electronics

MDPI

Article

SUM-GAN-GEA: Video Summarization Using GAN with Gaussian Distribution and External Attention

Qinghao Yu [1], Hui Yu [1,2,*], Yongxiong Wang [1] and Tuan D. Pham [3]

[1] School of Control Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
[2] School of Creative Technologies, University of Portsmouth, Portsmouth PO1 2DJ, UK
[3] Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar P.O. Box 1664, Saudi Arabia
* Correspondence: hui_yu@ieee.org

Abstract: Video summarization aims to generate a sparse subset that is more concise and less redundant than the original video while containing the most informative parts of the video. However, previous works ignore the prior knowledge of the distribution of interestingness of video frames, making it hard for the network to learn the importance of different frames. Furthermore, traditional models alone (such as RNN and LSTM) are not robust enough in capturing global features of the video sequence since the video frames are more in line with non-Euclidean data structure. To this end, we propose a new summarization method based on the graph model concept to learn the feature relationship connections between video frames, which can guide the summary generator to generate a robust global feature representation. Specifically, we propose to use adversarial learning to integrate Gaussian distribution and external attention mechanism (SUM-GAN-GEA). The Gaussian function is a priori mapping function that considers the distribution of the interestingness of actual video frames and the external attention can reduce the inference time of the model. Experimental results on two popular video abstraction datasets (SumMe and TVSum) demonstrate the high superiority and competitiveness of our method in robustness and fast convergence.

Keywords: video summarization; external attention mechanism; Gaussian distribution; GAN; graph model; video abstraction

Citation: Yu, Q.; Yu, H.; Wang, Y.; Pham, T.D. SUM-GAN-GEA: Video Summarization Using GAN with Gaussian Distribution and External Attention. *Electronics* **2022**, *11*, 3523. https://doi.org/10.3390/electronics11213523

Academic Editor: Gemma Piella

Received: 18 September 2022
Accepted: 23 October 2022
Published: 29 October 2022

1. Introduction

Video plays an increasingly important role in our daily life. The volume of video hosting platforms, such as YouTube, Twitter, and Instagram, has increased dramatically in an exponential explosion. In July 2015, it was revealed by YouTube that it received over 400 h of video content per minute, which is equivalent to 65.7 years of content uploads per day [1]. With the development of Internet technology in recent years, social media is almost inundated with video data [2]. With the increase of video data, there is an urgent need for an automated technique that could compress a video and allow for the rapid viewing of videos to solve this problem. This involves the technique of video summarization, which generates a concise summary that conveys the important parts of the full video. Video summarization can be used in various applications, such as video detection in football matches [3], smart surveillance [4], and online video management [5]. Video summaries include static video summaries and dynamic video summaries. A static digest contains a subset of frames from the original video but does not retain its temporal component. Dynamic summarization has a subset of shots in the input video, preserving its temporal component [6]. Video summarization technology further improves the consumption potential of video and accelerates the efficiency of information transmission.

Deep learning-based video summarization techniques can be generally classified into three categories, including supervised, weakly supervised, and unsupervised. Among them, unsupervised techniques have attracted more attention in recent years because they do not require manual external intervention (making expensive labels). In recent years, the

rapid development of deep learning technologies has achieved success in many fields of application, such as human trajectory prediction [7], sheet metal bending [8], video anomaly detection and classification [9], real-time target detection and tracking in video surveillance systems [10], multi-view learning methods [11], arterial blood pressure prediction [12], the automatic screening of COVID-19 [13], smart parking systems [14], the generation of 3D geometric objects [15], and the estimation of human food volume [16].

With the improvement of science and technology, more and more people tend to use camera devices to capture high-definition videos to record interesting events in their lives. However, it is a challenge for video summarization tasks. Because the higher definition of video images means that the number of neurons in the layers of the neural network has to change accordingly to match the changes generated by the input, causing the complexity of the model and the inference time to also increase. Therefore, we applied an external attention [17] module to solve this problem. It is based on the traditional attention mechanism but its complexity is O(N), while our baseline model [18] uses the traditional attention mechanism (complexity is $O(N^2)$). It can easily replace the attention mechanism and be embedded in other networks. It is based on two external, smaller, learnable, shared memory units, and they only need two tandem and consecutive linear layers to be implemented. It achieves a reduction of the video feature matrix by replacing the number of pixels of the feature map with a smaller hyperparameter. The lightweight core idea of external attention is that the number of elements in memory is much smaller than the number of elements in the input features, so that the number of elements in the input is linearly related to the computational complexity. Thus, significantly reducing the inference time of the model and lowering the computational cost. In addition, the introduction of an external attention module provides the possibility of the online, real-time use of the model.

The basic building blocks of a video are frames and sequences composed of individual frames are combined to form a complete video. Previous research [18,19] focused on modelling video summarization as an LSTM-based sequence model, which performed better than the traditional RNN sequence model in capturing global features of videos. However, it is not able to capture the global features of videos with longer sequences. In addition, data-like video frames are more in line with non-Euclidean structured data, as there are also intrinsic feature associations between frames to consider compared to traditional single picture data. Indeed, sequences can also be regarded as a special case of graphs, where only adjacent items are connected [20]. Therefore, this paper regards the video frame data as a graph data structure, where each frame represents a node and the edges between nodes represent the connections between frames. The importance score data after mapping the Gaussian distribution is also considered as a graph data structure, where each frame's score is a graph node, and the edges between nodes represent the weights between the scores. The weight relationship between the frame importance scores is learned through the technique of deep learning. The graph model is based on the information propagation mechanism and by iteratively using the local transfer function to update the hidden states of all nodes. The nodes and the feature information of the edges in the final output layer are aggregated with the local and global information of the input layer (the first layer), Therefore, the information of one node in the final output layer of GCN covers all the attributes of the whole graph with great probability, so it can better extract the more attractive frames in the video compared with the LSTM model.

The frame importance score represents the importance (interestingness) of each frame of the video. The frame selector in the previous study outputs an approximately uniform distribution of scores and this distribution may lead to ineffective feature learning. Because the uniform distribution differs significantly from the distribution of interestingness of the actual video frames, we assume that the important frames of the video satisfy the prior distribution of the Gaussian distribution. The uniform distribution indicates that the model considers the interestingness of each frame to be the same, and the entropy of this system is the maximum and so is the degree of confusion. For example, it is difficult to predict the interestingness of each frame. That is, if the interestingness of video frames obeys a

uniform distribution prior, it indicates that no useful information is learned from this prior distribution. We therefore propose for the first time a hypothesis that takes into account the practical situation: the frame importance scores satisfy a Gaussian function. In a given video, we consider that the attractive clips only account for a small portion of the video frames. We force the original uniform distribution to be mapped to a Gaussian distribution, aiming to approximate the interestingness space of the actual video frames, so that local as well as global features can be better learned to preserve the original content of the video.

Loss function plays a key role in the performance of many tasks, such as image restoration [21], semantic segmentation [22], object detection [23], etc. Since the baseline model's MSE loss output value is very small along with the corresponding partial derivative, which can lead to the problem of gradient disappearance. Furthermore, the MSE loss is less robust to outliers. To this end, we adopt SmoothL1Loss in this paper to solve the above two problems. It improves the robustness of our unsupervised model and the convergence of the loss function to facilitate the training of the network.

