

# Discriminative Feature Learning for Unsupervised Video Summarization

Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, In So Kweon

Korea Advanced Institute of Science and Technology, Korea

{yun9298a, cdh12242}@gmail.com, {mcahny, shwoo93, iskweon77}@kaist.ac.kr

## Abstract

In this paper, we address the problem of unsupervised video summarization that automatically extracts key-shots from an input video. Specifically, we tackle two critical issues based on our empirical observations: (i) Ineffective feature learning due to flat distributions of output importance scores for each frame, and (ii) training difficulty when dealing with long-length video inputs. To alleviate the first problem, we propose a simple yet effective regularization loss term called variance loss. The proposed variance loss allows a network to predict output scores for each frame with high discrepancy which enables effective feature learning and significantly improves model performance. For the second problem, we design a novel two-stream network named Chunk and Stride Network (CSNet) that utilizes local (chunk) and global (stride) temporal view on the video features. Our CSNet gives better summarization results for long-length videos compared to the existing methods. In addition, we introduce an attention mechanism to handle the dynamic information in videos. We demonstrate the effectiveness of the proposed methods by conducting extensive ablation studies and show that our final model achieves new state-of-the-art results on two benchmark datasets.

## Introduction

Video has become a highly significant form of visual data, and the amount of video content uploaded to various on-line platforms has increased dramatically in recent years. In this regard, efficient ways of handling video have become increasingly important. One popular solution is to summarize videos into shorter ones without missing semantically important frames. Over the past few decades, many studies (Song et al. 2015; Ngo, Ma, and Zhang 2003; Lu and Grauman 2013; Kim and Xing 2014; Khosla et al. 2013) have attempted to solve this problem. Recently, Zhang *et al.* showed promising results using deep neural networks, and a lot of follow-up work has been conducted in areas of supervised (Zhang et al. 2016a; 2016b; Zhao, Li, and Lu 2017; 2018; Wei et al. 2018) and unsupervised learning (Mahasseni, Lam, and Todorovic 2017; Zhou and Qiao 2018).

Supervised learning methods (Zhang et al. 2016a; 2016b; Zhao, Li, and Lu 2017; 2018; Wei et al. 2018) utilize ground

truth labels that represent importance scores of each frame to train deep neural networks. Since human-annotated data is used, semantic features are faithfully learned. However, labeling for many video frames is expensive, and overfitting problems frequently occur when there is insufficient label data. These limitations can be mitigated by using the unsupervised learning method as in (Mahasseni, Lam, and Todorovic 2017; Zhou and Qiao 2018). However, since there is no human labeling in this method, a method for supervising the network needs to be appropriately designed.

Our baseline method (Mahasseni, Lam, and Todorovic 2017) uses a variational autoencoder (VAE) (Kingma and Welling 2013) and generative adversarial networks (GANs) (Goodfellow et al. 2014) to learn video summarization without human labels. The key idea is that a good summary should reconstruct original video seamlessly. Features of each input frame obtained by convolutional neural network (CNN) are multiplied with predicted importance scores. Then, these features are passed to a generator to restore the original features. The discriminator is trained to distinguish between the generated (restored) features and the original ones.

Although it is fair to say that a good summary can represent and restore original video well, original features can also be restored well with uniformly distributed frame level importance scores. This trivial solution leads to difficulties in learning discriminative features to find key-shots. Our approach works to overcome this problem. When output scores become more flattened, the variance of the scores tremendously decreases. From this mathematically obvious fact, we propose a simple yet powerful way to increase the variance of the scores. Variance loss is simply defined as a reciprocal of variance of the predicted scores.

In addition, to learn more discriminative features, we propose Chunk and Stride Network (CSNet) that simultaneously utilizes local (chunk) and global (stride) temporal views on the video. CSNet splits input features of a video into two streams (chunk and stride), then passes both split features to bidirectional long short-term memory (LSTM) and merges them back to estimate the final scores. Using chunk and stride, the difficulty of feature learning for long-length videos is overcome.

Finally, we develop an attention mechanism to capture dynamic scene transitions, which are highly related to key-

shots. In order to implement this module, we use temporal difference between frame-level CNN features. If a scene changes only slightly, the CNN features of the adjacent frames will have similar values. In contrast, at scene transitions in videos, CNN features in the adjacent frames will differ a lot. The attention module is used in conjunction with CSNet as shown in Fig. 1, and helps to learn discriminative features by considering information about dynamic scene transitions.

We evaluate our network by conducting extensive experiments on SumMe (Gygli et al. 2014) and TVSum (Song et al. 2015) datasets. YouTube and OVP (De Avila et al. 2011) datasets are used for the training process in augmented and transfer settings. We also conducted an ablation study to analyze the contribution of each component of our design. Quantitative results show the selected key-shots and demonstrate the validity of difference attention. Similar to previous methods, we randomly split the test set and the train set five times. To make the comparison fair, we exclude duplicated or skipped videos in the test set.

