sensors

MDPI

Article

A Video Summarization Model Based on Deep Reinforcement Learning with Long-Term Dependency

Xu Wang, Yujie Li *, Haoyu Wang, Longzhao Huang and Shuxue Ding

School of Artificial Intelligence, Guilin University of Electronic Technology, Jinji Road, Guilin 541004, China
* Correspondence: yujieli@guet.edu.cn

Abstract: Deep summarization models have succeeded in the video summarization field based on the development of gated recursive unit (GRU) and long and short-term memory (LSTM) technology. However, for some long videos, GRU and LSTM cannot effectively capture long-term dependencies. This paper proposes a deep summarization network with auxiliary summarization losses to address this problem. We introduce an unsupervised auxiliary summarization loss module with LSTM and a swish activation function to capture the long-term dependencies for video summarization, which can be easily integrated with various networks. The proposed model is an unsupervised framework for deep reinforcement learning that does not depend on any labels or user interactions. Additionally, we implement a reward function ($R(S)$) that jointly considers the consistency, diversity, and representativeness of generated summaries. Furthermore, the proposed model is lightweight and can be successfully deployed on mobile devices and enhance the experience of mobile users and reduce pressure on server operations. We conducted experiments on two benchmark datasets and the results demonstrate that our proposed unsupervised approach can obtain better summaries than existing video summarization methods. Furthermore, the proposed algorithm can generate higher F scores with a nearly 6.3% increase on the SumMe dataset and a 2.2% increase on the TVSum dataset compared to the DR-DSN model.

Keywords: video summarization; reinforcement learning; unsupervised learning; long-term dependency; auxiliary summarization loss

Citation: Wang, X.; Li, Y.; Wang, H.; Huang, L.; Ding, S. A Video Summarization Model Based on Deep Reinforcement Learning with Long-Term Dependency. *Sensors* **2022**, *22*, 7689. https://doi.org/10.3390/s22197689

Academic Editors: Amin Ullah, Tanveer Hussain and Mohammad Farhad Bulbul

Received: 14 August 2022
Accepted: 3 October 2022
Published: 10 October 2022

1. Introduction

According to YouTube, there were approximately 5.5 billion daily video views in the first quarter of 2022. The massive amount of video information available causes people to spend significant time browsing and understanding redundant videos. Therefore, it is important to determine how to find relevant videos quickly among the endless video supply. Video retrieval techniques can help people find videos related to keywords, whereas video summarization techniques can extract representative information directly from a video. Video summarization techniques can generate concise summaries for videos that convey the important components of a complete video. Generating video summaries typically involves using a limited number of images or video clips to represent the main content of an original video sequence, which preserves the authenticity of the video information and saves a significant amount of space.

Research on video summarization techniques has been conducted for decades and good results have been achieved by traditional methods based on sparse modeling representative selection (SMRS) [1], DC programming [2], etc. With the development of deep learning, many researchers have used deep learning methods to extract features from videos and have achieved good results. M. Fei et al. [3] further introduced image entropy in deep learning to maintain the diversity of summaries. T. Hussain et al. [4] introduced aesthetic and entropic features to keep the summaries interesting and diverse. T. Hussain et al. [5] applied deep video summarization technology to real life. They proposed a

video summarization method using depth features of the lens segmentation method and applied it to resource-constrained devices. In addition, T. Hussain et al. [6] proposed a deep learning-based video summarization strategy for industrial surveillance scenes to achieve coarse and refined video data, which provides a great contribution to the application of video summarization technology. Additional layers are meaningful in these methods [4–6], but they will consume more computation time and increase the complexity of the system. Therefore, an efficient learning mechanism to train DL models remains a challenge. Furthermore, it is difficult to handle the long-term dependency relationship due to the long timeline when processing the long-term monitoring data in the existing methods.

However, deep-learning-based methods rely on labels, whereas reinforcement learning does not rely on labels and allows models to explore and select features in an unsupervised manner. Reinforcement learning updates model parameters using reward functions and gradient descent techniques. In recent years, various researchers have combined deep learning with reinforcement learning to apply deep reinforcement learning (DRL) methods to the task of video summarization. Zhou et al. [7] proposed a deep learning framework that combines a bidirectional long short-term memory (LSTM) network (BiLSTM) with DRL and called their framework the diversity-representativeness deep summarization network (DR-DSN). The deep summarization network (DSN) predicts a probability for each video frame that indicates the likelihood of selecting that video frame and then takes action to select frames based on the predicted probability distribution to perform video summarization. DSN achieves good results and is competitive in the field of unsupervised video summarization. In addition, most deep-learning-based methods generate video summaries based on supervised learning, where they learn the importance of frames by modeling the temporal dependency between frames or the spatiotemporal structure of the video. The cost of producing video summarization datasets with labels is very expensive, so we focus on an unsupervised video summarization model based on deep reinforcement learning.

However, the performance of a deep reinforcement learning model degrades as video length increases because of the long-term dependency problem. In video summarization tasks, connections between adjacent frames are often established using recursive neural networks, but for tens of thousands of frames of video information, it is difficult to retain information from distant frames, which may lead to the degradation of model performance. This problem is called the long-term dependency problem. This problem limits the performance of existing approaches that focus on semantic objects, actions, emotions, and diversity. Even BiLSTM-based methods are not immune to the problems caused by long-term dependency. Large kernel sizes and deep networks give convolutional neural networks (CNNs) the ability to alleviate the long-term dependency problem. However, RNNs have a completely different structure compared to CNNs and we cannot directly employ the CNN structure in an RNN. To solve this problem, Trinh et al. [8] introduced auxiliary loss into the main supervised loss function to reconstruct or predict sequence information, which allows RNNs to capture long-term dependencies.

In this paper, to address the long-term dependency problem in the video summarization task, we equate video summarization to a sequential decision process based on inspiration from Trinh et al. [8]. We propose a summarization selection network with unsupervised summarization loss based on an infinite norm. Similar to existing DSNs [7], this network also utilizes an encoder–decoder architecture, where the encoder is a CNN that extracts high-dimensional features from videos and the decoder is a one-way LSTM network utilizing unsupervised summary loss to capture the long-term dependencies between video frames. The value of video frames can be calculated from video frame features. The higher the value, the greater the probability of being selected as a key frame. We trained the proposed network using an end-to-end reinforcement learning framework to solve the video summarization problem by updating the model parameters using a reward function that does not rely on any labels or user interactions. Finally, we propose a discrete degree reward function $R(S)$ and compare it to other reward functions to demonstrate its

practical value. An overview of the proposed model is presented in Figure 1. Compared to a traditional RNN, the proposed network is much better at maintaining long-term memory.

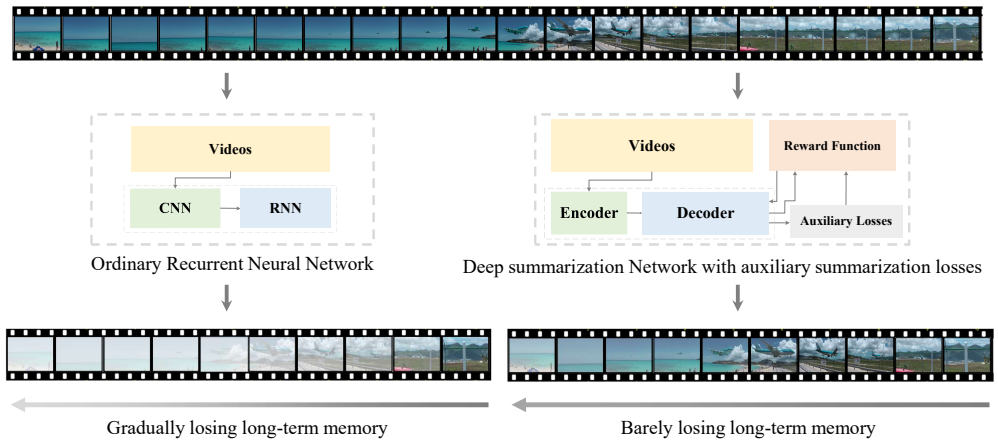


Figure 1. Overview of networks with long-term dependency capture capability (the proposed model) and recurrent neural networks.

An unsupervised auxiliary summarization loss module is proposed to reconstruct random segments of a video by connecting to the LSTM network in parallel at randomly determined anchor points during model training. This loss is used to adjust the parameters of the model by calculating the degree of video feature loss during training. Experimental results demonstrate that the proposed unsupervised auxiliary summary loss can accurately capture the long-term dependencies in videos, significantly improving the performance and generalization ability of the LSTM network.

Furthermore, we explored several reward functions in reinforcement learning for the video summary model. The diversity-representation reward was proposed by Zhou et al. [7]. The diversity reward evaluates the degree of diversity of generated summaries by computing the differences between selected frames. The representation reward is used to measure the degree to which generated summaries can represent an original video. Although this reward can identify diversity-representative video frames, the cluttered distribution of excessive non-key frames disrupts summarized videos. To address this issue, we propose dispersion rewards, which allow the proposed method to cluster key frames together and reduce the probability of selecting non-key frames. As a result, our model preserves segment integrity and generates high-quality summaries. Additionally, our proposed method is generic and can be easily integrated into existing unsupervised video summarization models.