The main contributions of the proposed method can be summarized as follows:

(1) We consider for the first time the distribution of interestingness of frames in real videos and propose to use a Gaussian function as a priori function for the interestingness of video frames, using it to learn frames importance scores, which better extract the interesting segments of the video.
(2) We use the idea of graph model to learn the feature relationship connections between video frames to guide the summary generator to generate better global feature representations.

The rest of this article is organized as follows: Section 2 summarizes the related work, and Section 3 describes our proposed SUM-GAN-GEA model. Section 4 presents the experimental results, and Section 5 presents the conclusions and outlook.

2. Related Works

2.1. Video Summarization Based on LSTM Sequence Model

Recently, video summarization methods based on LSTM sequence models have received increasing attention. Zhang et al. [24] proposed treating the video summarization task as a structured prediction problem, which used long short-term memory (LSTM) to model the variable-range temporal dependencies between video frames. Video summarization models with two vsLSTM and dppLSTM deep networks were proposed in [25], which allowed the modeling of frame correlations and similarities. Elfeki et al. [26] took a different approach by analyzing the main cues humans use for summary generation; the nature and intensity of actions and developed a method to integrate action data to explicitly specify the learning algorithms trained for summary generation. Satorras et al. [27] introduced a new architecture for multivariate time series forecasting that simultaneously infers and exploits relationships between time series, which was flexible enough to be extended according to the forecasting task under consideration. However, although LSTM is an improvement on the RNN in extracting global features, it is still not sufficient for some long sequences of videos. Since our baseline model uses the same LSTM approach to model the video summarization task, it also has this inherent disadvantage. Recent advances show that for global feature capture, better performance can be obtained using graph convolutional neural networks [28,29]. We thus introduce the idea of graph data structures in this paper. This is because graph networks are not only capable of modelling time-series problems, but the video frame data is also more in line with the non-Euclidean data structure. It is similar to a social network, where each frame is connected to others.

2.2. Complexity of Video Summarization

The high complexity of video summarization models is still a challenging problem and a large body of work has been proposed in recent years. For example, Ou et al. [30] proposed an online summarization method for video summarization using a Gaussian mixture model that produces summaries with shorter latency and lower computational resource

requirements. Furthermore, when outputting the summaries, the similarity between two video sequences S1 and S2 needs to be computed by calculating and averaging all the differences between frames in the two sequences. The complexity of this comparison mechanism is $O(N^2)$, where N is the number of frames in each comparison sequence. Therefore, to keep the computational complexity low, the number of frames in the video sequence has to be kept at a small value [31]. Ma et al. [32] developed a non-linear sparse dictionary selection model by mapping the high-dimensional data of the original video samples to the non-linear case using a kernel function to transform it into the linear case. A kernelised simultaneous block orthogonal matching pursuit (KSBOMP) method is designed to optimize the proposed model and reduce the computational complexity of the model. However, it has high complexity due to its nonlinearity.

2.3. Gaussian for Video Summarization

Many summary methods based on Gaussian functions have been developed. In SIFT [33], significant locations are first defined using the scale space of smoothed and resized images and a difference of Gaussian functions is applied to these images to find the maximum and minimum responses. Furthermore, the partial spatial derivatives of an image are usually estimated using a Gaussian filter [34]. Zhang et al. [35] proposed a novel video summarization method using the video frame similarity function (VFSF) and a Gaussian approach to summarize the video, similar to the Gaussian approach and the concept of extrema applied to extract the key frames from each clip to form a video summary. In this paper, we also propose a method of mapping a Gaussian function on the frame importance scores so that the mapped scores are closer to the interestingness space of the actual video frames to obtain a summary. Mahasseni et al. [36] combined an LSTM-based keyframe selector with a variational autoencoder for the first time and used the decoder as the input of a trainable discriminator to learn summarization in an adversarial manner, aiming to minimize the distance between the reconstructed video output by the generator and the original video. Following [36], Apostolidis et al. [19] employed linear layers to compress the parameters and proposed a step-by-step, label-based learning process to improve the training efficiency of the adversarial part of the model. As a further improvement, Apostolidis et al. [18] integrated the attention mechanism into the variable autoencoder of the original system, which made the training of the model more stable and significantly improved the performance.

3. The Proposed Approach

In this section, we describe our proposed SUM-GAN-GEA model. Our baseline is a generative adversarial network (GAN) model derived from [18]. GAN can better minimize the distance between the generated summary features and the original video features. The proposed method has the advantages of lower linear memory complexity and a more sensitive capture of keyframes. Figure 1 depicts the structure of SUM-GAN-GEA. The general framework of our SUM-GAN-GEA model is shown in Figure 1.

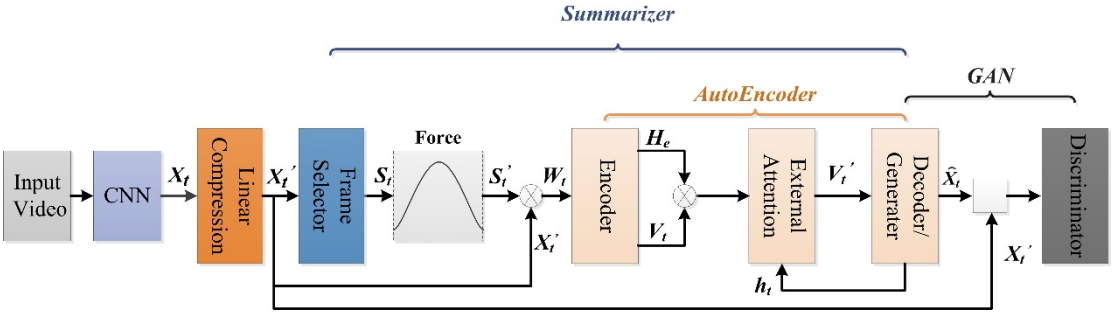

Figure 1. Structure of SUM-GAN-GEA.

The key idea of this method is to select summaries by minimizing the distance between the deep feature representation of the original video and the reconstructed importance of the selected keyframes. We model the video summarization problem as a Seq2Seq problem, where the input is a sequence of the raw video frames and the output is a sequence of summaries of key frame combinations. The framework shown in Figure 1 is mainly composed of a linear compression layer, a frame selector, an autoencoder with an external attention mechanism module, and a discriminator comprising four components.

Given an input video with T frames, feature extraction is based on the pool5 layer of GoogLeNet trained on ImageNet and x_t is the depth feature of the frame sequence of the original video being extracted. After a linear compression layer reducing the complexity of the model, $x_t{}'$ is obtained, which is then used as the input of the frame selector to obtain the output s_t, t $\in [1, \mathrm{T}]$, where s_t represents the importance score calculated by the frame selector and the value range is (0, 1). Then the frame importance score is forced to follow the Gaussian distribution. W_t is the weighted feature vector of $x_t{}'$ and $s_t{}'$, indicates that the importance of each frame is different. Taking W_t as the input of the encoder in the Encoder-Decoder module, after encoding by the encoder, the output coding vector V_t and the last hidden state H_e are obtained and are then dot-multiplied to obtain a feature vector focusing on the tail feature. Together with the initial state h_t input of the decoder, it is sent to the external attention mechanism module in order to obtain the global semantic vector $V_t{}'$, which is then input into the decoder (generator) to obtain the reconstructed feature vector $\hat{x}_t$. They are finally fed into the discriminator along with the original video for identification.