Our overall contributions are as follows. (i) We propose variance loss, which effectively solves the flat output problem experienced by some of the previous methods. This approach significantly improves performance, especially in unsupervised learning. (ii) We construct CSNet architecture to detect highlights in local (chunk) and global (stride) temporal view on the video. We also impose a difference attention approach to capture dynamic scene transitions which are highly related to key-shots. (iii) We analyze our methods with ablation studies and achieve the state-of-the-art performances on SumMe and TVSum datasets.

## Related Work

Given an input video, video summarization aims to produce a shortened version that highlights the representative video frames. Various prior work has proposed solutions to this problem, including video time-lapse (Joshi et al. 2015; Kopf, Cohen, and Szeliski 2014; Poleg et al. 2015), synopsis (Pritch, Rav-Acha, and Peleg 2008), montage (Kang et al. 2006; Sun et al. 2014) and storyboards (Gong et al. 2014; Gygli et al. 2014; Gygli, Grabner, and Van Gool 2015; Lee, Ghosh, and Grauman 2012; Liu, Hua, and Chen 2010; Yang et al. 2015; Gong et al. 2014). Our work is most closely related to storyboards, selecting some important pieces of information to summarize key events present in the entire video.

Early work on video summarization problems heavily relied on hand-crafted features and unsupervised learning. Such work defined various heuristics to represent the importance of the frames (Song et al. 2015; Ngo, Ma, and Zhang 2003; Lu and Grauman 2013; Kim and Xing 2014; Khosla et al. 2013) and to use the scores to select representative frames to build the summary video. Recent work has explored supervised learning approach for this problem, using training data consisting of videos and their ground-truth summaries generated by humans. These supervised learning methods outperform early work on unsupervised approach, since they can better learn the high-level semantic knowledge that is used by humans to generate summaries.

Recently, deep learning based methods (Zhang et al. 2016b; Mahasseni, Lam, and Todorovic 2017; Sharghi, Laurel, and Gong 2017) have gained attention for video summarization tasks. The most recent studies adopt recurrent models such as LSTMs, based on the intuition that using LSTM enables the capture of long-range temporal dependencies among video frames which are critical for effective summary generation.

Zhang *et al.* (Zhang et al. 2016b) introduced two LSTMs to model the variable range dependency in video summarization. One LSTM was used for video frame sequences in the forward direction, while the other LSTM was used for the backward direction. In addition, a determinantal point process model (Gong et al. 2014; Zhang et al. 2016a) was adopted for further improvement of diversity in the subset selection. Mahasseni *et al.* (Mahasseni, Lam, and Todorovic 2017) proposed an unsupervised method that was based on a generative adversarial framework. The model consists of the summarizer and discriminator. The summarizer was a variational autoencoder LSTM, which first summarized video and then reconstructed the output. The discriminator was another LSTM that learned to distinguish between its reconstruction and the input video.

In this work, we focus on unsupervised video summarization, and adopt LSTM following previous work. However, we empirically worked out that these LSTM-based models have inherent limitations for unsupervised video summarization. In particular, two main issues exists: First, there is ineffective feature learning due to flat distribution of output importance scores and second, there is the training difficulty with long-length video inputs. To address these problems, we propose a simple yet effective regularization loss term called Variance Loss, and design a novel two-stream network named the Chunk and Stride Network. We experimentally verify that our final model considerably outperforms state-of-the-art unsupervised video summarization. The following section gives a detailed description of our method.

## Proposed Approach

In this section, we introduce methods for unsupervised video summarization. Our methods are based on a variational autoencoder (VAE) and generative adversarial networks (GAN) as (Mahasseni, Lam, and Todorovic 2017). We firstly deal with discriminative feature learning under a VAE-GAN framework by using variance loss. Then, a chunk and stride network (CSNet) is proposed to overcome the limitation of most of the existing methods, which is the difficulty of learning for long-length videos. CSNet resolves this problem by taking a local (chunk) and a global (stride) view of input features. Finally, to consider which part of the video is important, we use the difference in CNN features between adjacent or wider spaced video frames as attention, assuming that dynamic plays a large role in selecting key-shots. Fig. 1 shows the overall structure of our proposed approach.

## Baseline Architecture

We adopt (Mahasseni, Lam, and Todorovic 2017) as our baseline, using a variational autoencoder (VAE) and generative adversarial networks (GANs) to perform unsupervised