In summary, the main contributions of this paper are threefold.

1. We propose a DRL framework with an unsupervised summary loss for video summarization called AuDSN, which solves the long-term dependency problem of reinforcement-learning-based key frame extraction algorithms. Compared to RNN/LSTM-based video summarization methods, our proposed video summarization framework can capture the long-term dependencies between video frames more effectively;
2. We employ an unsupervised auxiliary summarization loss for video summarization, which assists in tuning network parameters by calculating the percentage differences between original and selected features. Additionally, the unsupervised auxiliary summarization loss does not increase the parameters of the model and can be easily integrated into other video summarization models;
3. We propose a novel reward function called dispersion reward (R_{dis}) and employ it as a final reward function. Experimental results demonstrate the effectiveness of this reward function for video summarization.

The remainder of this paper is organized as follows. The related work is discussed in detail in Section 2. Section 3 describes the proposed pipeline-based AuDSN model. We present the experimental results in Section 4. Finally, Section 5 summarizes the key contributions of our work and concludes this paper.

2. Related Work

2.1. Video Summarization

Traditional video summarization methods [9–11] are shot- or segment-based methods, meaning an input video is divided into short shots or segments using various detection or segmentation algorithms. However, with the development of deep learning, the research on video summarization has made significant progress. Deep-learning-based video summarization utilizes neural networks to extract video information and select key segments and key frames in an end-to-end manner. Deep learning research on video summarization can be divided into three main categories: supervised video summarization [12–16], unsupervised video summarization [7,17–20], and weakly supervised video summarization. Our approach focuses on unsupervised video summarization.

Early supervised-learning-based approaches treated the selection of key frames as a prediction problem by modeling the temporal dependencies of each frame to estimate the importance of each frame and generate video summaries based on importance values. Zhang et al. [12] used LSTM to model the dependencies between video frames and derive representative and compact video summaries. One year later, Zhao et al. [13] proposed a hierarchical RNN based on LSTM that is more suitable for video summarization. Li et al. [15] proposed an efficient CNN based on global diverse attention by improving the self-attention mechanism of the transformer network, which adapts an attention mechanism from a global perspective to consider the pairwise temporal relationships between video frames.

In the context of unsupervised video summarization, because there is no universally accepted definition of video key frames, the goal of most existing unsupervised approaches is to maximize a viewer's ability to grasp the original content of a video based on selected representative key frames. Mahasseni et al. [21] first applied a generative adversarial network (GAN) to the task of video summarization and their key concept was to train a deep summarization network that uses an unsupervised approach to minimize the distance between an input video and summarization distribution.

He et al. [17] proposed a conditional GAN based on self-attentiveness. The generator generates weighted frame features and predicts the importance of each frame, while the discriminator is used to distinguish between weighted frame features and original frame features. Rochan et al. [18] proposed a video summarization model based on a GAN and fully convolutional sequence network to perform video summarization utilizing unpaired data. However, due to the instability of the training stage and the limitations of the evaluation metrics of GAN methods, the GAN-based methods do not achieve good results.

Zhou et al. [7] used a combination of reinforcement learning and reward functions for video summarization. Video summarization was formulated as a sequential decision process, and a diversity-representativeness reward was used to train the summarizer to produce diverse and representative video summaries. In this model, BiLSTM is used as a decoder to estimate the importance of frames, but the LSTM forgetting mechanism leads to the exponential decay of information, limiting the model's ability to capture long-term scale information. To address these issues, we introduce an unsupervised auxiliary summarization loss module to help train the LSTM network to capture the long-term dependencies between video frames. We also propose a novel reward with a dispersion reward. The dispersion reward function uses more intuitive criteria to help models improve performance.

2.2. DRL

It is common knowledge that deep learning has strong perceptual ability but lacks decision-making ability, whereas reinforcement learning has decision-making ability but lacks perceptual capabilities. Combining these two approaches can provide a solution to the perceptual decision problem in complex systems [22]. Therefore, DRL was devised to combine the perceptual ability of deep learning with the decision-making ability of reinforcement learning to achieve control directly based on input signals, which is an artificial intelligence method similar to the human thinking process [23]. In March of 2016, AlphaGo [24] defeated top professional player Lee Sedol with a score of 4:1 in a historic Go match after practicing and reinforcing itself over tens of thousands of games. This achievement was a testament to the potential of DRL development. Over the past few years, DRL has solved many real-world challenging decision problems with spectacular success. DRL has been successfully applied in the fields of computer vision, natural language processing, gaming, and robotics [25].

In the field of computer vision, DRL has been successfully applied to tasks such as landmark detection [26,27], object detection [28], object tracking [29], image registration [30], image segmentation [31], and video analysis [32]. Video summarization is a useful, but difficult task in video analysis that involves the prediction of objects or tasks in a video. The video summarization task can also be considered a decision-making task for each frame in a video. This task has been continuously improved since an unsupervised video summarization DRL method was first proposed by Zhou et al. [7]. Next, Zhou et al. [33] implemented a summarization network using deep Q-learning (DQSN) and used a trained classification network to provide rewards for training DQSN. This method is a weakly supervised approach based on reinforcement learning that utilizes easily accessible video-level category labels and encourages summaries to contain category-relevant information and maintain category identifiability. Liu et al. [34] used a 3D spatiotemporal U-net to encode the spatiotemporal information of input videos efficiently, allowing an RL again to learn from spatiotemporal information and predict the acceptance or rejection of video frames in a video summary. Unlike supervised learning methods, DRL learns information from the environment and does not require a large number of labeled data, which enables many machine learning applications for which no large labeled training data are available.

2.3. Long-Term Dependencies

In the field of artificial intelligence, many important applications require learning long-term dependencies between sequences [35]. It is challenging to provide neural networks with the ability to model the long-term dependencies in sequence data. In a video analysis, a 2-hour movie typically contains approximately 170,000 images. If we use deep learning to understand the content of a movie, long-term dependencies for 170,000 images must be constructed to capture the features of the movie. Typically, this is achieved using gradient descent and back-propagation through time (BPTT) in recursive networks. However, learning long-term dependencies using gradient descent is difficult because the gradients computed by BPTT tend to disappear or explode during training (Hochreiter et al. [36]). The solution to this problem is to mitigate vanishing gradients by allowing information to be stored in memory (LSTM [37]) or by using a short-circuiting mechanism in a residual structure (ResNet [38]). Additionally, for BPTT to work properly, intermediate hidden states in a sequence must be stored. However, the memory requirement is proportional to the sequence length, making it difficult to apply BPTT to long video sequences.

Vorontsov et al. [39] investigated the effects of the widening of the spectral boundary on convergence and performance by controlling the orthogonality constraint and regularization of the power matrix to solve the gradient disappearance and explosion problem associated with BPTT during the training process. This approach can capture long-term dependencies to some extent and improve the convergence speed of gradient descent. Trinh et al. [8] proposed an auxiliary loss for capturing long-term dependencies, which can help RNNs reconstruct the previous event in a sequence or predict the next event

in a sequence. Based on the development of transformers, Dai et al. [40] introduced the concept of recursion in a transformer by reusing historical representations to propose the transformer-xl model. This model does not compute hidden states for each new segment but reuses hidden states obtained from previous segments. Reused hidden states serve as memory for the current segment and circular connections are established between segments. However, the transformer models have a large number of parameters and consume a large amount of computational power for training. In contrast to these existing approaches, our proposed video summarization method reconstructs the current video information between segments by incorporating unsupervised auxiliary summarization loss functions, which helps the sequence model capture long-term dependencies. It is noteworthy that the unsupervised auxiliary summarization loss function does not require the use of any labeling information and the auxiliary module only works in the training model and does not increase the overall parameters and complexity of the model.

3. Methodology

This paper proposes a deep summarization network with auxiliary summarization losses to improve the performance of video summarization. Specifically, the auxiliary summarization losses are used as auxiliary training in the proposed AuDSN model. A brief overview of the AuDSN model is detailed as follows.

We consider video summarization as the task of making decisions for each frame in a video. We implement DRL models to predict the value of the information contained in each frame and determine the probability of each frame being a key frame based on these values. We then select the final key frames based on a probability distribution. Inspired by Zhou et al. [7], we developed an end-to-end reinforcement-learning-based summarization network. The pipeline of this network is presented in Figure 2. We train AuDSN using reinforcement learning. AuDSN receives observations (video frames) at each time step and performs various actions on the observations. Additionally, AuDSN generates auxiliary summary loss values. The environment receives the actions and auxiliary summary loss values and generates a reward and observation for the next time step. The data $(O_i, A_i, O_{i\_}, R_i)$ generated during this process are recorded in a sample database and when the sample data are sufficient, loss values are calculated and the model parameters are updated.

