T \odot E^V + Att_D \odot E^V \tag{6}$$

where $\odot$ is the element-wise multiplication, and $M \in \mathbb{R}^{(T+1)\times D}$ is the mixed representations. CSTA reflects weighted values into frame features by blending Att_T and Att_D with E^V by element-wise multiplication. Incorporating visual and sequential attention values by addition encompasses spatiotemporal importance at the same time.

Subsequently, to integrate the CLS token with frame features, adaptive pooling transforms $M \in \mathbb{R}^{(T+1)\times D}$ into $M' \in \mathbb{R}^{T\times D}$ by average. Unlike STVT [15], in which linear layers are used to merge the CLS token with constant numbers of frames, CSTA uses adaptive pooling to cope with various lengths of videos. Adaptive pooling fuses the CLS token with a few frames; however, it intensifies our model owing to the generalization of the classifier, which consists of fully connected layers. M' from adaptive pooling enters into the classifier computing importance scores of frames.

3.6. Classifier

Based on the output of the adaptive pooling, the classifier exports the importance scores. We follow [1, 7, 13] to construct the structure of the classifier as follows:

$$R = LayerNorm(Dropout(ReLU(FC(M')))) \tag{7}$$

$$S = Sigmoid(FC(R)) \tag{8}$$

where $R \in \mathbb{R}^{T\times D}$ is derived after M' passes through a fully connected layer, relu, dropout, and layer normalization. Another fully connected layer maps the representation

of each frame into single values, and the sigmoid computes scores $S \in \mathbb{R}^T$.

We train CSTA by comparing predicted and ground truth scores. For the loss function, we use the mean squared loss as follows:

$$Loss = \frac{1}{T}\sum(S_p - S_g)^2 \tag{9}$$

where S_p is the predicted score, and S_g is the ground truth score.

The CSTA creates summary videos based on shots that KTS [31] derives. It computes the average importance scores of shots into which KTS splits videos [42]. The summary videos consist of shots with two constraints:

$$max \sum S_i \tag{10}$$

$$\sum Length_i \leq 15\% \tag{11}$$

where i is the index of selected shots. $S_i \in [0, 1]$ is the importance score of the ith shot between 0 and 1, and $Length_i$ is the percentage of the length of the ith shot in the original videos. Our model picks shots with high scores by exploiting the 0/1 knapsack algorithm as in [34]. Following [12], summary videos have a length limit of 15% of the original videos.

4. Experiments

4.1. Settings

Evaluation Methods. We evaluate CSTA using Kendall's (τ) [22] and Spearman's (ρ) [49] coefficients. Both metrics are rank-based correlation coefficients that are used to measure the similarities between model-estimated and ground truth scores. The F1 score is the most commonly used metric in video summarization; however, it has a significant drawback when used to evaluate summary videos. Based on [30, 37], due to the limitation of the summary length, the F1 score is evaluated to be higher if models choose as many short shots as possible and ignore long key shots. This fact implies that the F1 score might not represent the correct performance in video summarization. A detailed explanation of how to measure correlations is provided in Appendix A.1.

Datasets. This study utilizes two standard video summarization datasets - SumMe [12] and TVSum [34]. SumMe consists of videos with different contents (*e.g.*, holidays, events, sports) and various types of camera angles (*e.g.*, static, egocentric, or moving cameras). The videos are raw or edited public ones with lengths of 1-6 minutes. At least 15 people create ground truth summary videos for all data, and the models predict the average number of selections by people for every frame. TVSum comprises 50 videos from 10 genres (*e.g.*, documentaries, news, vlogs). The videos are 2-10 minutes long, and 20 people annotated the ground truth for each video. The ground truth is a shot-level importance score ranging from 1 to 5, and models try to estimate the average shot-level scores.