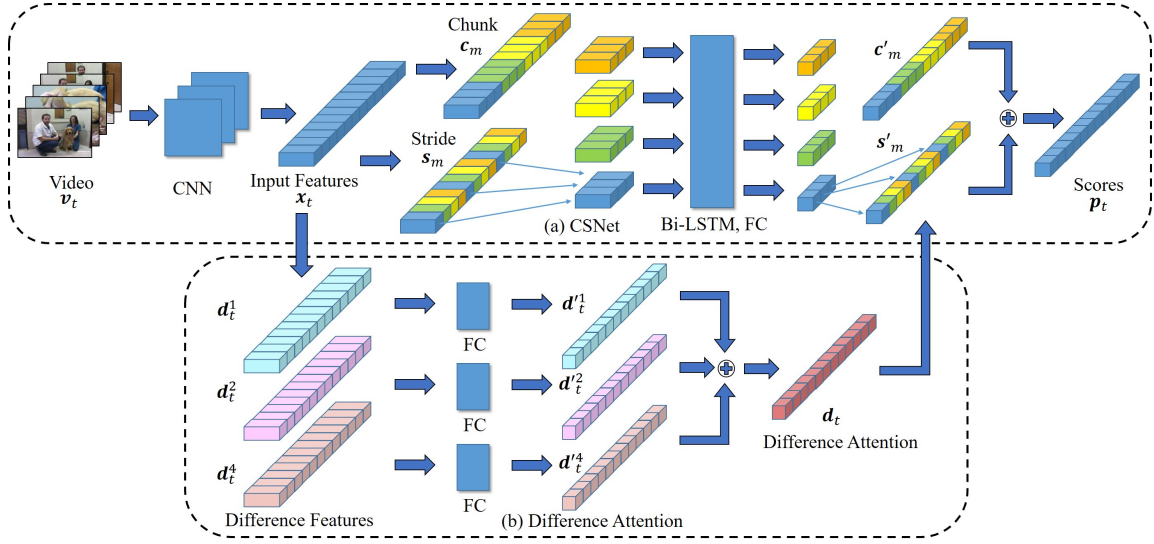


Figure 1: The overall architecture of our network. (a) chunk and stride network (CSNet) splits input features  $x_t$  into  $c_t$  and  $s_t$  by chunk and stride methods. Each orange, yellow, green, and blue color represents how the chunk and stride divide the input features  $x_t$ . Divided features are combined in the original order after going through LSTM and FC separately. (b) Difference attention is a approach for designing dynamic scene transitions at different temporal strides.  $d_t^1$ ,  $d_t^2$ ,  $d_t^4$  are difference of input features  $x_t$  with 1, 2, 4 temporal strides. Each difference features are summed after FC, which is denoted as difference attention  $d_t$ , and summed again with  $c'_t$  and  $s'_t$ , respectively.

video summarization. The key idea is that a good summary should reconstruct original video seamlessly and adopt a GAN framework to reconstruct the original video from summarized key-shots.

In the model, an input video is firstly forwarded through the backbone CNN (i.e., GoogleNet), Bi-LSTM, and FC layers (encoder LSTM) to output the importance scores of each frame. The scores are multiplied with input features to select key-frames. Original features are then reconstructed from those frames using the decoder LSTM. Finally, a discriminator distinguishes whether it is from an original input video or from reconstructed ones. By following Mahasseni *et al.*'s overall concept of VAE-GAN, we inherit the advantages, while developing our own ideas, significantly overcoming the existing limitations.

## Variance Loss

The main assumption of our baseline (Mahasseni, Lam, and Todorovic 2017) is “well-picked key-shots can reconstruct the original image well”. However, for reconstructing the original image, it is better to keep all frames instead of selecting only a few key-shots. In other words, mode collapse occurs when the encoder LSTM attempts to keep all frames, which is a trivial solution. This results in flat importance output scores for each frame, which is undesirable. To prevent the output scores from being a flat distribution, we propose a variance loss as follows:

$$\mathcal{L}_V(p) = \frac{1}{\hat{V}(p) + \epsilon}, \quad (1)$$

where  $p = \{p_t : t = 1, \dots, T\}$ ,  $\epsilon$  is epsilon, and  $\hat{V}(\cdot)$  is the variance operator.  $p_t$  is an output importance score at time  $t$ , and  $T$  is the number of frames. By enforcing Eq. (1), the network makes the difference in output scores per frames larger, then avoids a trivial solution (flat distribution).

In addition, in order to deal with outliers, we extend variance loss in Eq. (1) by utilizing the median value of scores. The variance is computed as follows:

$$\hat{V}_{median}((p)) = \frac{\sum_{t=1}^T |p_t - med(p)|^2}{T}, \quad (2)$$

where  $med(\cdot)$  is the median operator. As has been reported for many years (Pratt 1975; Huang, Yang, and Tang 1979; Zhang, Xu, and Jia 2014), the median value is usually more robust to outliers than the mean value. We call this modified function variance loss for the rest of the paper, and use it for all experiments.

## Chunk and Stride Network

To handle long-length videos, which are difficult for LSTM-based methods, our approach suggests a chunk and stride network (CSNet) as a way of jointly considering a local and a global view of input features. For each frame of the input video  $v = \{v_t : t = 1, \dots, T\}$ , we obtain the deep features  $x = \{x_t : t = 1, \dots, T\}$  of the CNN which is GoogLeNet pool-5 layer.