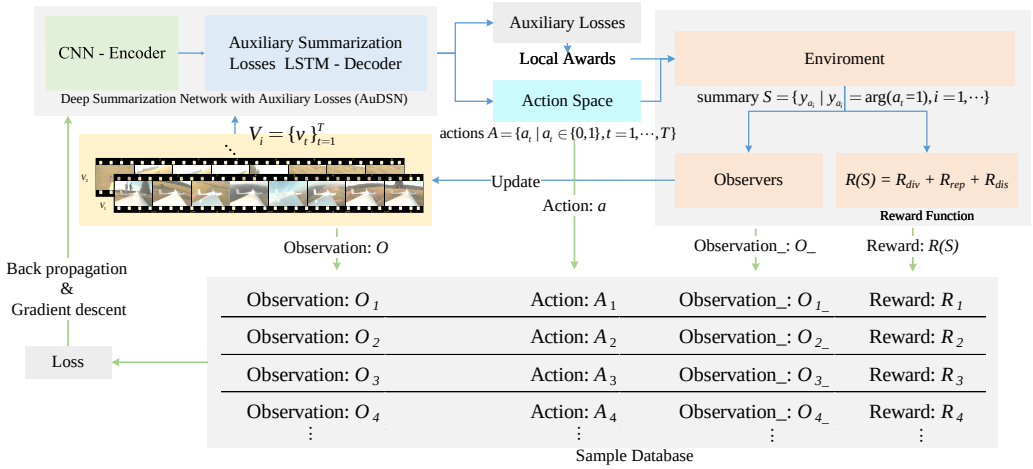


Figure 2. Pipeline of the proposed AuDSN. The blue arrow represents the process of generating a summary and the green arrow represents the process of model training.

The proposed AuDSN is an encoder–decoder architecture, where the encoder uses a CNN (typically GoogLeNet [41]) to extract visual features $\{x_t\}_{t=1}^{T}$ from an input video frame sequence of length t $\{v_t\}_{t=1}^{T}$. The decoder uses an LSTM network that can capture temporal features. It randomly inserts anchor points in the LSTM network and introduces unsupervised auxiliary summarization loss. A fully connected layer serves as the final layer

of the network. The LSTM network takes the high-dimensional features $\{x_t\}_{t=1}^{T}$ generated by the encoder as inputs and generates the corresponding hidden states $\{h_t\}_{t=1}^{T}$, some of which will be passed to the unsupervised auxiliary summarization loss function to help the summary network reconstruct historical information as a method of establishing long-term dependencies. The hidden state $\{h_t\}_{t=1}^{T}$ is loaded with past information. The value of each frame is obtained through a fully connected layer using the swish activation function [42] and the fully connected layer using the sigmoid activation function. The value of each frame is used to determine the action a as follows:

$$\text{value}_{\,t} = R(w_r \times \sigma(w_\sigma \times h_t + b_\sigma) + b_r), \tag{1}$$

$$p_t = \frac{\text{value}_{\,t}}{\sum_{t=1}^{T} \text{value}_{\,t}}, \tag{2}$$

$$a_t = \text{Bernoulli}\,(p_t), \tag{3}$$

where $R(\cdot)$ is the rectified linear unit (ReLU) activation function, σ is the sigmoid activation function, w_r, w_x are the weights of the fully connected layer, and b_r, b_x are the biases of the fully connected layer. p_t denotes the probability distributions over all possible action sequences. a_t are binary samples drawn from the Bernoulli distribution indicating whether the $i-th$ frame is selected. The Bernoulli distribution is parameterized by p_t. We obtain a video summary composed of the selected key frames $S = \{\cdots y_{a_t} \cdots\}$, where y_{a_t} is the selected key frame with $a_t = 1$. We only update the decoder during the training process.

3.1. Auxiliary Summarization Loss

To enhance the ability of LSTM to capture long-term dependencies, we propose a form of unsupervised auxiliary summarization loss that applies to video summarization. This loss not only enhances the memory capabilities of LSTM but can also be easily migrated to other models. It is experimentally demonstrated that this auxiliary summarization loss can help the summarization network establish longer-lasting dependencies and improve the performance of video summarization. During the training progress, we randomly sample multiple anchor locations in the LSTM and insert an unsupervised auxiliary summarization loss function at each location.

An unsupervised auxiliary summarization loss function first reconstructs memories from the past by sampling the subsequence after its anchor point, copying the hidden state of a subsequence of length n after its anchor point, and inserting the first input of this subsequence into the decoder network. The rest of the subsequence features are reconstructed using the decoder network, as shown in Figure 3. By using this training method, anchor points can be used as temporary memory points for the decoder network to store memories in a sequence. When we insert enough anchor points, the decoder network remembers the entire sequence and reconstructs memories. Therefore, when we reach the end of the sequence, the decoder network remembers a sufficient number of sequences and can generate a high-quality video summary.

The introduction of unsupervised auxiliary summarization loss in the decoder network results in the need for some additional hyperparameters, namely the sampling interval $\mathcal{T}$ and subsequence length $\mathcal{N}$. The sampling interval is defined as the samples per unit length that are extracted from the decoder network and composed into subsequences. The subsequence length represents the number of original features contained in each subsequence. We define the auxiliary summarization loss as follows:

$$L_{\text{auxiliary}} = \frac{\sum_{i=1}^{\mathcal{T}} L_i}{\sum_{i=1}^{\mathcal{T}} l_i}, \tag{4}$$

where $\mathcal{T}$ denotes the sampling interval, L_i denotes the loss evaluated on the i-th sampling segment, and the overall auxiliary summarization loss is calculated by summing all subse-

quence auxiliary summarization losses for a segment. We define the subsequence auxiliary summarization loss as follows:

$$L_i = \frac{1}{\mathcal{N} \times l} \sum_{j=1}^{l} \sum_{i=1}^{\mathcal{N}} |Y - C|, \tag{5}$$

where $\mathcal{N}$ denotes the subsequence length, l denotes the feature dimension (based on the CNN encoder, the feature dimension we obtain is typically 1024), Y denotes the original features of the subsequence, and C denotes the key summary features of the subsequence after decoder selection. The auxiliary summarization loss of each subsequence is obtained by computing the infinite norm error between the vectorized Y and C features. This auxiliary summarization loss represents the percentage of the variance between the original features Y and key summary features C of the subsequence selected by the decoder.

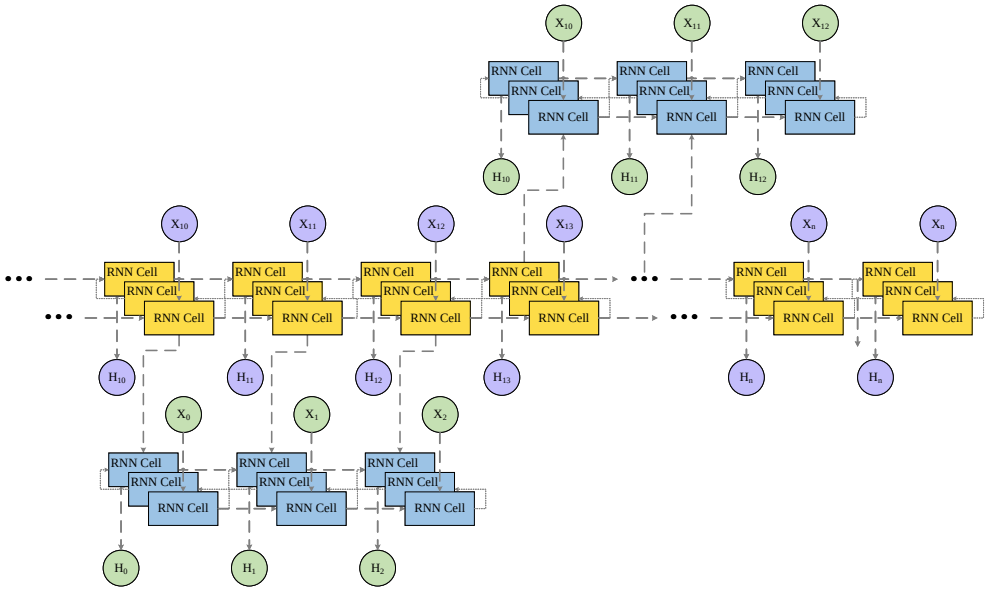


Figure 3. The decoder module with unsupervised auxiliary summarization loss. Yellow RNN cells represent decoder networks, blue RNN cells represent auxiliary loss modules. X represents the input features. H represents the decision of RNN cells based on input.

Tuning hyperparameters is a very expensive process, so we set the lengths of all selected subsequences $\mathcal{N}$ to be equal. Because the addition of unsupervised auxiliary summarization loss does not change the structure of the decoder module, and high-quality embeddings of input sequences can be learned based on the unsupervised auxiliary summarization loss, the weights of the LSTM network can be fine-tuned using the back-propagation step only.

3.2. Reward Function

In reinforcement learning, the goal of an agent is represented as a special signal called a reward, which is transmitted to the agent through the environment. At each moment, the reward is a single scalar value. Informally, the goal of an agent is to maximize the total reward it receives. This means that what needs to be maximized is not the current reward, but the long-term cumulative reward.