3.1. External Attention Mechanism

To alleviate the square-level complexity and high memory consumption, we introduce an external attention mechanism module. Since the upper module of the external attention mechanism is the encoder of LSTM, to obtain the deep semantic features between video sequences, we do not discard the hidden state of LSTM but pass the hidden state H_e through a small linear neural network. The obtained output is a Hadamard product with the output V of the encoder, which is used as the input of the external attention mechanism. The structure diagram of the external attention mechanism is shown in the dotted box in the right block of Figure 2. The dotted box on the left is the input change caused by the external attention module. The external attention mechanism algorithm can be expressed using the following formula:

$$EA = Norm(F * M_k^T) \tag{1}$$

$$Output = EA * M_V \tag{2}$$

where F is the given input feature map. The size of F is [seq, batch size, hidden size], where seq represents the sequence length of the input feature, batch size is the same with the batch size of datasets, and hidden size is the same with the size of the hidden layer units of the encoder. In addition, M_k is a learnable parameter independent of input features, which is equivalent to a memory unit for the entire dataset. Norm indicates normalization. The asterisk "*" indicates matrix multiplication. The M_k size of is [d, S]. It does not have a bias term. d here is the feature dimension of the video sequence, and S is a hyperparameter, which is set it as 2. Then we normalize the matrix product on the first dimension by softmax, followed by dividing the normalized tensor by the original tensor to sum on the second dimension and keep the original dimension. This operation is equivalent to the first two normalization shown in Equation (1). Finally, we pass it through a small linear neural network layer M_v of size [S, d], which is also unbiased, where M_v is the second memory unit that can learn parameters. The output of the former is normalized with M_v conducting the matrix product using Equation (1).

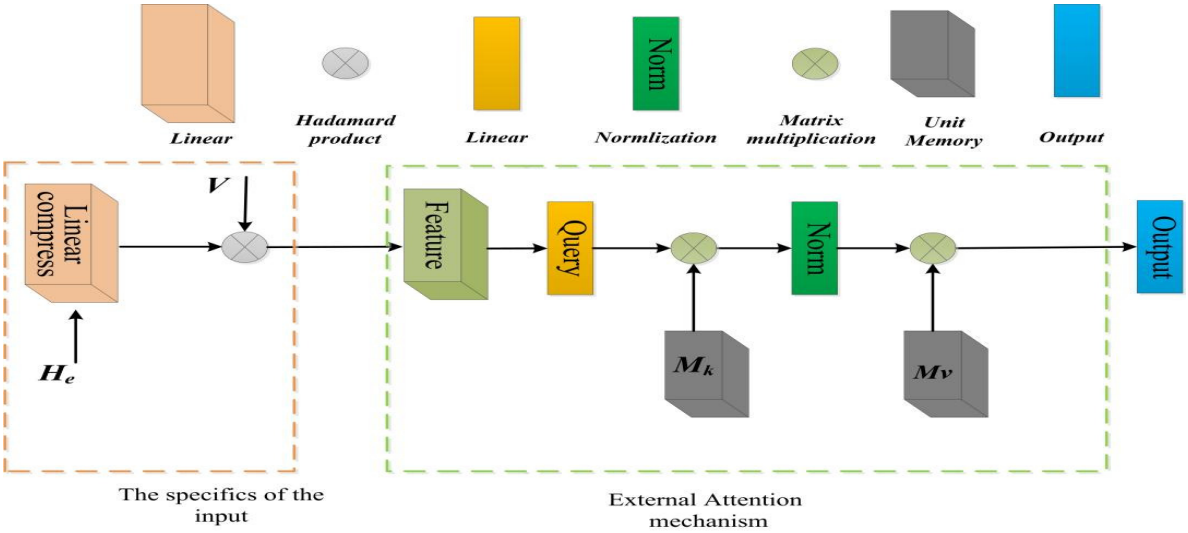

Figure 2. The dotted box on the left is the input change caused by the replacement of the attention mechanism. The dotted box on the right is the external attention mechanism module.

Through applying this external attention module, the computational complexity can be reduced from O(d $*$ N^2) to O(d $*$ S $*$ N), where N is the number of pixels of the feature map, S is a relatively small hyperparameter, and d is a high-dimensional input feature. The complexity is significantly reduced. The structure in Figure 2 is shown in Pseudocode 1.

Pseudocode 1. The structure pseudocode depicted in Figure 2

```
h: the hidden state of the encoder
linear: linear layer
attn_en: attention energy vector
*: hadamard product
enc_out: the original output of the encoder
enc_out': the final output of the encoder
enc_out_q: The output of "enc_out" after the Query linear layer
query: linear layer
mk: linear memory unit
attn: attention vector
softmax: softmax activation function
torch: A common library in the field of deep learning
mv: linear memory unit
out: the output of external attention module

h = h.transpose(0,1)                          # shape = [num_layers,batch_size, hidden_size]
h = h.reshape(h.shape [0], −1)                # shape = [batch_size, num_layers × hidden_size]
attn_en = linear(h)                           # shape = [batch_size, hidden_size]
attn_en = attn_en.unsqueeze(0)                # shape = [1,batch_size, hidden_size]
enc_out' = attn_en * enc_out                  # shape = [seq,batch_size,num_layers]
enc_out_q = query(enc_out')                   # shape = [seq,batch_size,hidden_size]
attn = mk(enc_out_q)                          # shape = [seq,batch_size,num_layers]
attn = softmax(attn,dim = 1)                  # shape = [seq,batch_size,num_layers]
attn = attn/torch.sum(attn,dim = 2,keepdim = True)   # shape = [seq,batch_size,num_layers]
out = mv(attn)                                # shape = [seq,batch_size,num_layers]
```

3.2. Gaussian Distribution of Frames Importance Scores

Jadon et al. [33] used Gaussian clustering and chose the number of clusters according to the size of the video summary, making the summary smoother and easier to understand. Jiang et al. [37] proposed a new Gaussian mixture vector quantization (GMVQ) method to

summarize video content with optimal compression. Both posterior and prior distributions in variational autoencoders followed Gaussian distributions, and a Gaussian mixture model was used to approximate the true distribution of the dataset. In addition, the posterior and prior distributions in variational autoencoders are Gaussian. Furthermore, considering that the exciting parts of the videos tend to have very little drama and are usually concentrated in a certain area, most of them are quite bland. Based on the above inspiration, we assume that the distribution of importance scores of video frames follows the prior distribution of the Gaussian function.

In addition, experiments show that the importance score of the frame selector output approximately meets the uniform distribution. We propose using the importance score of Gaussian distribution for two reasons: (1) Uniform distribution cannot represent the interest degree of the actual video frame well; that is, the importance score should fluctuate with a certain range. (2) We assume that the uniform distribution of importance scores makes the feature learning of common shots in video ineffective, while the global feature information learning becomes more difficult. We perform the following operations on the importance score to make it satisfy the Gaussian distribution: (1) The vector compressed by the convolutional neural network and the linear layer is equivalent to an ordered sequence composed of an undirected graph of t(t $\in$ [1, T]) nodes, and T is the number of frames of the input video. The edge between any two different nodes represents the feature dependency between different frames. (2) Calculate the mean and variance of the importance score of the frame selector output. (3) The importance scores that are force to generate satisfies the Gaussian distribution $F_{(f)} \sim N(\mu, \sigma^2)$, while maintaining the original mean and variance. We obtain the importance score $s_t{}'$, $k \in$ [1, T] of the relearned frame sequence graph, which can be regarded as an unordered diagram of similar frames after the aggregation of Gaussian distribution. Each node in the graph needs to update the hidden state by its neighbor nodes. The Gaussian distribution adjusts the frames and centralizes frames with similar features in a small interval. Furthermore, the gradient transformation of updating the hidden state of neighbor nodes and the information flow of the whole graph is smaller. It can thus better approximate the distribution of frame importance scores in actual video. Because the graph nodes are readjusted, the original global features that are difficult to learn can also be converted into local features. (4) Use the Sigmoid activation function to restrict $s_t{}'$ to the range of (0, 1) to update the original importance scores. The schematic diagram is shown in Figure 3. As Gaussian distribution is a probability model, we assume that it satisfies the frame importance fraction distribution based on the actual situation.