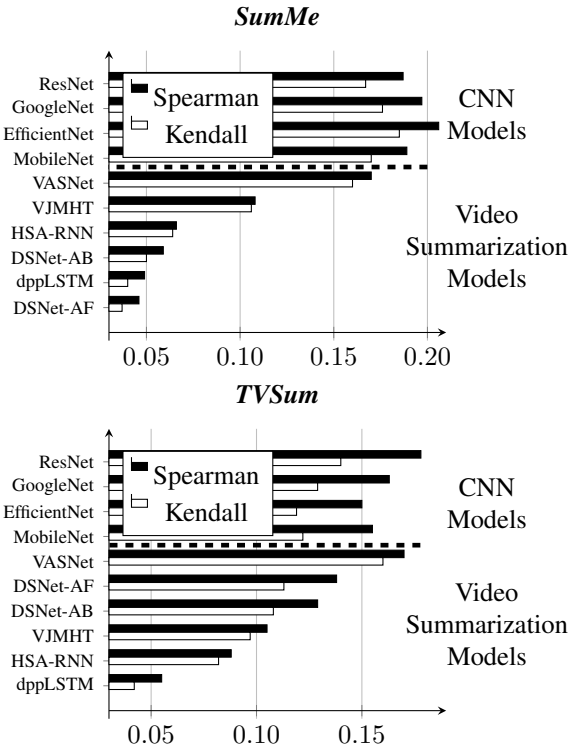


Figure 4. Comparison of summarizing performance between CNN and video summarization models. The x-axis shows performance, and the y-axis shows model names. Based on the dashed line, the performance of CNN is displayed above, and the video summarization models are below.

Implementation details are explained in Appendix A.2.

4.2. Verification of Attention Maps being Created using CNN

Previous studies on video summarization have yet to apply 2D CNN directly to frame features. Therefore, we verify that CNN can create attention maps from frame features. We choose MobileNet-V2 [33], EfficientNet-B0 [36], GoogleNet [35], and ResNet-18 [14] as CNN models since we focus on limited computation costs. This study applies CNN models to frame features and trains them to compute the frame-level importance scores without the classifier. The CNN directly exports T scores by inputting its output features into the adaptive pooling layer with a target shape $T \times 1$. As the importance score of each frame is be-

Method	SumMe			TVSum		
	Rank	τ	ρ	Rank	τ	ρ
Random	-	0.000	0.000	-	0.000	0.000
Human	-	0.205	0.213	-	0.177	0.204
dppLSTM[42]	15	0.040	0.049	22	0.042	0.055
DAC[8]^{T}	12.5	0.063	0.059	21	0.058	0.065
HSA-RNN[45]	11.5	0.064	0.066	19.5	0.082	0.088
DAN[27]ST	-	-	-	19.5	0.071	0.099
STVT[15]ST	-	-	-	15.5	0.100	0.131
DSNet-AF[47]^{T}	16	0.037	0.046	13.5	0.113	0.138
DSNet-AB[47]^{T}	13.5	0.051	0.059	15	0.108	0.129
HMT[46]^{M}	10.5	0.079	0.080	17.5	0.096	0.107
VJMHT[24]^{T}	8.5	0.106	0.108	17.5	0.097	0.105
CLIP-It[29]^{M}	-	-	-	13.5	0.108	0.147
iPTNet[19]$^{+}$	8.5	0.101	0.119	11	0.134	0.163
A2Summ[13]^{M}	7	0.108	0.129	10	0.137	0.165
VASNet[7]^{T}	6	0.160	0.170	9	0.160	0.170
AAAM[37]^{T}	-	-	-	6.5	0.169	0.223
MAAM[37]^{T}	-	-	-	5.5	0.179	0.236
VSS-Net[43]ST	-	-	-	3	0.190	0.249
DMASum[39]ST	11	0.063	0.089	**1**	**0.203**	**0.267**
RR-STG[48]ST	2.5	0.211*	0.234	7.5	0.162	0.212
MSVA[9]^{M}	3.5	0.200	0.230	5.5	0.190	0.210
SSPVS[25]^{M}	3*	0.192	0.257*	4.5	0.181	0.238
GoogleNet[35]ST	5	0.176	0.197	11.5	0.129	0.163
CSTAST	**1**	**0.246**	**0.274**	2*	0.194*	0.255*

Table 1. Comparison between CSTA and state-of-the-art on SumMe and TVSum. Rank is the average rank between Kendall's (τ) and Spearman's (ρ) coefficients. We categorize different types of video summarization models: temporal (T) and spatiotemporal (ST) attention-based, multi-modal based (M), and external dataset-based ($+$) models. The scores marked in bold and by the asterisk are the best and second-best ones, respectively. GoogleNet is the baseline model. Note that all feature extraction models are CNNs for a fair comparison.

tween 0 and 1, each score is similar to the weighted value of each frame. Thus, we can test whether CNN generates attention maps based on the video summarization performance. Surprisingly, the CNN models predict the importance scores much better than the previous video summarization models on SumMe, as shown in Figure 4. Even though the CNN models do not perform best on TVSum, they still show promising performance compared to existing video summarization models. The results show that the CNN produces attention maps by capturing the spatiotemporal relations and detecting crucial attributes in frame features based on absolute position encoding ability, unlike conventional methods that solely address the temporal dependency.