An agent always learns how to maximize its gain and the amount of gain is determined by the reward function. If we want an agent to perform a particular task, then we must provide rewards in such a manner that the agent maximizes its gain while completing the task. Therefore, it is critical that we design the reward function in a way that truly achieves our goal. In video summarization, the key frames are selected from the video, and the selected frames should be diverse, representative, and uniform. Gygli et al. [10]

mentioned that there is no standard answer for a correct summary, but we must guarantee the generation of a high-quality summary. Figure 4 presents the results of the label visualization of three sample videos (Cooking, Fire Domino, and Jumps) from the SumMe dataset. Each video was labeled by 15 to 18 people and each line in the video visualization represents one person's selection result. As shown in Figure 4, we should select key frames from labels such as those in Figure 4a,b for video summarization. In our proposed model, we demonstrate that combining dispersion rewards with two rewards for diversity and representativeness can produce enhanced summaries.

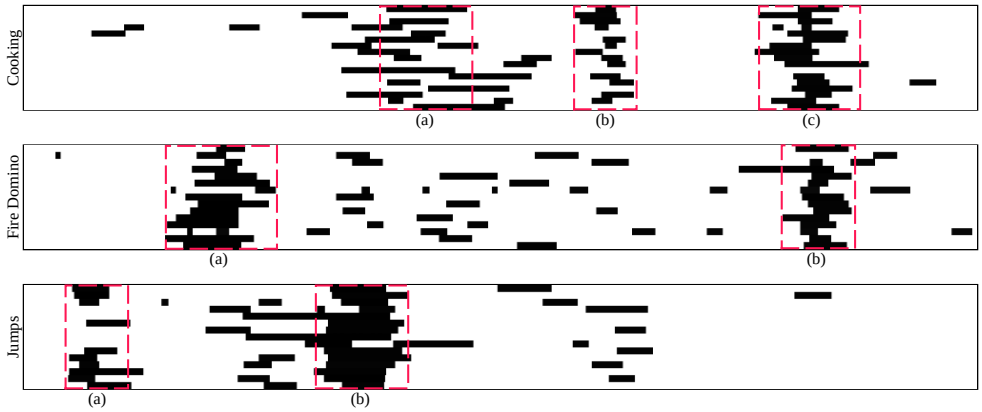


Figure 4. Visual results of human-selected labels for three example videos (cooking, fire domino, and jumps) from SumMe dataset. (**a**–**c**) are example labels which are selected by most people.

3.2.1. Diversity Reward

The diversity reward function calculates the degree of diversity of a video summary by evaluating the dissimilarity between selected key frames. Specifically, it calculates the variability between selected video frames in the computational feature space to evaluate the diversity of a video summary quantitatively. The diversity reward function is defined as follows:

$$R_{div} = \frac{1}{|y||y-1|} \sum_{t \in y} \sum_{\substack{t' \in y \\ t \neq t'}} d(x_t, x_{t'}), \tag{6}$$

where $y = \{y_{a_t} | a_t = 1, t = 1 \cdots T\}$ is the set of selected video frames, x_t is the feature sequence corresponding to the video frames and $d(\cdot, \cdot)$ represents the dissimilarity function to calculate the dissimilarity between two frames, which is defined as follows:

$$d(x_t, x_{t'}) = 1 - \frac{x_t^T x_{t'}}{\|x_t\|_2 \|x_{t'}\|_2}. \tag{7}$$

Therefore, the reward obtained by the agent is higher when the selected frames are more diverse (i.e., larger differences between selected video frames are favored). To ignore the similarity between two temporally distant frames, we set $d(x_t, x_t) = 1$ if $|t - t'| > \lambda$, where λ controls the degree of temporal distance consideration [7].

3.2.2. Representativeness Reward

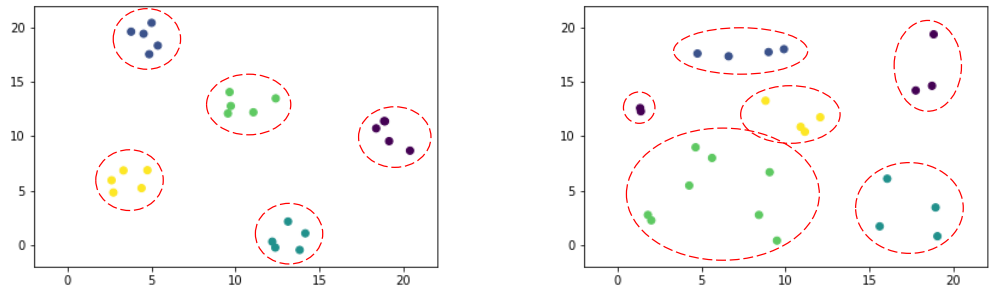
The representativeness reward is used to measure how well selected key frames represent an original video [7]. Evaluating the representativeness of a video summary can be formulated as the k-medoids problem [43], where the representativeness reward is defined as follows:

$$R_{rep} = \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \min_{t' \in y} \|x_t - x_{t'}\|_2\right), \tag{8}$$

where x_t is the feature of the $t-th$ frame and y is the set of selected frames, as described above.

3.2.3. Dispersion Reward

When using representativeness rewards, the degree of representativeness is defined as a k-medoids problem. To ensure that the selected video frames are uniform, we propose a dispersion reward function to complement the representativeness reward function to achieve better selection results. The dispersion reward function uses more intuitive criteria. As shown in Figure 5, the distribution of (a), which has a higher dispersion reward, exhibits significantly better clustering than that of (b), which has a lower dispersion reward. We want clustering results to have small intra-class distances and large inter-class distances (i.e., classification results should have high discrimination). To this end, a criteria function reflecting both intra- and inter-class distances can be constructed.



(a) $R_{dis} = 228$ (b) $R_{dis} = 75$

Figure 5. The dispersion reward results for two different clustering cases; (**a**) 20 artificially generated points; (**b**) 20 randomly generated points in a certain range. Different colors of points represent different classes.

The intra-class departure matrix is generated by calculating the distance from each sample point to the cluster center. The intra-class distance criteria function, which produces an intra-class departure matrix S_W, is defined as follows:

$$S_W^j = \frac{1}{n_j} \sum_{i=1}^{n_j^s} \left(\bar{x}_i^{(j)} - \bar{m}_j \right) \left(\bar{x}_i^{(j)} - \bar{m}_j \right)^T, j = 1, 2 \ldots c, \tag{9}$$

where $\bar{m}$ is the sample mean vector of class w, sn denotes the number of samples in class w, and c denotes the number of categories classified. $\bar{x}_i^{(j)}$ denotes the i^{th} data sample in the j^{th} class.

The inter-class distance criterion function, which produces an inter-class deviation matrix s_b, is defined as follows:

$$S_B = \sum_{j=1}^{c} \frac{n_j^s}{N^S} \left(\bar{m}_j - \bar{m} \right) \left(\bar{m}_j - \bar{m} \right)^T, \tag{10}$$

where c denotes the number of classes to be classified, m is the mean vector of all samples to be classified, n^s denotes the number of samples in class w, N^S denotes the total number of samples, and m_j denotes the class center of each class.

The dispersion reward based on the intra-class distance criteria function and inter-class distance criteria function is defined as follows:

$$R_{dis} = \mathrm{Tr} \left[S_w^{-1} S_b \right], \tag{11}$$

where $\mathrm{Tr}[\cdots]$ is the matrix trace.

R_{div}, R_{rep}, and R_{dis} complement each other and work jointly to guide the learning of AuDSN. The determination of optimal weights for the different reward types is discussed later in this paper. The basic reward function is expressed as follows:

$$R = R_{div} + R_{rep} + R_{dis}. \tag{12}$$

3.3. Deep Summarization Network with Auxiliary–Summarization Loss-Based Video Summarization

We train the summary network using a DRL framework. The goal of the proposed summarization method [7] is to learn a policy function π_θ with parameters θ by maximizing the expected rewards

$$J(\theta) = \mathbb{E}_{p_\theta(a_{1:T})}[R], \tag{13}$$

where p_θ is computed using Equation (2) and R is computed using Equation (12). $\mathbb{E}$ represents the expected value for calculating the reward. π_θ is defined by our AuDSN.

The agent selects key frames based on the policy function π_θ. This selection is defined as an action a_t. The agent combines the current observation state o_t and feature space of the selected video frames to predict a new observation state $o_{i\_}$. The next action a_{t+1} is determined based on the n-th reward R_n. The derivative of $J(\theta)$ is computed using the REINFORCE algorithm proposed by [44]. Only the encoder of the network is updated during training.

To reduce the computational load, the final gradient is calculated as the average gradient after executing N iterations on the same video. A constant baseline b is subtracted from the n^{th} reward R_n to avoid the type of high variance that leads to divergence. This baseline b is calculated based on the moving average of the rewards from the previous time steps. The gradient formula is defined as follows:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} (R_n - b) \nabla_\theta \log \pi_\theta(a_t \mid h_t). \tag{14}$$

Additionally, we introduce a regularization term into the probability distributions to constrain the percentage of frames selected for the summary and add the L_2 regularization term to the weight parameters θ to avoid overfitting [7]. The parameters of the optimal policy function are optimized using the stochastic gradient algorithm by combining all conditions. We use Adam [45] as an optimization algorithm.