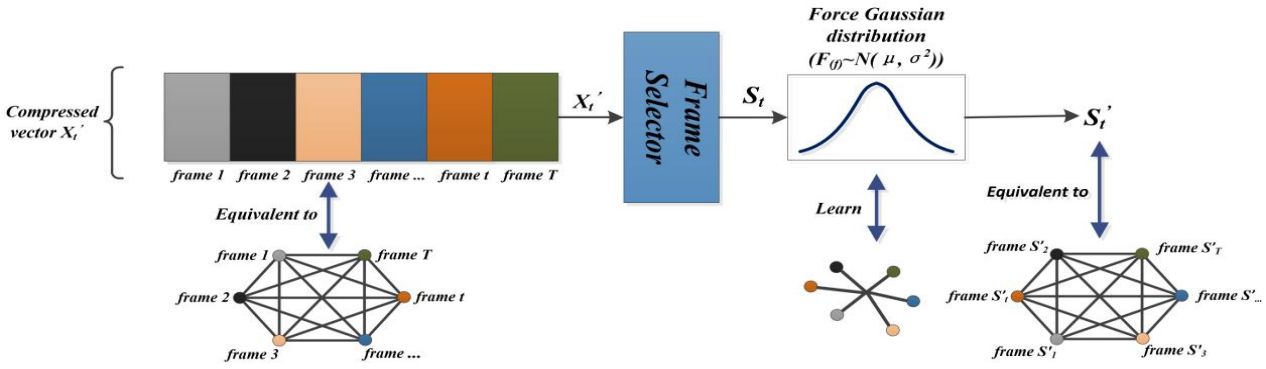

Figure 3. Forces the mapping of the compressed frame sequence importance score graph into an unordered graph with a Gaussian function. Different frames are represented by the rectangles of different colors and correspond to nodes of different colors.

3.3. SmoothL1Loss

We propose using SmoothL1Loss mainly due to the following two reasons: (1) Since the MES loss tends to become zero when the error approaches zero, it can cause the effect of gradient vanishing. In the case of a large error value, the corresponding gradient value

becomes large, which may lead to the problem of gradient explosion. (2) During the shooting process of the video, it may be due to the limitation of the shooting equipment or the problem of loss in the transmission process, etc. These issues may cause some noise points in the video. If MSE is used as the loss, these noises influencing the points will be amplified. When the error is large, the gradient of the MSE is also large. After the back-propagation algorithm, the parameter update is affected by these noise points, which is unfavorable for the entire learning process. The SmoothL1Loss formula is shown as follows:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2/\delta, & |y - f(x)| < \delta \\ |y - f(x)| - \frac{1}{2}\delta, & |y - f(x)| \geq \delta \end{cases} \tag{3}$$

The parameter δ determines the tendency of SmoothL1Loss to MSE and MAE. When the absolute value of $y - f(x)$ is less than δ, it becomes the MSE loss. When the absolute value of $y - f(x)$ is greater than or equal to δ, it becomes the MAE loss. In the small range where the error value belongs to (0, δ), SmoothL1Loss overcomes the issue that the MAE loss is not smooth in the vicinity of the zero point, so that the global optimal solution can be approached at a small rate. When the error is greater than or equal to δ, the loss value of SmoothL1Loss will be smaller than the MSE loss, and it is not very sensitive to outliers. It thus can reduce the influence of noises to a certain extent and improve the robustness of the model. Here, instead of fine-tuning the parameter δ, we adopt the default setting parameter 1.

4. Experiments

We implement our proposed framework on the Pytorch platform using an NVIDIA GeForce 3090 GPU. We do not use the momentum-based Adam optimizer because it could cause the model to fail to converge [38]. We use the non-momentum RMSprop optimizer in our experiments instead. The parameters are set as follows: the smoothing constant α is 0.99, the parameter eps (to prevent the denominator from being 0) is 10^{-8}, and other parameters and training process are set to be consistent with [18].

4.1. Experiment Datasets and Evaluation Metrics

Datasets. We evaluate the performance of the proposed method on two widely used benchmark datasets, SumMe [39] and TVSum [40]. The SumMe dataset contains 25 videos of 1–6 min. With 15–18 user-annotated key segments per video, the dataset provides a single ground-truth summary in the form of frame-level importance scores (by computing the average of user-annotated key segment scores). The TVSum dataset contains 50 videos with durations between 1 and 5 min, capturing 10 categories of the TRECVid multimedia event detection dataset. It also provides a single ground-truth summary of frame-level importance scores (computed by averaging users' scores).

Evaluation Metric. For a fair comparison with other video summarization algorithms, we employ the currently popular F-Score to evaluate the similarity of the model's automatically generated summaries to the ground truth summaries. Automatically generated summaries are denoted by A, and ground truth summaries are denoted by G. Calculate the precision P and recall R of each pair of A and G as a measure of their overlapping time, and the F-Score is the harmonic mean of the two ($\cap$ indicates overlapping time). The calculation is as follows:

$$P = \frac{A \cap G}{A} \times 100\% \tag{4}$$

$$R = \frac{A \cap G}{G} \times 100\% \tag{5}$$

$$F = \frac{2 \times P \times R}{P + R} \times 100\% \tag{6}$$

4.2. Performance Evaluation

4.2.1. F-Score Data Distribution

In the dataset used, the training set accounts for 80% and the remaining 20% is used as the test set. We iteratively trained our model 100 times. Finally, we experiment with five different random splits and report the average performance of these splits. The distribution of F-Score of our model SUM-GAN-GEA on the SumMe and TVSum datasets is shown in Figures 4 and 5.