As shown in Fig. 1 (a), CSNet takes a long video feature  $x$  as an input, and divides it into smaller sequences in two ways. The first way involves dividing  $x$  into successive frames, and the other way involves dividing it at a uniform interval. The streams are denoted as  $c_m$ , and  $s_m$ , where

$\{m = 1, \dots, M\}$  and  $M$  is the number of divisions. Specifically,  $c_m$  and  $s_m$  can be explained as follows:

$$c_m = \left\{ x_i : i = (m-1) \cdot \left(\frac{T}{M}\right) + 1, \dots, m \cdot \left(\frac{T}{M}\right) \right\}, \quad (3)$$

$$s_m = \{x_i : i = m, m+k, m+2k, \dots, m+T-M\}, \quad (4)$$

where  $k$  is the interval such that  $k = M$ . Two different sequences,  $c_m$  and  $s_m$ , pass through the chunk and stride stream separately. Each stream consists of bidirectional LSTM (Bi-LSTM) and a fully connected (FC) layer, which predicts importance scores at the end. Then, each of the outputs are reshaped into  $c'_m$  and  $s'_m$ , enforcing the maintenance of the original frame order. Then,  $c'_m$  and  $s'_m$  are added with difference attention  $d_t$ . Details of the attentioning process are described in the next section. The combined features are then passed through sigmoid function to predict the final scores  $p_t$  as follows:

$$p_t^1 = \text{sigmoid}\left(c'_t + d_t\right), \quad (5)$$

$$p_t^2 = \text{sigmoid}\left(s'_t + d_t\right), \quad (6)$$

$$p_t = W[p_t^1 + p_t^2]. \quad (7)$$

where  $W$  is learnable parameters for weighted sum of  $p_t^1$  and  $p_t^2$ , which allows for flexible fusion of local (chunk) and global (stride) view of input features.

## Difference Attention

In this section, we introduce the attention module, exploiting dynamic information as guidance for the video summarization. In practice, we use the differences in CNN features of adjacent frames. The feature difference softly encodes temporally different dynamic information which can be used as a signal for deciding whether a certain frame is relatively meaningful or not.

As shown in Fig. 1 (b), the differences  $d_t^1, d_t^2, d_t^4$  between  $x_{t+k}$ , and  $x_t$  pass through the FC layer ( $d_t^1, d_t^2, d_t^4$ ) and are merged to become  $d_t$ , then added to both  $c_m$  and  $s_m$ . The proposed attention modules are represented as follows:

$$d_{1t} = |x_{t+1} - x_t|, \quad (8)$$

$$d_{2t} = |x_{t+2} - x_t|, \quad (9)$$

$$d_{4t} = |x_{t+4} - x_t|, \quad (10)$$

$$d_t = d'_{1t} + d'_{2t} + d'_{4t}. \quad (11)$$

While the difference between the features of adjacent frames can model the simplest dynamic, the wider temporal stride can include a relatively global dynamic between the scenes.

## Experiments

### Datasets

We evaluate our approach on two benchmark datasets, SumMe (Gygli et al. 2014) and TVSum (Song et al. 2015). SumMe contains 25 user videos with various events. The videos include both cases where the scene changes quickly or slowly. The length of the videos range from 1 minute to

Setting	Training set	Test set
Canonical	80% SumMe	20% SumMe
Augmented	OVP + YouTube + TVSum + 80% SumMe	20% SumMe
Transfer	OVP + YouTube + TVSum	SumMe

Table 1: Evaluation setting for SumMe. In the case of TVSum, we switch between SumMe and TVSum in the above table.

6.5 minutes. Each video has an annotation of mostly 15 user annotations, with a maximum of 18 users. TVSum contains 50 videos with lengths ranging from 1.5 to 11 minutes. Each video in TVSum is annotated by 20 users. The annotations of SumMe and TVSum are frame-level importance scores, and we follow the evaluation method of (Zhang et al. 2016b). OVP (De Avila et al. 2011) and YouTube (De Avila et al. 2011) datasets consist of 50 and 39 videos, respectively. We use OVP and YouTube datasets for transfer and augmented settings.

### Evaluation Metric

Similar to other methods, we use the F-score used in (Zhang et al. 2016b) as an evaluation metric. In all datasets, user annotation and prediction are changed from frame-level scores to key-shots using the KTS method in (Zhang et al. 2016b). The precision, recall, and F-score are calculated as a measure of how much the key-shots overlap. Let “predicted” be the length of the predicted key-shots, “user annotated” be the length of the user annotated key-shots and “overlap” be the length of the overlapping key-shots in the following equations.

$$P = \frac{\text{overlap}}{\text{predicted}}, R = \frac{\text{overlap}}{\text{user annotated}}, \quad (12)$$

$$\text{F-score} = \frac{2PR}{P + R} * 100\%. \quad (13)$$

### Evaluation Settings

Our approach is evaluated using the Canonical (C), Augmented (A), and Transfer (T) settings shown in Table 1 in (Zhang et al. 2016b). To divide the test set and the training set, we randomly extract the test set five times, 20% of the total. The remaining 80% of the videos is used for the training set. We use the final F-score, which is the average of the F-scores of the five tests. However, if a test set is randomly selected, there may be video that is not used in the test set or is used multiple times in duplicate, making it difficult to evaluate fairly. To avoid this problem, we evaluate all the videos in the datasets without duplication or exception.