4. Experiments

4.1. Datasets

We primarily used the SumMe [10] and TVSum [46] datasets to evaluate our approach. SumMe consists of 25 videos ranging from 1 to 6 min in length with a variety of video content covering holidays, events, and sports captured from first- and third-person perspectives. Each video was annotated by 15 to 18 humans and the summary lengths range from 5 to 15% of the initial video durations. TVSum consists of 50 videos ranging from 1 to 11 min in length and contains video contents from 10 categories of the TRECVid Med dataset [47]. The TVSum videos were annotated by 20 users in the form of shot- and frame-level importance scores (from 1 to 5).

4.2. Evaluation Settings and Metrics

For fair comparison to other methods, we used the common F-score measure to assess the similarity between the selected key frames and ground-truth labels [10]. The F score is defined as follows:

$$F_i = \frac{1}{N_i} \sum_{j=1}^{N_i} 2 \frac{P_{i,j} R_{i,j}}{P_{i,j} + R_{i,j}}, \tag{15}$$

where N_i is the number of available user-generated summaries for the i-th test video $P_{i,j}$ and $R_{i,j}$ is the accuracy and recall ratio for the j-th user summary. These values are calculated on a per-frame basis. Accuracy refers to the degree of agreement between measured values and the corresponding "true" values. The recall ratio represents the total number of positive queries among the query sample, which can also be interpreted as the number of correct predictions among the true positive samples.

For a test video, we used a trained AuDSN to predict the value of each frame as importance scores. We computed shot-level scores by averaging frame-level scores within the same shot. For video shots, the kernel temporal segmentation (KTS) [48] method split the video into a set of non-intersecting temporal segments. The algorithm input the matrix of frame-to-frame similarities and output a set of optimal "change points" that correspond to the boundaries of temporal segments. A summary was generated by selecting the video shots with the highest total scores, which were calculated from the frame-level scores of the shots. The maximized total score problem is essentially the 0/1 knapsack problem, for which a near-optimal solution can be obtained using dynamic programming [7].

4.3. Implementation Details

To compare AuDSN to existing methods, we used a GoogleNet [41] model pre-trained on ImageNet to extract features and a 1024-dimensional feature vector from the penultimate layer of GoogleNet (pool 5) as the features for each video frame. We used the standard five-fold cross-validation in the training stage, meaning 80% of the videos were used for training and 20% were used for testing. We utilized the auxiliary summarization loss in our decoder module. The auxiliary summarization loss and reward mechanism were used to tune the parameters of the model and the KTS implementation [48] was used for temporal segmentation. During our experiments, we tested different activation functions to compare their effects on the model and kept adjusting the hyperparameters to achieve optimal performance.

4.4. Quantitative Results

In Table 1, we present the definitions of various models with different active functions and rewards. AuDSN corresponds to a deep summary network using unsupervised auxiliary summarization loss with a reward function using a diversity reward (R_{div}) and representativeness reward (R_{rep}). In practice, the swish activation function outperforms the ReLU activation function on some CNN-based tasks, such as image classification and machine translation. Therefore, we adopted swish as an activation function in AuDSN-S. AuDSN-D corresponds to deep summary networks using a diversity reward (R_{div}), representativeness reward (R_{rep}), and dispersion reward (R_{dis}). AuDSN-SD corresponds to a combination of swish as an activation function and all three reward functions.

Table 1. Definitions of different models.

Model Name	Activation Function	Reward
AuDSN	ReLU	$R_{div} + R_{rep}$
AuDSN-S	swish	$R_{div} + R_{rep}$
AuDSN-D	ReLU	$R_{div} + R_{rep} + R_{dis}$
AuDSN-SD	swish	$R_{div} + R_{rep} + R_{dis}$

Table 2 reports the performances of different models on the SumMe and TVSum datasets. One can see that AuDSN-SD performs significantly better than AuDSN-S and AuDSN-D on both datasets, indicating that using swish as an activation function and using $R_{div} + R_{rep} + R_{dis}$ summation as a reward function can better teach the summary network to generate high-quality video summaries. The performance of AuDSN-S indicates that using swish as an activation function improves performance by 0.4% (46.8% − 46.4%) and 0.5% (59.3% − 58.8%) on the two datasets, respectively, demonstrating that the swish activation function can provide the neural network layers with better performance for video

summarization. The performance of AuDSN-D indicates that using R_dis in combination with R_{div} and R_{rep} can improve performance on the SumMe and TVSum datasets by 0.2% (46.6% − 46.4%) and 0.7% (59.5% − 58.8%), respectively. These results demonstrate that adding the dispersion reward R_{dis} to the reward function can improve model performance. The proposed method performs slightly better on the TVSum dataset than on the SumMe dataset. This may indicate that the dispersion reward function has a greater impact on longer videos.

Table 2. Performance (F-scores (%)) of different variants of the proposed method.

Method	SumMe	TVSum
AuDSN	46.4	58.8
AuDSN-S	46.8	59.3
AuDSN-D	46.6	59.5
AuDSN-SD	**47.7**	**59.8**

In Table 3, we compare the proposed method to existing GRU- and LSTM-based methods using DRL. DR-DSN [7] is an unsupervised deep summarization method that uses BiLSTM as the decoder in its summarization network. DR-DSN$_{\text{SUP}}$ [7] is a supervised version of DR-DSN. DR1-DSN [49] uses Chebyshev distance instead of Euclidean distance in its diversity reward. DR2-DSN [49] is a version of DR1-DSN that uses a double hidden layer in its GRU. The results demonstrate that our proposed methods obtain higher scores than the previous methods on both datasets. These results indicate that our proposed unsupervised auxiliary summarization loss has stronger long-term dependency-capturing ability. The proposed AuDSN-SD yields higher F scores with a nearly 4.3% (47.7% − 43.4%) increase on the SumMe dataset and 1.3% (59.8% − 58.5%) increase on the TVSum dataset compared to the state-of-the-art DR2-DSN model [49]. Compared to DR-DSN [7], the proposed AuDSN-SD improves performance on the SumMe and TVSum datasets by 6.3% (47.7% − 41.4%) and 2.2% (59.8% − 57.6%), respectively. Additionally, we can see that our proposed method also outperforms the supervised model DR-DSN$_{sup}$ (5.6% (47.7% − 42.1%) improvement on SumMe and 1.7% (59.8% − 58.1%) improvement on TVSum).

Table 3. Performance (F-scores (%)) of the proposed methods compared to other DRL-based methods.

Method	Network	SumMe	TVSum
DR-DSN [7]	LSTM	41.4	57.6
DR-DSN$_{sup}$ [7]	LSTM	42.1	58.1
DR1-DSN [49]	GRU	42.9	58.3
DR2-DSN [49]	GRU	43.4	58.5
AuDSN [OURS]	LSTM	46.4	58.8
AuDSN-S [OURS]	LSTM	46.8	59.3
AuDSN-D [OURS]	LSTM	46.6	59.5
AuDSN-SD [OURS]	LSTM	**47.7**	**59.8**

Table 4 reports the F-score results of the proposed method and state-of-the-art methods on both datasets. In this table, the performance of both supervised and unsupervised methods is reported. We can see that our proposed framework outperforms most of the other methods. The use of auxiliary summarization loss instead of bidirectional LSTM to capture long-term dependencies [7] yields a significant improvement in performance on both datasets. The results show that our proposed method can achieve competitive results compared to the state-of-the-art methods. It is worth noting that our proposed method is based on unsupervised learning, and the structure is simpler and lighter.

It is noteworthy that most of the previous classical baseline models are tens or even hundreds of megabytes in size. These models require a large number of parameters, large memory, high computing power, and a long time for training. The size of our proposed

model is only 2.7 MB, which is much smaller than most previous models. In Table 5, we list the model sizes of some open-source models. The addition of auxiliary summarization loss allows our model to capture the dependencies between video frames accurately without using a complex network structure. Furthermore, the addition of auxiliary summarization loss does not increase the size of our model because auxiliary summarization loss is only applied in the training stage and automatically discarded at the end of training.

Table 4. Performance (F-scores (%)) of the proposed method compared to existing methods.