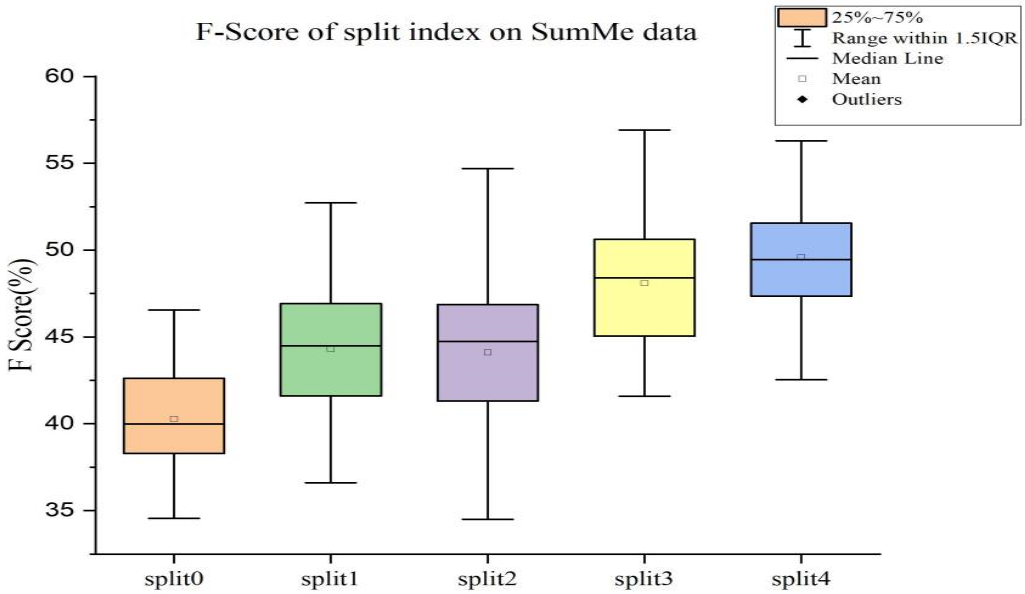

Figure 4. F-Score of 5-fold cross-validation on SumMe dataset.

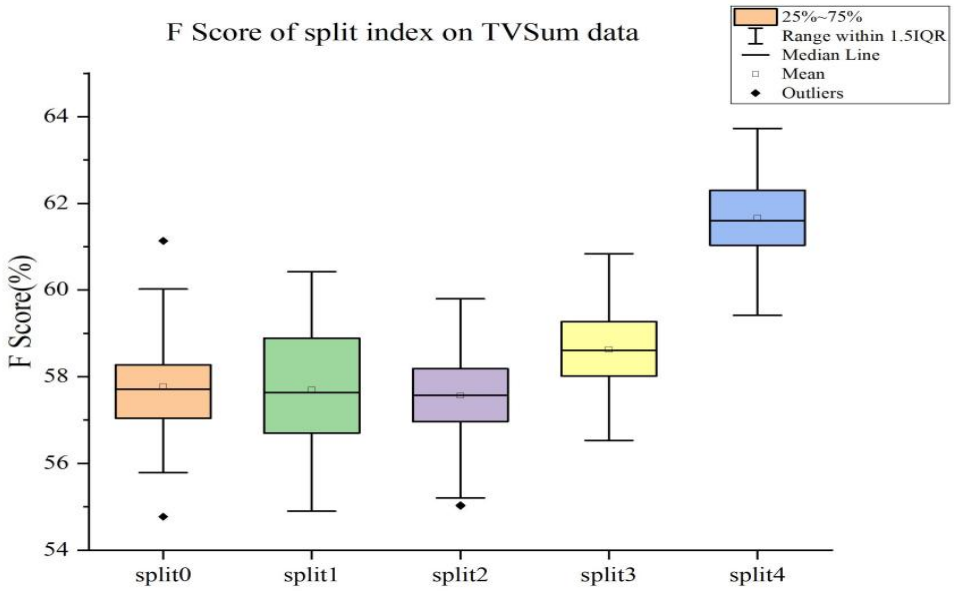

Figure 5. F-Score of 5-fold cross-validation on TVSum data.

Through comparing Figures 4 and 5, we can see that: (1) the distribution of F-Score is more concentrated in the TVSum dataset, while the distribution of F-Scores is more dispersed in the SumMe dataset. (2) On the TVSum dataset, the cross-validation performance of either fold is better than that on the corresponding SumMe dataset. (3) It is worth noting that the minimum value of the F-Score trained on the TVSum dataset is close to the maximum value on the SumMe dataset. The possible reason behind this is that TVSum has

more data and richer video sequence scenes, thus making the model trained on this dataset more complex and robust.

4.2.2. Comparison with Unsupervised Methods

For a fair comparison, we evaluate the quality of summaries generated by our model using two evaluation methods: multiple user-annotated summaries per video and single ground truth summaries. Under the evaluation of the former method, we compare the performance of our model with other state-of-the-art unsupervised methods on SumMe and TVSum, as shown in Table 1. For a better comparison, we add the mean of the F-Score on the two datasets on the right column as Average.

Table 1. The summarization approach using multiple user annotations is compared with other unsupervised methods. ("±" indicates better or worse performance than our method).

Methods	SumMe	TVSum	Average
Tessellation [41]	41.4(−)	**64.1**(+)	52.8(−)
DR-DSN [42]	41.4(−)	57.6(−)	49.5(−)
UnpairedVSN [43]	47.5(−)	55.6(−)	51.6(−)
SUM-Ind$_{LU}$ [44]	51.9(−)	61.5(+)	56.7(−)
AC-SUM-GAN [45]	50.8(−)	60.6(−)	55.7(−)
CAAN [46]	50.8(−)	59.6(−)	55.2(−)
CSNet [47]	51.3(−)	58.8(−)	55.1(−)
SUM-GDA [48]	50.0(−)	59.6(−)	54.8(−)
SUM-GAN-sl [19]	47.3(−)	58.0(−)	52.7(−)
SUM-GAN-AAE [18]	48.9(−)	58.3(−)	53.6(−)
SUM-GAN-GEA (Ours)	**53.4**	**61.3**	**57.4**

From the data presented in Table 1, we can see that the proposed method has obtained the best performance. This demonstrates the superiority of our proposed method. (1) On the SumMe dataset, our method outperforms SUM-Ind$_{LU}$ (ranked second) by 1.5%, and the baseline model SUM-GAN-AAE by 4.5%; (2) on the TVSum dataset, our method also displays a substantial increase of 3% compared with the baseline model SUM-GAN-AAE; (3) the Tessellation [41] method is 2.8% higher than ours on the TVSum dataset but it achieved only 41.4% on the SumMe dataset, which is 12% lower than our method on this dataset. This is because the Tessellation [41] method is a dataset customization technique and thus performs extremely well on the TVSum dataset; (4) our method is 0.7% higher than the SUM-Ind$_{LU}$ method in terms of average metrics but it is 0.2% lower than the SUM-Ind$_{LU}$ method on the TVSum dataset. The possible reason for this is that SUM-Ind$_{LU}$ combined deep reinforcement learning with independent recurrent neural networks (IndRNN) to solve the gradient disappearance between layers and the entanglement between neurons. The model can be trained with more steps and have more layers (in most cases, deeper layers tend to have better results) without any of the problems associated with gradients [44]. However, the network of our model is not as deep as that model. Overall, our proposed method shows good performance on both datasets (especially on SumMe) and is the most competitive.

4.2.3. Evaluation Using a Single Ground-Truth Summarization

We evaluate our model performance using the evaluation method of a single ground-truth summary of each video, as shown in Table 2.

From the data shown in Table 2, we can draw the following conclusions: (1) On the SumMe dataset, our method obtains the best performance, reaching 65.9%, thus far exceeding the original SUM-GAN-AAE (ranked 2nd) model by 9%, and our method outperforms some supervised methods. (2) On the TVSum dataset, we still obtain the best performance (68.2%), outperforming the baseline model SUM-GAN-AAE by 4.3% and surpassing the second-ranked SUM-GAN-sl [19] model by 2.9%. (3) On average, our method outperforms the baseline model by 4.3%. Overall, the proposed method performs

the best (especially on SumMe) using a single ground-truth summary for each video evaluation. (4) The performance of the assessment metric in Table 2 only differs by 2.3% on the two datasets, while it differs by 7.9% using the assessment metric in Table 1. The fact that Table 1 uses multiple user-annotated video summaries makes it more objective, while the "gold standard" of a single ground truth summary used in Table 2 may not be standard due to the subjective nature of the video summarization task. Therefore, it is necessary to use different kinds of evaluation metrics.