### Implementation Details

For input features, we extract each frame by 2fps as in (Zhang et al. 2016b), and then obtain a feature with 1024 dimensions through GoogLeNet pool-5 (Szegedy et al. 2015) trained on ImageNet (Russakovsky et al. 2015). The LSTM input and hidden size is 256 reduced by FC (1024 to 256) for fast convergence, and the weight is shared with each chunk and stride input. The maximum epoch is 20, the learning rate is  $1e-4$ , and 0.1 times after 10 epochs. The weights of the

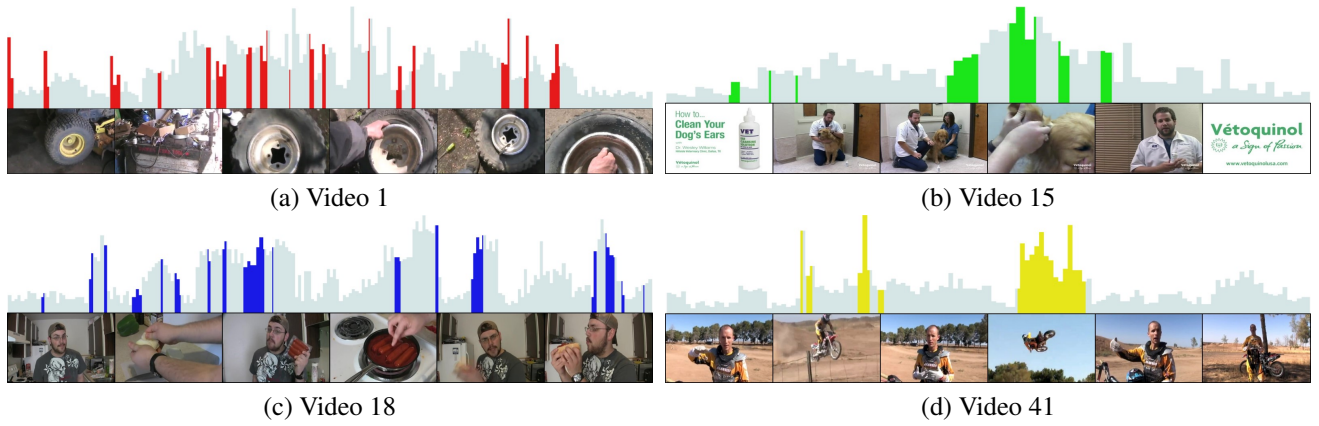


Figure 2: Visualization of which key-shots are selected in the various videos of TVSum dataset. The light blue bars represent the labeled scores. Our key-shots are painted in red, green, blue, and yellow respectively in (a) - (d).

Exp.	CSNet	Difference	Variance Loss	F-score (%)
1				40.8
2	✓			42.0
3		✓		42.0
4			✓	44.9
5	✓	✓		43.5
6	✓		✓	49.1
7		✓	✓	46.9
8	✓	✓	✓	<b>51.3</b>

Table 2: F-score (%) of all cases where each proposed methods can be applied. When CSNet is not applied, LSTM without chunk and stride is used. Variance loss and difference attention can be simply on/off. This experiment uses SumMe dataset, unsupervised learning and canonical setting.

network are randomly initialized.  $M$  in CSNet is experimentally picked as 4. We implement our method using Pytorch.

**Baseline** Our baseline (Mahasseni, Lam, and Todorovic 2017) uses the VAE and GAN in the model of Mahasseni *et al.* We use their adversarial framework, which allows us unsupervised learning. Specifically, basic sparsity loss, reconstruction loss, and GAN loss are adopted. For supervised learning, we add binary cross entropy (BCE) loss between ground truth scores and predicted scores. We also put fake input, which has uniform distribution.

## Quantitative Results

In this section, we show the experimental results of our various approach proposed in the ablation study. Then, we compare our methods with the existing unsupervised and supervised methods and finally show the experimental results in canonical, augmented, and transfer settings. For fair comparison, we quote performances of previous research recorded in (Zhou and Qiao 2018).

**Ablation study.** We have three proposed approaches: CSNet, difference attention and variance loss. When all three methods are applied, the highest performance can be obtained. The ablation study in Table 2 shows the contribution of each proposed method to the performance by conducting experiments on the number of cases in which each method can be applied. We call these methods shown in exp. 1 to exp. 8 CSNet<sub>1</sub> through CSNet<sub>8</sub>, respectively. If any of our proposed methods is not applied, we experiment with a version of the baseline in that we reproduce and modify some layers and hyper parameters. In this case, the lowest F-score is shown, and it is obvious that performance increases gradually when each method is applied.

Analyzing the contribution to each method, first of all, the performance improvement due to variance loss is immensely large, which proves that it is a way to solve the problem of our baseline precisely. CSNet<sub>4</sub> is higher than CSNet<sub>1</sub> by 4.1%, and CSNet<sub>8</sub> is better than CSNet<sub>5</sub> by 7.8%. The variance of output scores is less than 0.001 without variance loss, but as it is applied, the variance increases to around 0.1. Since we use a reciprocal of variance to increase variance, we can observe the loss of an extremely large value in the early stages of learning. Immediately after, the effect of the loss increases the variance as a faster rate, giving the output a much wider variety of values than before.