Method	SumMe		TVSum	
	F1	Rank	F1	Rank
Random summary [50]	40.2	14	54.4	14
Online Motion-AE [51]	37.7	15	51.5	15
DR-DSN [7]	41.4	13	57.6	10
CSNet [52]	**51.3**	1	58.8	4
Cycle-SUM [53]	41.9	12	57.6	10
DR1-DSN [49]	42.9	9	58.3	8
DR2-DSN [49]	43.4	8	58.5	5
UnpairedVSN [18]	47.5	3	55.6	13
EDSN [54]	42.6	11	57.3	12
PCDL [55]	42.7	10	58.4	7
ACGAN [17]	46.0	5	58.5	5
DHAVS$_{sup}$ [56]	45.6	6	60.8	2
3DST-Unet$_{sup}$ [34]	47.4	4	58.3	8
M-AVS$_{sup}$ [57]	44.4	7	**61.0**	1
AuDSN-SD [OURS]	47.7	2	59.8	3

An extremely small number of parameters allows AuDSN to save significant training time and computational cost, which is important given the exponential growth in the number of online videos in recent years. AuDSN can significantly reduce the computational pressure on video websites so that they can handle larger batches of videos simultaneously. It is worth noting that our lightweight model also makes it possible to deploy video summarization on edge devices and mobile devices. When AuDSN is successfully deployed on mobile devices, it can improve the experience of mobile users and reduce the pressure on servers. Deploying AuDSN to edge devices can enrich the information processing capabilities of such devices and may give rise to novel application scenarios.

Table 5. Sizes of the parameters generated by different methods.

Method	Model Size (MB)
DR-DSN [7]	10.0
CSNet [52]	28.0
UnpairedVSN [18]	95.4
AuDSN [OURS]	**2.7**

4.5. Comparison of the Augmented (A) and Transfer (T) Settings

Table 6 reports the F-score results of the proposed method in the canonical (C) augmented (A), and transfer (T) settings. C setting means that the training and testing sets are from the same dataset. A setting means that OVP, YouTube, TVSum, and 80% of SumMe are used as the training sets and the remaining 20% of SumMe is used as the testing set. T setting means that OVP, YouTube, and TVSum are used as the training sets and SumMe is used as the testing set. Experimental results show that using the A setting can improve the performance of our proposed model by 0.2%.

Table 6. F-score (%) of the LSTM-based approaches on SumMe and TVSum in the canonical (C), augmented (A), and transfer (T) settings.

Method	SumMe			TVSum		
	C	A	T	C	A	T
SUM-GAN$_{rep}$ [21]	38.5	42.5	-	51.9	59.3	-
SUM-GAN$_{dpp}$ [21]	39.1	43.4	-	51.7	59.5	-
DR-DSN [7]	41.4	42.8	42.4	57.6	58.4	57.8
CSNet [52]	**51.3**	**52.1**	45.1	58.8	59	**59.2**
Cycle-SUM [53]	41.9	-	-	57.6	-	-
AuDSN-SD [OURS]	47.7	47.9	**45.1**	**59.8**	**59.7**	57.8

4.6. Qualitative Comparisons

In this section, we present qualitative comparison results for an example video called "playing ball" from the SumMe dataset. The frame labels of the video are presented in Figure 6. Figure 7 presents the qualitative results of our proposed models for the video. The gray bars in Figure 7 represent the results of user selection, the height of the bars represents the number of people who selected each frame as a key frame, and the colored bars represent the frame segments selected by the proposed methods. Some selected frames are presented at the bottom of each bar.

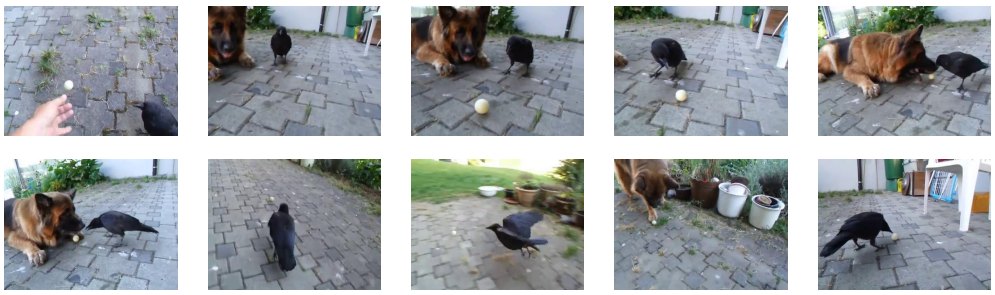
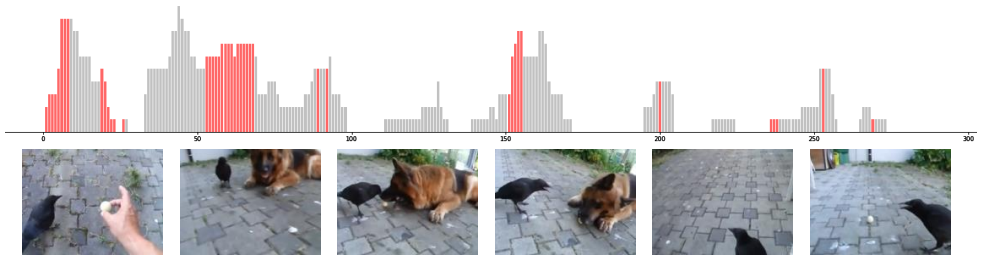
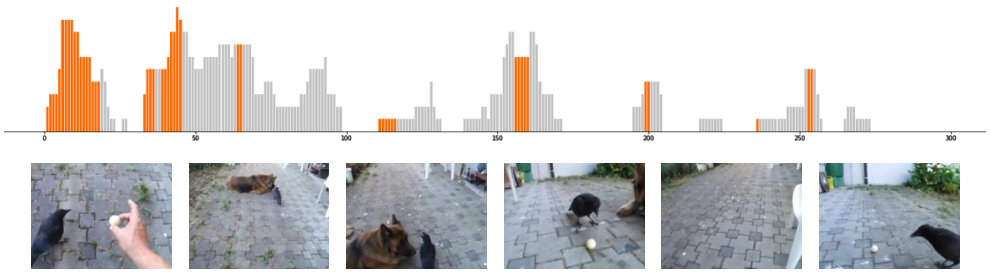


Figure 6. Examples of frame labels from the "playing ball" video.

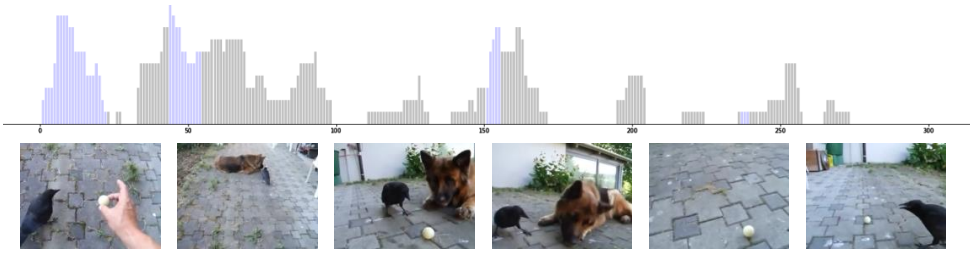


(a) A generated sumary by AuDSN

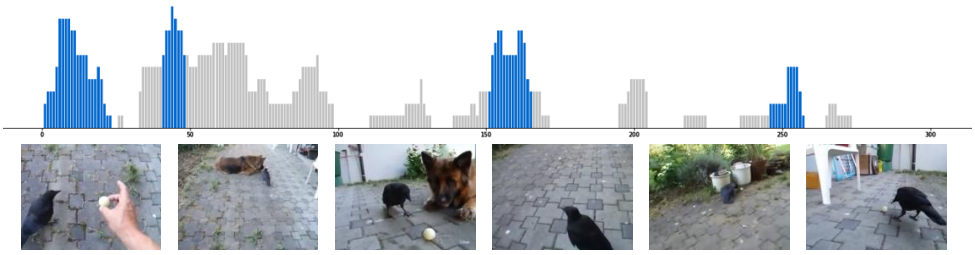


(b) A generated sumary by AuDSN-S

Figure 7. *Cont.*



(c) A generated sumary by AuDSN-D



(d) A generated sumary by AuDSN-SD

Figure 7. Video summaries generated by all variants of the proposed method.

In general, our proposed models produce high-quality video summaries. The basic AuDSN model typically selects the high-quality parts of the summaries. AuDSN-SD using swish as an activation function with the combination of $R_{div} + R_{rep} + R_{dis}$ as reward functions can select frames with even higher values. Furthermore, our proposed AuDSN-SD does not produce jumps between frames, so the selected video summary is more coherent, making it easier for viewers to understand the video content without making them feel uncomfortable based on larges jumps between frames.

4.7. Ablation Experiments on the Effects of Hyperparameters

Table 7 presents the performance of AuDSN-SD when using different values for the hyperparameter $\mathcal{T}$. The optimal hyperparameters were determined experimentally. One can see that the performance is optimal for the SumMe dataset when the sampling interval is 40 and for the TVSum dataset when the sampling interval is 30. The greater the video length, the better long-term dependencies can be captured by appropriately reducing the sampling interval.

Table 7. Performance with different sampling intervals $\mathcal{T}$ on two datasets.

$\mathcal{T}$	10	20	30	40	50
SumMe	46.7	46.9	47.1	47.7	46.8
TVSum	59.3	59.3	59.8	59.3	58.7

Figure 8 presents the performance of AuDSN-SD when using different values for the hyperparameter $\mathcal{N}$. We considered 10 to 90% of the video length as the subsequence length $\mathcal{N}$ to determine the optimal $\mathcal{N}$. On the TVSum dataset, the model achieves the optimal performance when the subsequence length is 70% of the video length. On the SumMe dataset, the model achieves the optimal performance when the subsequence length is 50% of the video length. Model performance decreases when the subsequence length continues to increase. Therefore, as the video length increases, the selected subsequence length should also be increased appropriately to ensure optimal model performance.