4.3. Performance Comparison

We compare CSTA with existing state-of-the-art methods on SumMe and TVSum. The results in Table 1 show that CSTA achieves the best performance on SumMe and the second-best score on TVSum based on the average rank. DMASum [39] shows the best performance on TVSum but does not perform well on SumMe, as indicated in Table 1.

Module	SumMe		TVSum	
	τ	ρ	τ	ρ
GoogleNet (Baseline)	0.176	0.197	0.129	0.163
$(+)$*Attention Module*	0.184	0.205	0.176	0.231
$(+)Att_D$	0.189	0.211	0.182	0.240
$(+)$*Key,Value Embedding*	0.207	0.231	0.193	0.253
$(+)$*Positional Encoding*	0.225	0.251	0.189	0.248
$(+)X_{CLS}$	0.231	0.257	0.193	0.254
$(+)$*Skip Connection*	0.246	0.274	0.194	0.255

Table 2. We listed Kendall's (τ) and Spearman's (ρ) coefficients for different modules. $(+)$ denotes the stacking of modules on top of the previous ones.

DMASum has τ and ρ coefficients of 0.203 and 0.267 on TVSum, respectively, whereas 0.063 and 0.089 on SumMe, respectively. This implies that CSTA provides more stable performances than DMASum, although it provides slightly lower performance than DMASum on TVSum. Based on the overall performance of both datasets, our CSTA has achieved state-of-the-art results.

Further, CSTA excels in video summarization models relying on classical pairwise attention [7, 8, 24, 37, 47], focusing on temporal attention only. This clarifies that considering the visual dependency helps CSTA understand crucial moments by capturing meaningful visual contexts. Like CSTA, some approaches, including DMASum, focus on spatial and temporal dependency [15, 27, 43, 48], but they perform poorly compared to our proposed methodology. This is because CNN is much more helpful than previous methods by using the ability to learn the absolute position in frame features.

CSTA also outperforms methods that require additional datasets from other modalities or tasks [9, 13, 19, 25, 29, 46]. Our observations suggest that CSTA can find essential moments in videos solely based on images without assistance from extra data. We also show the visualization of generated summary videos from different models in Appendix B.

4.4. Ablation Study

This study verifies all components step-by-step, as indicated in Table 2. We deploy an attention structure with GoogleNet and a classifier for temporal dependency, denoted as the $(+)$*Attention Module*. With the assistance of the weighted values from CNN, there is a 0.008 increment on SumMe and at least 0.047 on TVSum, showing the power of CNN as attention. $(+)Att_D$ is the result obtained using softmax along the time and dimension axis to reflect the spatiotemporal importance. The improvement from 0.005 to 0.009 in both datasets indicates that considering the spatial importance is meaningful. The *Key* and *Value Embeddings* strengthen CSTA as a linear projection based

on [38]. Although the $(+)$*Positional Encoding* reveals a small performance drop of 0.004 for τ coefficient and 0.005 for ρ coefficient on TVSum, the performance increases significantly from 0.207 to 0.225 for τ coefficient and from 0.231 to 0.251 for ρ coefficient on SumMe. $(+)X_{CLS}$ is the result obtained when utilizing the CLS token. Because this study combines the CLS token with adaptive pooling, the CLS token only affects a few video frames. However, adding the CLS token improves the performance on both datasets because it generalizes the classifier, which contains fully connected layers. We also see the effects of skip connection, denoted by $(+)$*Skip Connection*, as suggested by [14]. The skip connection exhibits a similar performance on TVSum and an improvement of about 0.015 on SumMe.

We also tested different CNN models as the baseline in Appendix C, various experiments of detailed construction of our model in Appendix D, and several hyperparameters in Appendix E.