Table 2. Comparison with other video summarization methods using single ground truth summarization (asterisks indicate unsupervised methods, "±" indicate better or worse performance than our method).

Methods	SumMe	TVSum
* SUM-GAN [36]	38.7(−)	50.8(−)
* SUM-GAN$_{dpp}$ [36]	39.1(−)	51.7(−)
SUM-GAN$_{sup}$ [36]	41.7(−)	56.3(−)
SASUM [49]	45.3(−)	58.2(−)
DTR-GAN [50]	44.6(−)	59.1(−)
A-AVS [51]	43.9(−)	59.4(−)
M-AVS [51]	44.4(−)	61.0(−)
AALVS [52]	46.2(−)	63.6(−)
* Cycle-SUM [53]	41.9(−)	57.6(−)
* SUM-GAN-sl [19]	46.8(−)	65.3(−)
* SUM-GAN-AAE	56.9(−)	63.9(−)
*** SUM-GAN-GEA (Ours)**	**65.9**	**68.2**

4.3. Ablation Study

In this section, we test the impact of each module of our models. Table 3 shows the performance of each variation on both datasets (running times are also averaged over five cross-validations).

Table 3. Ablation study.

Experiments	SmoothL1Loss	External Attention	Gaussian Distribution	SumMe (F-Score) and Time (s)	TVSum (F-Score) and Time (s)
SUM-GAN-AAE				48.9 and45.2	58.3 and 145.8
Exp1	√			50.1 and 45.4	60.3 and 151.4
Exp2		√		50.5 and 37.8	59.9 and 123.8
Exp3			√	53.4 and 44.6	61.2 and 145.4
Exp4	√	√		51.0 and 37.0	60.5 and 118.8
Exp5	√		√	53.4 and 46.0	61.2 and 146.8
Exp6		√	√	53.4 and 34.6	61.1 and 113
Exp7(SUM-GAN-GEA)	√	√	√	53.4 and 34.0	61.3 and 112.6

From Table 3, we can draw the following conclusions: (1) Only SmoothL1Loss is introduced based on the baseline model [18]. Although the running time on the two datasets increases by 0.44% and 3.8%, the F-Score increases by 2.45% and 3.4%. In addition, since there is a coefficient of 1/2 in Equation (3), to better compare the training situation of the loss function, we multiply the value of the loss function obtained by 2. Regarding the loss function of the discriminator trained in these two datasets, such as those shown in Figure 6 (the loss functions involved in the figure are all in the loss functions involved in the figure are all in [18]) and Figure 7. From Figures 6 and 7, we can see that the fluctuations of the curves of the three loss functions are much smaller at roughly 1000 steps, and the values of the three loss functions converge on 0.25. Therefore, the introduction of SmoothL1loss can assist the training process, which enhances the robustness and convergence speed of the model. (2) The introduction of any single one of these three modules improves the performance of our model, especially the Gaussian distribution of frame importance scores

compared to our baseline model. Our method improves by 4.5% over the SUM-GAN-AAE on the SumMe dataset and 2.9% on the TVSum dataset (see baseline model and Exp3). It proves the validity of our assumption that the prior distribution of frame importance scores is more like a Gaussian distribution. (3) In terms of running time, comparing Exp2 with the baseline model under the premise that the former uses linear complexity, the time is reduced by 16.4% on the SumMe dataset. Comparison between SUM-GAN-AAE and Exp1 shows that the introduction of SmoothL1Loss has a slight increase in runtime (or see Exp6 and Exp7). We can also see that comparing the original model SUM-GAN-AAE [18] with Exp3, the latter yields a slight reduction in time reduction for Gaussian distribution (or see Exp4 and Exp7). Our final model is compared with SUM-GAN-AAE, the time is reduced by 24.8%, which is close to 1/4. It illustrates that the introduction of an external attention module allows the model to run (infer) in much less time, which is significant from both an academic and practical point of view. (4) On the TVSum dataset, the Gaussian distribution still has the greatest impact on the model, which is 2.9% higher than the original model (see Exp3 and the original model). (5) TVSum has a 15.1% reduction in running time (see Exp2 and the baseline model), while our method reduces the time by 22.8% compared to the baseline method [18]. The performance on this dataset shows that it is comparable to the SumMe dataset. (6) By comparing Exp3 and Exp4, we can see that the performance after introducing Gaussian distribution alone is even better than introducing SmoothL1Loss and external attention mechanism. (7) An interesting finding is that as long as the Gaussian distribution module is introduced, the performance of the proposed method does not change much but there is still a big difference in the running time (see Exp3, Exp5, Exp6, and our proposed method).

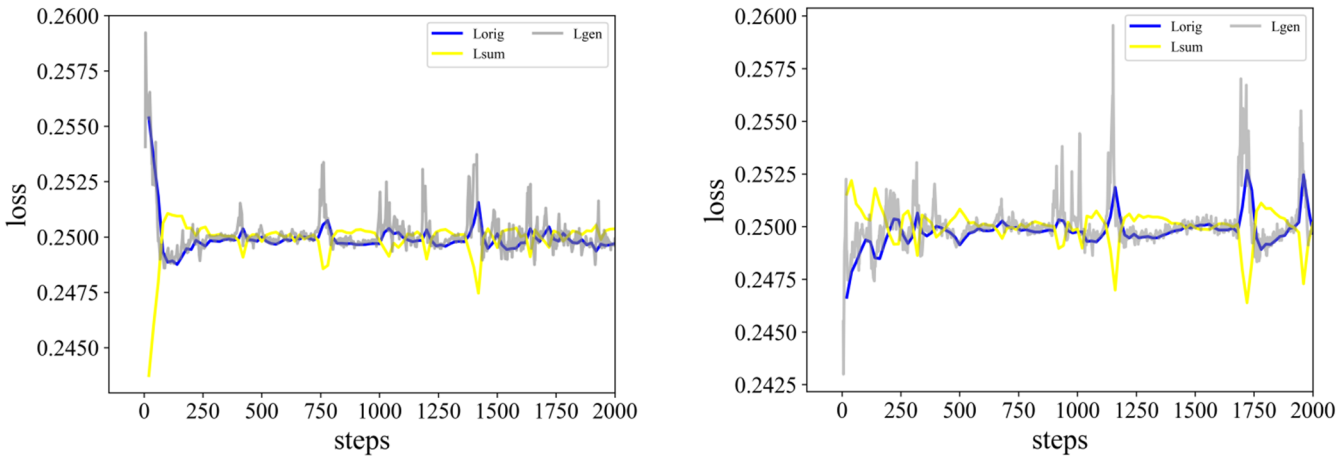

Figure 6. The left and right figures are the loss function curves of the discriminator Exp1 and the baseline model SUM-GAN-AAE on the SumMe dataset, respectively.