By comparing the performance with and without the difference attention, we can see that difference attention is well modeled in the relationship between static or dynamic scene changes and frame-level importance scores. By comparing CSNet<sub>1</sub> to CSNet<sub>3</sub>, the F-score is increased by 1.2%. Similarly, CSNet<sub>5</sub> and CSNet<sub>7</sub> are higher than CSNet<sub>2</sub> and CSNet<sub>4</sub> by 1.5% and 2.0%. CSNet<sub>8</sub> is greater than CSNet<sub>6</sub> by 2.2%. These comparisons mean that the difference attention always contributes to these four cases.

We can see from our Table 2 that CSNet also contributes to performance, and it is effective to design the concept of local and global features with chunk and stride while reducing input size of LSTM in temporal domain. Experiments on the number of cases where CSNet can be removed are as follow.



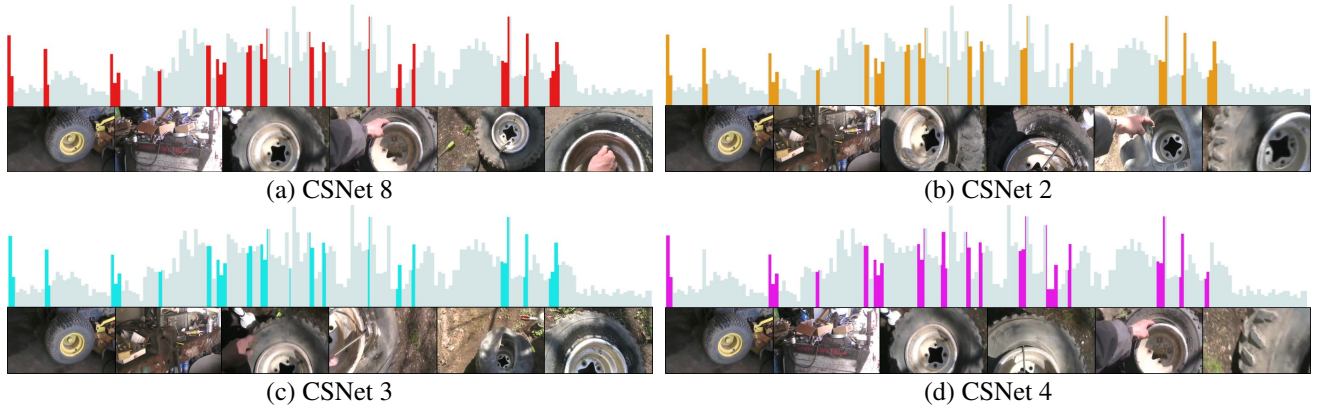


Figure 3: Similar to Fig. 2, key-shots are selected by variants of CSNet denoted in ablation study. A video 1 in TVSum is used.

Method	SumMe	TVSum
K-medoids	33.4	28.8
Vsumm	33.7	-
Web image	-	36.0
Dictionary selection	37.8	42.0
Online sparse coding	-	46.0
Co-archetypal	-	50.0
GAN <sub>dpp</sub>	39.1	51.7
DR-DSN	41.4	57.6
<b>CSNet</b>	<b>51.3</b>	<b>58.8</b>

Table 3: F-score (%) of unsupervised methods in canonical setting on SumMe and TVSum datasets. Our approach outperforms other existing methods. Dramatic performance improvement is shown on the SumMe dataset.

CSNet<sub>2</sub> is better than CSNet<sub>1</sub> by 1.2%, and each CSNet<sub>5</sub>, CSNet<sub>6</sub> outperform CSNet<sub>3</sub>, CSNet<sub>4</sub> by 1.5%, 4.2%. Lastly, CSNet<sub>8</sub> and CSNet<sub>7</sub> have 4.4% difference.

Since each method improves performance as it is added, the three proposed approaches contribute individually to performance. With the combination of the proposed methods, CSNet<sub>8</sub> achieves a higher performance improvement than the sum of each F-score increased by CSNet<sub>2</sub>, CSNet<sub>3</sub> and CSNet<sub>4</sub>. In the rest of this section, we use CSNet<sub>8</sub>.

**Comparison with unsupervised approaches.** Table 3 shows the experimental results for SumMe and TVSum datasets using unsupervised learning in canonical settings. Since our approach mainly target unsupervised learning, CSNet outperforms both SumMe and TVSum over the existing methods (Elhamifar, Sapiro, and Vidal 2012; Khosla et al. 2013; De Avila et al. 2011; Zhao and Xing 2014; Song et al. 2015; Zhou and Qiao 2018; Mahasseni, Lam, and Todorovic 2017). As a significant improvement in performance for the SumMe dataset, Table 3 shows a F-score enhancement over 9.9% compared to the best of the existing methods (Zhou and Qiao 2018).

To the best of our knowledge, all existing methods are

Method	SumMe	TVSum
Interestingness	39.4	-
Submodularity	39.7	-
Summary transfer	40.9	-
Bi-LSTM	37.6	54.2
DPP-LSTM	38.6	54.7
GAN <sub>sup</sub>	41.7	56.3
DR-DSN <sub>sup</sub>	42.1	58.1
<b>CSNet<sub>sup</sub></b>	<b>48.6</b>	<b>58.5</b>

Table 4: F-score (%) of supervised methods in canonical setting on SumMe and TVSum datasets. We achieve the state-of-the-art performance.

scored at less than 50% of the F-score in the SumMe dataset. Evaluation of the SumMe dataset is more challenging than the TVSum dataset in terms of performance. DR-DSN has already made a lot of progress for the TVSum dataset, but for the first time, we have achieved extreme advancement in the SumMe dataset which decreases the gap between SumMe and TVSum.