4.8. Effects of different CNN Encoders

Table 8 presents the performance of AuDSN-SD using different encoders. We have attempted the most advanced feature extractor proposed in recent years to compare with

our CNN encoder. From Table 6, we can see that using GoogleNet as the CNN encoder achieves the best performance. Furthermore, MobileNet, which can obtain lightweight features, is also a competitive encoder. We can give priority to using MobileNet as the CNN encoder of the model in the case of insufficient computational power.

4.9. Effects of Reward Weights

As shown in Table 3, the performance of AuDSN is improved by 0.2% (46.6% − 46.4%) and 0.7% (59.5% − 58.8%) when using discrete rewards on the SumMe and TVSum datasets, respectively. The main contribution of the discrete reward is improved video summary quality. The discrete reward attempts to group interrelated segments together. Therefore, the model with the discrete reward preserves the story line of the video by eliminating redundant jumps between adjacent clips. The comparison in Figure 7 reveals that the model with the dispersion reward outputs more continuous clips and fewer single video frames.

To allow our model to achieve better results, we evaluated the proposed AuDSN model by assigning different weights to the three reward functions. For the TVSum and SumMe datasets, we present several results based on different weights in Table 9 and visualize these results in Figure 9. One can see that representativeness rewards have a greater impact on the overall reward function than diversity rewards. In Figure 9, we present the results of the optimal R_{div} = 0.2, R_{rep} = 0.2, and R_{rep} = 0.6 weights for the SumMe dataset, and R_{div} = 0.4, R_{rep} = 0.2, and R_{rep} = 0.6 weights for the TVSum dataset. The three weights should satisfy: $R_{div} + R_{rep} + R_{dis} = 1$. Therefore, the edges of the figure are triangular.

In Table 10, one can see that the best results are obtained when the dispersion reward weight is 0.2 for the TVSum Dataset. For long videos, the representativeness and diversity rewards play a greater role in the selection of summary segments/frames. The dispersion reward plays an auxiliary role and its purpose is to make video summaries more uniform and continuous while avoiding jumps between video frames and maintaining the continuity and integrity of the story line of a summary. The dispersion reward function does not improve model performance significantly, but it has a noticeable impact on the quality of video summaries.

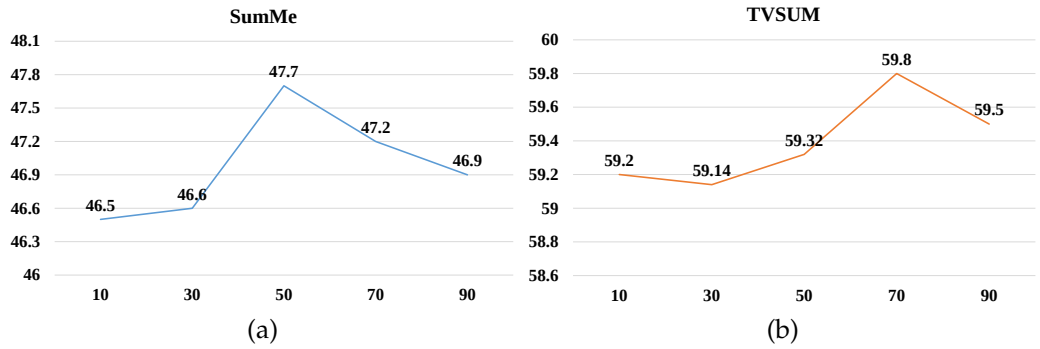


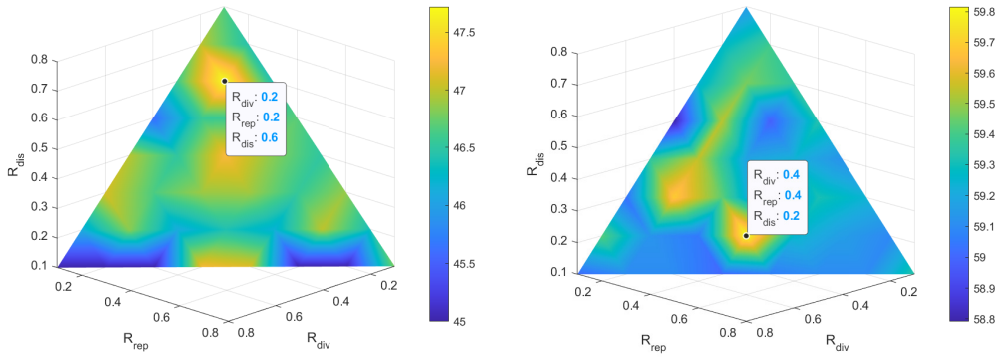
Figure 8. Performance with different subsequence lengths $\mathcal{N}$ on two datasets: (**a**) SumMe dataset and (**b**) TVSum dataset. The x-axis represents the percentage of video length and the y-axis represents the F score.

Table 8. Performances with different CNN encoders on two datasets.

	GoogleNet [41]	EfficientNet [58]	MobileNet [59]	ResNet [38]	VGG [60]	VIT [61]
SumMe	47.7	46.8	47.3	47.2	46.9	46.8
TVSum	59.8	59.3	59.5	59.4	59.4	59.2

Table 9. Effects of different reward weights on SumMe and TVSum performance (F-scores).

R_{div}	R_{rep}	R_{dis}	SumMe	TVSum
0.1	0.1	0.8	46.7	59.1
0.1	0.2	0.7	46.4	59.4
0.1	0.3	0.6	46.4	59.4
0.1	0.4	0.5	46.7	59.3
0.1	0.5	0.4	46.5	59.1
0.1	0.6	0.3	46.9	59.2
0.1	0.7	0.2	46.4	59.1
0.1	0.8	0.1	46.8	59.3
0.2	0.1	0.7	47.0	59.3
0.2	0.2	0.6	**47.7**	59.5
0.2	0.3	0.5	46.6	59.0
0.2	0.4	0.4	46.8	59.2
0.2	0.5	0.3	46.5	59.3
0.2	0.6	0.2	47.0	59.2
0.3	0.1	0.6	46.6	59.2
0.3	0.2	0.5	46.6	59.5
0.3	0.3	0.4	47.2	59.2
0.3	0.4	0.3	46.9	59.0
0.3	0.5	0.2	46.2	59.3
0.4	0.1	0.5	45.6	58.8
0.4	0.2	0.4	46.5	59.6
0.4	0.3	0.3	46.9	59.2
0.4	0.4	0.2	46.4	**59.8**
0.4	0.5	0.1	47.1	59.1
0.5	0.1	0.4	47.0	59.4
0.5	0.2	0.3	46.6	59.2
0.5	0.3	0.2	46.1	59.0
0.5	0.4	0.1	47.2	59.0
0.6	0.1	0.3	47.1	59.0
0.6	0.2	0.2	46.8	59.1
0.7	0.1	0.2	46.1	59.4
0.8	0.1	0.1	46.5	59.0



(a) (b)

Figure 9. Effects of different reward weights on SumMe and TVSum performances (F score): (**a**) SumMe dataset and (**b**) TVSum dataset. The three axes correspond to the weights of the three reward functions and the depth of the colors represents the F scores.

Table 10. Weights that yield the best performances for the reward functions on both datasets.

DataSet	R_{div}	R_{rep}	R_{dis}	F1
SumMe	0.20	0.20	0.60	47.7
TVSum	0.40	0.40	0.20	59.8

5. Conclusions

In this paper, we proposed AuDSN, which is a deep reinforcement network model with unsupervised auxiliary summarization loss. We introduced unsupervised auxiliary summarization loss in the decoder and explored a novel reward function with a dispersion reward. Experimental results on two datasets (SumMe and TVSum) demonstrated that introducing unsupervised auxiliary summarization loss can improve the long-term dependency-capturing ability for a deep summarization network. Additionally, the swish activation function and dispersion reward function can help a deep summarization network construct more coherent, diverse, and representative video summaries. Furthermore, AuDSN is a very lightweight model with a size of only 2.7 MB, presenting the opportunity of deploying it on low-computing-power edge devices. In future work, we will also try to incorporate multi-information, such as sound features, into the model and explore multi-information versions of the deep summarization network framework. Furthermore, we will try to incorporate the proposed video summarization approach into modern media tools to make practical use of the proposed summarization algorithms.