4.5. Computation Comparison

Method	SumMe			TVSum		
	Rank	*FE*	*SP*	Rank	*FE*	*SP*
DSNet-AF[47]^{T}	16	413.03G	1.18G	13.5	661.83G	1.90G
DSNet-AB[47]^{T}	13.5	413.03G	1.29G	15	661.83G	2.07G
VJMHT[24]^{T}	8.5	413.03G	18.21G	17.5	661.83G	28.25G
VASNet[7]^{T}	6	413.03G	1.43G	9	661.83G	2.30G
RR-STG[48]ST	2.5	54.82T	0.31G	7.5	88.41T	0.20G
MSVA[9]^{M}	3.5	13.76T	3.63G	5.5	22.08T	5.81G
SSPVS[25]^{M}	3	413.49G	20.72G	4.5	662.46G	44.22G
CSTAST	1	413.03G	9.78G	2	661.83G	15.73G

Table 3. Comparison of MACs between video summarization models. Rank is the average rank between Kendall's and Spearman's coefficients in Table 1. *FE* is the MACs during feature extraction, and *SP* is that during score predictions. We categorize models as temporal attention-based (T), spatiotemporal (ST) attention-based, and multi-modal based (M) models.

In this paper, we analyze the computation burdens of video summarization models, focusing on the feature extraction and score prediction steps. The standard procedure for creating summary videos comprises feature extraction, score prediction, and key-shot selection. Feature extraction is a necessary step in converting frames into features using pre-trained models so that video summarization models can take frames of videos as inputs. Score prediction is the step in which video summarization models infer the importance score for videos. Existing studies generally use the same key-shot selection process based on the knapsack algorithm to determine important video segments, so we ignore computations of key-shot selection.

Table 3 displays MACs measurements and compares the computation resources during the inference per video. CSTA performs best with relatively fewer MACs than the other video summarization models. Based on the average rank from Table 1, more computational costs or supplemental data from other modalities is inevitable for better video summarization performance. Unlike previous approaches, CSTA exhibits high performance with fewer computational resources by exploiting CNN as a sliding window.

We find that our model is more efficient than previous ones when considering spatiotemporal contexts. RR-STG [48] shows much fewer MACs than CSTA during score predictions; however, it shows exceptionally more MACs during feature extraction than others. RR-STG utilizes feature extraction steps for visual relationships by inputting each frame into the object detection model [32], thereby, relying heavily on the pre-processing steps. While summarizing the new videos, RR-STG needs significant time to get spatial associations even though the score prediction takes less time. Other methods [15, 27, 39, 43] design two modules to reflect spatial and temporal dependency, respectively, as shown in Figure 1a and Figure 1b. These approaches become costly when processing numerous frames in long videos for video summarization. CSTA effectively captures spatiotemporal importance in one way using CNN, as illustrated in Figure 1c. Thus, our proposed method shows superior performance by focusing on temporal and visual importance.

5. Conclusion

This study addresses the problem of attention in video summarization. The existing pairwise attention-based video summarization mechanisms fail to account for visual dependencies, and prior research addressing this issue involves significant computational demands. To deal with the same problem efficiently, we propose CSTA, in which a CNN's ability is used for video summarization for the first time. We also verify that the CNN works on frame features and creates attention maps. The strength of the CNN allows CSTA to achieve state-of-the-art results based on the overall performance of two popular benchmark datasets with fewer MACs than before. Our proposed model even outperforms multi-modal or external dataset-based models without additional data. For future work, we suggest further exploring how CNN affects video representations by tailoring frame feature-specific CNN models or training feature-extraction and attention-based CNN models. We believe this study can encourage follow-up research on video summarization and other video-related deep-learning studies.

Acknowledgements. This work was supported by Korea Internet & Security Agency(KISA) grant funded by the Korea government(PIPC) (No.RS-2023-00231200, Development of personal video information privacy protection technology capable of AI learning in an autonomous driving environment)

References

[1] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Combining global and local attention with positional encoding for video summarization. In *2021 IEEE international symposium on multimedia (ISM)*, pages 226–234. IEEE, 2021. 1, 2, 4, 5, 3

[2] Evlampios Apostolidis, Georgios Balaouras, Vasileios Mezaris, and Ioannis Patras. Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 407–415, 2022. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[4] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *The Eleventh International Conference on Learning Representations*, 2022. 3, 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[6] 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*, 2020. 3, 4, 6

[7] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*, pages 39–54. Springer, 2019. 1, 2, 3, 5, 7, 8