An interesting observation of supervised learning in video summarization is the non-optimal ground truth scores. Users who evaluated video for each data set are different, and every user does not make a consistent evaluation. In such cases, there may be a better summary than the ground truth which is a mean value of multiple user annotations. Surprisingly, during our experiments we observe that predictions for some videos receive better F-scores than in the results of ground truth. Unsupervised approaches do not use the ground truth, so it provides a step closer to the user annotation.

**Comparison with supervised approaches.** We implemented CSNet<sub>sup</sub> for supervised learning by simply adding binary cross entropy loss between prediction and ground truth to existing loss for CSNet. In Table 4, CSNet<sub>sup</sub> obtains state-of-the-art results compared to existing methods (Gygli et al. 2014; Gygli, Grabner, and Van Gool 2015; Zhang et al. 2016a; 2016b; Zhou and Qiao 2018), but does

Method	SumMe			TVSum		
	C	A	T	C	A	T
Bi-LSTM	37.6	41.6	40.7	54.2	57.9	56.9
DPP-LSTM	38.6	42.9	41.8	54.7	59.6	58.7
GAN <sub>dpp</sub>	39.1	43.4	-	51.7	59.5	-
GAN <sub>sup</sub>	41.7	43.6	-	56.3	<b>61.2</b>	-
DR-DSN	41.4	42.8	42.4	57.6	58.4	57.8
DR-DSN <sub>sup</sub>	42.1	43.9	42.6	58.1	59.8	58.9
HSA-RNN	-	44.1	-	-	59.8	-
CSNet	<b>51.3</b>	<b>52.1</b>	<b>45.1</b>	<b>58.8</b>	59.0	<b>59.2</b>
CSNet <sub>sup</sub>	48.6	48.7	44.1	58.5	57.1	57.4

Table 5: F-score (%) of both unsupervised and supervised methods in canonical, augmented and transfer settings on SumMe and TVSum datasets.

not provide a better performance than CSNet. In general, supervision improves performance, but in our case, the point of view mentioned in the unsupervised approaches may fall out of step with using ground truth directly.

**Comparison in augmented and transfer settings.** We compare our CSNet with other state-of-the-art literature with augmented and transfer settings in Table 5. We can make a fair comparison using the 256 hidden layer size of LSTM used by DR-DSN (Zhou and Qiao 2018), which is a previous state-of-the-art method. We obtain better performance in CSNet than CSNet<sub>sup</sub>, and our unsupervised CSNet performs better than the supervised method in any other approach except for GAN<sub>sup</sub>, which uses 1024 hidden size in TVSum dataset with augmented setting.

## Qualitative Results

**Selected key-shots.** In this section, we visualize selected key-shots in two ways. First, in Fig. 2, selected key-shots are visualized in bar graph form using various genre of videos. (a) - (d) show that many of our key-shots select peak points of labeled scores. In terms of the content of the video, the scenes selected by CSNet are mostly meaningful scenes by comparing colored bars with the images in Fig. 2. Then, in Fig. 3, we compare variants of our approach with a video 1 in TVSum. Although minor differences exist, each approach select peak points well.

**Difference attention.** With a deeper analysis of difference attention, we visualize the difference attention in the TVSum dataset. Its motivation is to capture dynamic information between frames of video. We can verify our assumption that the dynamic scene should be more important than the static scene with this experiment. As shown in Fig. 4, the plotted blue graph is in line with the selected key-shots, which highlight portions with high scores. The selected key-shots are of a motorcycle jump, which is a dynamic scene in the video. As a result, difference attention can effectively predict key-shots using dynamic information.

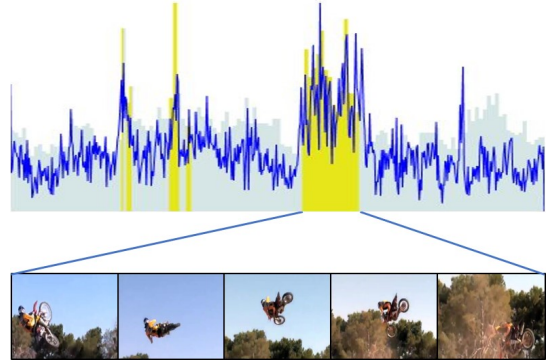


Figure 4: Experiment with video 41 in the TVSum dataset. In addition to the visualization results in Fig. 2, difference attention is plotted with blue color. When visualizing the difference attention, it is normalized to have a same range of ground truth scores. The picture is the video frames which are mainly predicted part with key-shots.

## Conclusion

In this paper, we propose discriminative feature learning for unsupervised video summarization with our approach. Variance loss tackles the temporal dependency problem, which causes a flat output problem in LSTM. CSNet designs a local and global scheme, which reduces temporal input size for LSTM. Difference attention highlights dynamic information, which is highly related to key-shots in a video. Extensive experiments on two benchmark datasets including ablation study show that our state-of-the-art unsupervised approach outperforms most of the supervised methods.