Author Contributions: Conceptualization, X.W. and Y.L.; Formal analysis, X.W., Y.L. and S.D.; Methodology, X.W.; Software, X.W. and H.W.; Validation, Y.L. and S.D.; Visualization, X.W.; Writing – original draft, X.W., H.W. and L.H.; Writing – review and editing, Y.L. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This workwas supported in part by the National Natural Science Foundation of China (61903090), Guangxi Natural Science Foundation (2022GXNSFBA035644), and the Guangxi Science and Technology Major Project (AA22068057).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Elhamifar, E.; Sapiro, G.; Vidal, R. See all by looking at a few: Sparse modeling for finding representative objects. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1600–1607.
2. Tan, B.; Li, Y.; Ding, S.; Paik, I.; Kanemura, A. DC programming for solving a sparse modeling problem of video key frame extraction. *Digit. Signal Process.* **2018**, *83*, 214–222. [CrossRef]
3. Fei, M.; Jiang, W.; Mao, W. Memorable and rich video summarization. *J. Vis. Commun. Image Represent.* **2017**, *42*, 207–217. [CrossRef]
4. Muhammad, K.; Hussain, T.; Tanveer, M.; Sannino, G.; de Albuquerque, V.H.C. Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. *IEEE Internet Things J.* **2019**, *7*, 4455–4463. [CrossRef]
5. Muhammad, K.; Hussain, T.; Baik, S.W. Efficient CNN based summarization of surveillance videos for resource-constrained devices. *Pattern Recognit. Lett.* **2020**, *130*, 370–375. [CrossRef]
6. Muhammad, K.; Hussain, T.; Del Ser, J.; Palade, V.; De Albuquerque, V.H.C. DeepReS: A deep learning-based video summarization strategy for resource-constrained industrial surveillance scenarios. *IEEE Trans. Ind. Inform.* **2019**, *16*, 5938–5947. [CrossRef]
7. 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; Volume 32.
8. Trinh, T.; Dai, A.; Luong, T.; Le, Q. Learning longer-term dependencies in rnns with auxiliary losses. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4965–4974.
9. Ejaz, N.; Mehmood, I.; Baik, S.W. Efficient visual attention based framework for extracting key frames from videos. *Signal Process. Image Commun.* **2013**, *28*, 34–44. [CrossRef]
10. Gygli, M.; Grabner, H.; Riemenschneider, H.; Gool, L.V. Creating summaries from user videos. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 505–520.
11. Otani, M.; Nakashima, Y.; Rahtu, E.; Heikkilä, J.; Yokoya, N. Video summarization using deep semantic features. In *Asian Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 361–377.
12. Zhang, K.; Chao, W.L.; Sha, F.; Grauman, K. Video summarization with long short-term memory. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 766–782.
13. Zhao, B.; Li, X.; Lu, X. Hierarchical recurrent neural network for video summarization. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 863–871.

14. Yan, X.; Gilani, S.Z.; Feng, M.; Zhang, L.; Qin, H.; Mian, A. Self-supervised learning to detect key frames in videos. *Sensors* **2020**, *20*, 6941. [CrossRef] [PubMed]
15. Li, P.; Ye, Q.; Zhang, L.; Yuan, L.; Xu, X.; Shao, L. Exploring global diverse attention via pairwise temporal relation for video summarization. *Pattern Recognit.* **2021**, *111*, 107677. [CrossRef]
16. Rafiq, M.; Rafiq, G.; Agyeman, R.; Choi, G.S.; Jin, S.I. Scene classification for sports video summarization using transfer learning. *Sensors* **2020**, *20*, 1702. [CrossRef]
17. He, X.; Hua, Y.; Song, T.; Zhang, Z.; Xue, Z.; Ma, R.; Robertson, N.; Guan, H. Unsupervised video summarization with attentive conditional generative adversarial networks. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 2296–2304.
18. 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.
19. Yoon, U.N.; Hong, M.D.; Jo, G.S. Interp-SUM: Unsupervised Video Summarization with Piecewise Linear Interpolation. *Sensors* **2021**, *21*, 4562. [CrossRef]
20. Yaliniz, G.; Ikizler-Cinbis, N. Using independently recurrent networks for reinforcement learning based unsupervised video summarization. *Multimed. Tools Appl.* **2021**, *80*, 17827–17847. [CrossRef]
21. 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.
22. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
23. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
24. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
25. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep reinforcement learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
26. Ghesu, F.C.; Georgescu, B.; Zheng, Y.; Grbic, S.; Maier, A.; Hornegger, J.; Comaniciu, D. Multi-scale deep reinforcement learning for real-time 3D-landmark detection in CT scans. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *41*, 176–189. [CrossRef]
27. Alansary, A.; Oktay, O.; Li, Y.; Le Folgoc, L.; Hou, B.; Vaillant, G.; Kamnitsas, K.; Vlontzos, A.; Glocker, B.; Kainz, B.; et al. Evaluating reinforcement learning agents for anatomical landmark detection. *Med Image Anal.* **2019**, *53*, 156–164. [CrossRef]
28. Wang, Y.; Zhang, L.; Wang, L.; Wang, Z. Multitask learning for object localization with deep reinforcement learning. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *11*, 573–580. [CrossRef]
29. Dunnhofer, M.; Martinel, N.; Luca Foresti, G.; Micheloni, C. Visual tracking by means of deep reinforcement learning and an expert demonstrator. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
30. Sun, S.; Hu, J.; Yao, M.; Hu, J.; Yang, X.; Song, Q.; Wu, X. Robust multimodal image registration using deep recurrent reinforcement learning. In *Asian Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 511–526.
31. Tian, Z.; Si, X.; Zheng, Y.; Chen, Z.; Li, X. Multi-step medical image segmentation based on reinforcement learning. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 1–12. [CrossRef]
32. Le, N.; Rathour, V.S.; Yamazaki, K.; Luu, K.; Savvides, M. Deep reinforcement learning in computer vision: A comprehensive survey. *Artif. Intell. Rev.* **2021**, *55*, 2733–2819. [CrossRef]
33. Zhou, K.; Xiang, T.; Cavallaro, A. Video summarisation by classification with deep reinforcement learning. *arXiv* **2018**, arXiv:1807.03089.
34. Liu, T.; Meng, Q.; Huang, J.J.; Vlontzos, A.; Rueckert, D.; Kainz, B. Video summarization through reinforcement learning with a 3D spatio-temporal u-net. *IEEE Trans. Image Process.* **2022**, *31*, 1573–1586. [CrossRef] [PubMed]
35. Chandar, S.; Sankar, C.; Vorontsov, E.; Kahou, S.E.; Bengio, Y. Towards non-saturating recurrent units for modelling long-term dependencies. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27–28 January 2019; Volume 33, pp. 3280–3287.
36. Hochreiter, S.; Bengio, Y.; Frasconi, P.; Schmidhuber, J. Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks,* Kremer; C, S., Kolen; F, J., Eds.; Wiley-IEEE Press: Hoboken, NJ, USA, 2001.
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 July 2016; pp. 770–778.
39. Vorontsov, E.; Trabelsi, C.; Kadoury, S.; Pal, C. On orthogonality and learning recurrent networks with long term dependencies. In Proceedings of the International Conference on Machine Learning. PMLR, Sydney, Australia, 6–11 August 2017; pp. 3570–3578.
40. Dai, Z.; Yang, Z.; Yang, Y.; Cohen, W.W.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. *Transformer-xl: Language Modeling with Longer-Term Dependency*. 2018. Available online: https://openreview.net/forum?id=HJePno0cYm (accessed on 13 August 2022).

41. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 2–12 June 2015; pp. 1–9.
42. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
43. Gygli, M.; Grabner, H.; Van Gool, L. Video summarization by learning submodular mixtures of objectives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3090–3098.
44. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
45. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
46. 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.
47. Smeaton, A.F.; Over, P.; Kraaij, W. Evaluation campaigns and TRECVid. In Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, Santa Barbara, CA, USA, 26–27 October 2006; pp. 321–330.
48. Potapov, D.; Douze, M.; Harchaoui, Z.; Schmid, C. Category-specific video summarization. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 540–555.
49. Wang, L.; Zhu, Y.; Pan, H. Unsupervised reinforcement learning for video summarization reward function. In Proceedings of the 2019 International Conference on Image, Video and Signal Processing, Wuhan, China, 29–31 October 2019; pp. 40–44.
50. 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]
51. Zhang, Y.; Liang, X.; Zhang, D.; Tan, M.; Xing, E.P. Unsupervised object-level video summarization with online motion auto-encoder. *Pattern Recognit. Lett.* **2020**, *130*, 376–385. [CrossRef]
52. 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, 27–28 January 2019; Volume 33, pp. 8537–8544.
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, 27–28 January 2019; Volume 33, pp. 9143–9150.
54. Gonuguntla, N.; Mandal, B.; Puhan, N. Enhanced deep video summarization network. In Proceedings of the BMVC, Cardiff, UK, 9–12 September 2019.
55. Zhao, B.; Li, X.; Lu, X. Property-constrained dual learning for video summarization. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3989–4000. [CrossRef]
56. Lin, J.; Zhong, S.h.; Fares, A. Deep hierarchical LSTM networks with attention for video summarization. *Comput. Electr. Eng.* **2022**, *97*, 107618. [CrossRef]
57. Li, P.; Tang, C.; Xu, X. Video summarization with a graph convolutional attention network. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 902–913. [CrossRef]
58. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
59. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
60. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
61. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.