[8] Hao Fu, Hongxing Wang, and Jianyu Yang. Video summarization with a dual attention capsule network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 446–451. IEEE, 2021. 7

[9] Junaid Ahmed Ghauri, Sherzod Hakimov, and Ralph Ewerth. Supervised video summarization via multiple feature sets with parallel attention. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6s. IEEE, 2021. 3, 7, 8, 1

[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 1

[11] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12175–12185, 2022. 3

[12] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 505–520. Springer, 2014. 2, 6

[13] Bo He, Jun Wang, Jielin Qiu, Trung Bui, Abhinav Shrivastava, and Zhaowen Wang. Align and attend: Multimodal summarization with dual contrastive losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14867–14878, 2023. 5, 7, 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3, 5, 6, 8

[15] Tzu-Chun Hsu, Yi-Sheng Liao, and Chun-Rong Huang. Video summarization with spatiotemporal vision transformer. *IEEE Transactions on Image Processing*, 2023. 1, 3, 5, 7, 8

[16] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? In *International Conference on Learning Representations*, 2019. 2, 3, 5

[17] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1709–1717, 2019. 1, 2

[18] Zhong Ji, Yuxiao Zhao, Yanwei Pang, Xi Li, and Jungong Han. Deep attentive video summarization with distribution consistency learning. *IEEE transactions on neural networks and learning systems*, 32(4):1765–1775, 2020. 1

[19] Hao Jiang and Yadong 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*, pages 16388–16398, 2022. 7

[20] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *European Conference on Computer Vision*, pages 167–183. Springer, 2020. 2

[21] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020. 2, 3, 5

[22] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. 6

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[24] Haopeng Li, Qiuhong Ke, Mingming Gong, and Rui Zhang. Video joint modelling based on hierarchical transformer for co-summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3904–3917, 2022. 2, 7, 8, 1

[25] Haopeng Li, Qiuhong Ke, Mingming Gong, and Tom Drummond. Progressive video summarization via multimodal self-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5584–5593, 2023. 3, 7, 8, 1

[26] Ping Li, Qinghao Ye, Luming Zhang, Li Yuan, Xianghua Xu, and Ling Shao. Exploring global diverse attention via

pairwise temporal relation for video summarization. *Pattern Recognition*, 111:107677, 2021. 1, 2

[27] Guoqiang Liang, Yanbing Lv, Shucheng Li, Xiahong Wang, and Yanning Zhang. Video summarization with a dual-path attentive network. *Neurocomputing*, 467:1–9, 2022. 1, 3, 7, 8

[28] 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. 1

[29] Medhini Narasimhan, Anna Rohrbach, and Trevor Darrell. Clip-it! language-guided video summarization. *Advances in Neural Information Processing Systems*, 34:13988–14000, 2021. 2, 7

[30] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7596–7604, 2019. 6

[31] 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. 6

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3, 8

[33] 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. 3, 6

[34] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015. 2, 6

[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3, 4, 5, 6, 7, 1

[36] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3, 6

[37] Hacene Terbouche, Maryan Morel, Mariano Rodriguez, and Alice Othmani. Multi-annotation attention model for video summarization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3142–3151, 2023. 6, 7, 1

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1, 3, 4, 5, 8

[39] Junyan Wang, Yang Bai, Yang Long, Bingzhang Hu, Zhenhua Chai, Yu Guan, and Xiaolin Wei. Query twice: Dual mixture attention meta learning for video summarization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4023–4031, 2020. 1, 3, 7, 8

[40] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22–31, 2021. 3

[41] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 579–588, 2021. 3

[42] 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. 3, 6, 7, 1

[43] Yunzuo Zhang, Yameng Liu, Weili Kang, and Ran Tao. Vssnet: Visual semantic self-mining network for video summarization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 1, 3, 7, 8

[44] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 863–871, 2017. 1

[45] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7405–7414, 2018. 7

[46] Bin Zhao, Maoguo Gong, and Xuelong Li. Hierarchical multimodal transformer to summarize videos. *Neurocomputing*, 468:360–369, 2022. 1, 7

[47] 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. 2, 3, 7, 8, 1

[48] Wencheng Zhu, Yucheng Han, Jiwen Lu, and Jie Zhou. Relational reasoning over spatial-temporal graphs for video summarization. *IEEE Transactions on Image Processing*, 31: 3017–3031, 2022. 1, 3, 7, 8

[49] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999. 6