The three elements in the network all play important roles. The introduction of SmoothL1Loss enables the faster and smoother convergence of the model. This loss function reduces the influence of noise and enhances the robustness of the model. The addition of the external attention mechanism is very helpful to reduce the training time of the model because the linear complexity of two linear layers is clearly less than the original square-level complexity. The Gaussian distribution of the frame importance score has a significant impact on the model and can also slightly reduce the running time. One possible explanation is that the Gaussian distribution of the frame importance score destroys the original uniform distribution of invalid feature learning. Gaussian distribution is more representative of the importance distribution of actual video frames than uniform distribution. Moderate changes in the magnitude of the frame importance scores can make the generator (summary) better to improve the ability to capture different interesting frames between video frames, so that the generated summaries are more efficient and of better quality and are more expressive than the original video. In addition, the discriminative

ability of the corresponding discriminator has also been improved. Two possible reasons for the above conclusion (7) are: (1) Since we adopt the same gradient clipping settings as in the baseline model [18], although it can alleviate the problems of vanishing and exploding gradients, to some extent it also inhibits the fitting ability of the network, so that the model performance cannot be further improved. (2) Because the number of videos contained in the SumMe and TVSum datasets is small, it is difficult for the model to learn more effective parameters, so the performance of the model stagnates.

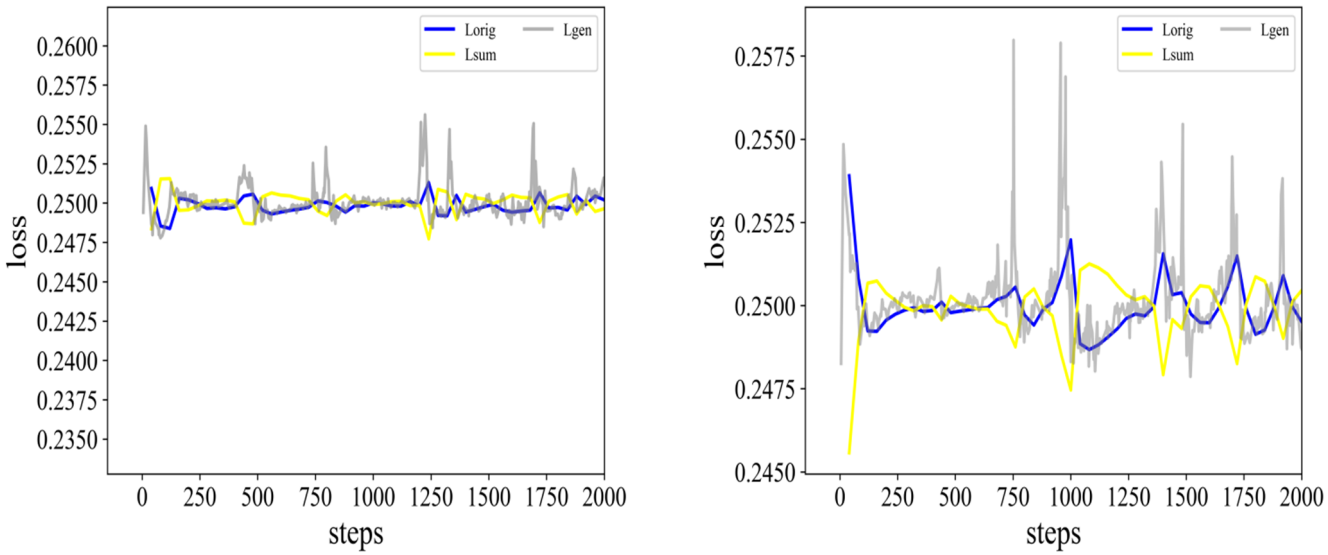

Figure 7. The left and right figures are the loss function curves of the discriminator Exp1 and the baseline model SUM-GAN-AAE on the TVSum dataset, respectively.

5. Conclusions

In this paper, we propose a novel unsupervised video summarization method, which combines the advantages of the external attention mechanism module and the frame importance fractional Gaussian distribution. We learn the feature-relational connections between video frames with the help of graph model ideas to guide the summary generator to generate better global feature representations. We adopt SmoothL1Loss to improve the robustness of the model. Extensive experiments on two mainstream video summarization datasets demonstrate the superiority of our method. Ablation studies demonstrate the effectiveness of each element. Currently, we only study the dependencies between the features of the video frame sequence, while ignoring the intrinsic connection between the basic unit shots that make up the video. In the future, we will enhance the generalization capabilities of the model by mixing data tricks. We will also take advantage of graph networks to handle multi-hop problems in order to explore global dependency features between different shots in a video to improve performance.

Author Contributions: Investigation, Q.Y.; resources, Y.W.; writing—original draft preparation, Q.Y.; writing—review and editing, H.Y. and T.D.P.; supervision, H.Y. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Apostolidis, E.; Adamantidou, E.; Metsai, A.I.; Mezaris, V.; Patras, I. Video summarization using deep neural networks: A survey. *Proc. IEEE* **2021**, *109*, 1838–1863. [CrossRef]
2. Sreeja, M.; Kovoor, B.C. A multi-stage deep adversarial network for video summarization with knowledge distillation. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–16. [CrossRef]
3. Agyeman, R.; Muhammad, R.; Choi, G.S. Soccer Video Summarization using Deep Learning. In Proceedings of the 2nd IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 28–30 March 2019; pp. 270–273.
4. Thomas, S.S.; Gupta, S.; Subramanian, V.K. Smart Surveillance Based on Video Summarization. In Proceedings of the IEEE Region 10 Symposium on Technologies for Smart Cities (TENSYMP), IEEE Kerala Sect, Kochi, India, 14–16 July 2017.
5. Almeida, J.; Leite, N.J.; Torres, R.d.S. VISON: Video Summarization for ONline applications. *Pattern Recognit. Lett.* **2012**, *33*, 397–409. [CrossRef]
6. Nair, M.S.; Mohan, J. VSMCNN-dynamic summarization of videos using salient features from multi-CNN model. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–10. [CrossRef]
7. Li, X.; Liu, Y.; Wang, K.; Wang, F.-Y. A recurrent attention and interaction model for pedestrian trajectory prediction. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 1361–1370. [CrossRef]
8. Liu, S.; Xia, Y.; Shi, Z.; Yu, H.; Li, Z.; Lin, J. Deep learning in sheet metal bending with a novel theory-guided deep neural network. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 565–581. [CrossRef]
9. Mansour, R.F.; Escorcia-Gutierrez, J.; Gamarra, M.; Villanueva, J.A.; Leal, N. Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model. *Image Vis. Comput.* **2021**, *112*, 104229. [CrossRef]
10. Alotaibi, M.F.; Omri, M.; Abdel-Khalek, S.; Khalil, E.; Mansour, R.F. Computational Intelligence-Based Harmony Search Algorithm for Real-Time Object Detection and Tracking in Video Surveillance Systems. *Mathematics* **2022**, *10*, 733. [CrossRef]
11. Yan, X.; Hu, S.; Mao, Y.; Ye, Y.; Yu, H. Deep multi-view learning methods: A review. *Neurocomputing* **2021**, *448*, 106–129. [CrossRef]
12. Paviglianiti, A.; Randazzo, V.; Villata, S.; Cirrincione, G.; Pasero, E. A Comparison of Deep Learning Techniques for Arterial Blood Pressure Prediction. *Cogn. Comput.* **2021**, *14*, 1689–1710. [CrossRef]
13. Goel, T.; Murugan, R.; Mirjalili, S.; Chakrabartty, D.K. Automatic screening of covid-19 using an optimized generative adversarial network. *Cogn. Comput.* **2021**, 1–16. [CrossRef] [PubMed]
14. Ali, G.; Ali, T.; Irfan, M.; Draz, U.; Sohail, M.; Glowacz, A.; Sulowicz, M.; Mielnik, R.; Faheem, Z.B.; Martis, C. IoT Based Smart Parking System Using Deep Long Short Memory Network. *Electronics* **2020**, *9*, 1696. [CrossRef]
15. Park, S.; Kim, H. FaceVAE: Generation of a 3D Geometric Object Using Variational Autoencoders. *Electronics* **2021**, *10*, 2792. [CrossRef]
16. Yang, Z.; Yu, H.; Cao, S.; Xu, Q.; Yuan, D.; Zhang, H.; Jia, W.; Mao, Z.-H.; Sun, M. Human-Mimetic Estimation of Food Volume from a Single-View RGB Image Using an AI System. *Electronics* **2021**, *10*, 1556. [CrossRef]
17. Guo, M.-H.; Liu, Z.-N.; Mu, T.-J.; Hu, S.-M. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv* **2021**, arXiv:2105.02358. [CrossRef]
18. Apostolidis, E.; Adamantidou, E.; Metsai, A.I.; Mezaris, V.; Patras, I. Unsupervised video summarization via attention-driven adversarial learning. In *International Conference on Multimedia Modeling*; Springer: Cham, Switzerland, 2020; pp. 492–504.
19. Apostolidis, E.; Metsai, A.I.; Adamantidou, E.; Mezaris, V.; Patras, I. A stepwise, label-based approach for improving the adversarial training in unsupervised video summarization. In Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery, Nice, France, 21 October 2019; pp. 17–25.
20. Zhao, B.; Li, H.; Lu, X.; Li, X. Reconstructive Sequence-Graph Network for Video Summarization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2793–2801. [CrossRef]
21. Zhao, H.; Gallo, O.; Frosio, I.; Kautz, J. Loss functions for image restoration with neural networks. *IEEE Trans. Comput. Imaging* **2016**, *3*, 47–57. [CrossRef]
22. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Via del Mar, Chile, 27–29 October 2020; pp. 1–7.
23. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
24. Zhang, K.; Chao, W.-L.; Sha, F.; Grauman, K. Video summarization with long short-term memory. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 766–782.
25. Lebron Casas, L.; Koblents, E. Video summarization with LSTM and deep attention models. In *International Conference on MultiMedia Modeling*; Springer: Cham, Switzerland, 2019; pp. 67–79.
26. Elfeki, M.; Borji, A. Video summarization via actionness ranking. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 754–763.
27. Satorras, V.G.; Rangapuram, S.S.; Januschowski, T. Multivariate time series forecasting with latent graph inference. *arXiv* **2022**, arXiv:2203.03423.
28. Mao, F.; Wu, X.; Xue, H.; Zhang, R. Hierarchical video frame sequence representation with deep convolutional graph network. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; pp. 262–270.

29. Li, P.; Tang, C.; Xu, X. Video summarization with a graph convolutional attention network. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 902–913. [CrossRef]
30. Ou, S.-H.; Lee, C.-H.; Somayazulu, V.-S.; Chen, Y.-K.; Chien, S.-Y. Low complexity on-line video summarization with Gaussian mixture model based clustering. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 1260–1264. [CrossRef]
31. Valdes, V.; Martinez, J.M. On-line video summarization based on signature-based junk and redundancy filtering. In Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, Klagenfurt, Austria, 7–9 May 2008; pp. 88–91.
32. Ma, M.; Mei, S.; Wan, S. Nonlinear Block Sparse Dictionary Selection for Video Summarization. *J. Xi'an Jiaotong Univ.* **2019**, *53*, 142–148. (In Chinese) [CrossRef]
33. Jadon, S.; Jasim, M. Unsupervised video summarization framework using keyframe extraction and video skimming. In Proceedings of the 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 30–31 October 2020; pp. 140–145.
34. Laganière, R.; Bacco, R.; Hocevar, A.; Lambert, P.; Ionescu, B.E. Video summarization from spatio-temporal features. In Proceedings of the 2nd ACM Workshop on Video Summarization, TVS 2008, Vancouver, BC, Canada, 31 October 2008.
35. Zhang, Y.; Wei, Z.; Zhao, Z.; Song, X.; Fu, L. A gaussian video summarization method using video frames similarity function. *ICIC Express Lett.* **2013**, *7*, 1997–2003.
36. Mahasseni, B.; Lam, M.; Todorovic, S. Unsupervised video summarization with adversarial lstm networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 202–211.
37. Jiang, J.; Zhang, X.-P. Gaussian mixture vector quantization-based video summarization using independent component analysis. In Proceedings of the 2010 IEEE International Workshop on Multimedia Signal Processing, Saint-Malo, France, 4–6 October 2010; pp. 443–448.
38. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. *arXiv* **2019**, arXiv:1904.09237.
39. Gygli, M.; Grabner, H.; Riemenschneider, H.; Gool, L.V. Creating summaries from user videos. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2014; pp. 505–520.
40. Song, Y.; Vallmitjana, J.; Stent, A.; Jaimes, A. Tvsum: Summarizing web videos using titles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 5179–5187.
41. Kaufman, D.; Levi, G.; Hassner, T.; Wolf, L. Temporal tessellation: A unified approach for video analysis. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 94–104.
42. Zhou, K.; Qiao, Y.; Xiang, T. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
43. Rochan, M.; Wang, Y. Video summarization by learning from unpaired data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7902–7911.
44. Yaliniz, G.; Ikizler-Cinbis, N. Unsupervised Video Summarization with Independently Recurrent Neural Networks. In Proceedings of the 2019 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey, 24–26 April 2019.
45. Apostolidis, E.; Adamantidou, E.; Metsai, A.I.; Mezaris, V.; Patras, I. AC-SUM-GAN: Connecting actor-critic and generative adversarial networks for unsupervised video summarization. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 3278–3292. [CrossRef]
46. Liang, G.; Lv, Y.; Li, S.; Zhang, S.; Zhang, Y. Unsupervised Video Summarization with a Convolutional Attentive Adversarial Network. *arXiv* **2021**, arXiv:2105.11131.
47. Jung, Y.; Cho, D.; Kim, D.; Woo, S.; Kweon, I.S. Discriminative feature learning for unsupervised video summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 8537–8544.
48. Li, P.; Ye, Q.; Zhang, L.; Yuan, L.; Xu, X.; Shao, L. Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recognit.* **2021**, *111*, 107677. [CrossRef]
49. Wei, H.; Ni, B.; Yan, Y.; Yu, H.; Yang, X.; Yao, C. Video summarization via semantic attended networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
50. Zhang, Y.; Kampffmeyer, M.; Zhao, X.; Tan, M. Dtr-gan: Dilated temporal relational adversarial network for video summarization. In Proceedings of the ACM Turing Celebration Conference-China, Chengdu, China, 17–19 May 2019; pp. 1–6.
51. Ji, Z.; Xiong, K.; Pang, Y.; Li, X. Video summarization with attention-based encoder–decoder networks. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1709–1717. [CrossRef]
52. Fu, T.-J.; Tai, S.-H.; Chen, H.-T. Attentive and adversarial learning for video summarization. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 7–11 January 2019; pp. 1579–1587.
53. Yuan, L.; Tay, F.E.; Li, P.; Zhou, L.; Feng, J. Cycle-SUM: Cycle-consistent adversarial LSTM networks for unsupervised video summarization. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9143–9150.