**Acknowledgements** This research is supported by the Study on Deep Visual Understanding funded by the Samsung Electronics Co., Ltd (Samsung Research)

## References

- De Avila, S. E. F.; Lopes, A. P. B.; da Luz Jr, A.; and de Albuquerque Araújo, A. 2011. Vsum: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters* 32(1):56–68.
- Elhamifar, E.; Sapiro, G.; and Vidal, R. 2012. See all by looking at a few: Sparse modeling for finding representative objects. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 1600–1607. IEEE.
- Gong, B.; Chao, W.-L.; Grauman, K.; and Sha, F. 2014. Diverse sequential subset selection for supervised video summarization. In *Proc. of Neural Information Processing Systems (NIPS)*, 2069–2077.
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. of Neural Information Processing Systems (NIPS)*, 2672–2680.
- Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating summaries from user videos. In *Proc.*

- of *European Conf. on Computer Vision (ECCV)*, 505–520. Springer.
- Gygli, M.; Grabner, H.; and Van Gool, L. 2015. Video summarization by learning submodular mixtures of objectives. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 3090–3098.
- Huang, T.; Yang, G.; and Tang, G. 1979. A fast two-dimensional median filtering algorithm. *IEEE Trans. on Acoustics, Speech and Signal Processing. (APSP)* 27(1):13–18.
- Joshi, N.; Kienzle, W.; Toelle, M.; Uyttendaele, M.; and Cohen, M. F. 2015. Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics (TOG)* 34(4):63.
- Kang, H.-W.; Matsushita, Y.; Tang, X.; and Chen, X.-Q. 2006. Space-time video montage. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, volume 2, 1331–1338. IEEE.
- Khosla, A.; Hamid, R.; Lin, C.-J.; and Sundaresan, N. 2013. Large-scale video summarization using web-image priors. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2698–2705.
- Kim, G., and Xing, E. P. 2014. Reconstructing storyline graphs for image recommendation from web community photos. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 3882–3889.
- Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. In *Proc. of Int’l Conf. on Learning Representations (ICLR)*.
- Kopf, J.; Cohen, M. F.; and Szeliski, R. 2014. First-person hyper-lapse videos. *ACM Transactions on Graphics (TOG)* 33(4):78.
- Lee, Y. J.; Ghosh, J.; and Grauman, K. 2012. Discovering important people and objects for egocentric video summarization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 1346–1353. IEEE.
- Liu, D.; Hua, G.; and Chen, T. 2010. A hierarchical visual model for video object summarization. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* 32(12):2178–2190.
- Lu, Z., and Grauman, K. 2013. Story-driven summarization for egocentric video. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2714–2721.
- Mahasseni, B.; Lam, M.; and Todorovic, S. 2017. Unsupervised video summarization with adversarial lstm networks. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2982–2991.
- Ngo, C.-W.; Ma, Y.-F.; and Zhang, H.-J. 2003. Automatic video summarization by graph modeling. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, 104–109. IEEE.
- Poleg, Y.; Halperin, T.; Arora, C.; and Peleg, S. 2015. Egosampling: Fast-forward and stereo for egocentric videos. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 4768–4776.
- Pratt, W. K. 1975. Median filtering. *Semiannual Report, Univ. of Southern California*.
- Pritch, Y.; Rav-Acha, A.; and Peleg, S. 2008. Nonchronological video synopsis and indexing. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* 30(11):1971–1984.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *Int’l Journal of Computer Vision (IJCV)* 115(3):211–252.
- Sharghi, A.; Laurel, J. S.; and Gong, B. 2017. Query-focused video summarization: Dataset, evaluation, and a memory network based approach. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2127–2136.
- Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. Tvsum: Summarizing web videos using titles. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 5179–5187.
- Sun, M.; Farhadi, A.; Taskar, B.; and Seitz, S. 2014. Salient montages from unconstrained videos. In *Proc. of European Conf. on Computer Vision (ECCV)*, 472–488. Springer.
- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 1–9.
- Wei, H.; Ni, B.; Yan, Y.; Yu, H.; Yang, X.; and Yao, C. 2018. Video summarization via semantic attended networks. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*.
- Yang, H.; Wang, B.; Lin, S.; Wipf, D.; Guo, M.; and Guo, B. 2015. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proc. of Int’l Conf. on Computer Vision (ICCV)*, 4633–4641.
- Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016a. Summary transfer: Exemplar-based subset selection for video summarization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 1059–1067.
- Zhang, K.; Chao, W.-L.; Sha, F.; and Grauman, K. 2016b. Video summarization with long short-term memory. In *Proc. of European Conf. on Computer Vision (ECCV)*, 766–782. Springer.
- Zhang, Q.; Xu, L.; and Jia, J. 2014. 100+ times faster weighted median filter. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2830–2837.
- Zhao, B., and Xing, E. P. 2014. Quasi real-time summarization for consumer videos. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2513–2520.
- Zhao, B.; Li, X.; and Lu, X. 2017. Hierarchical recurrent neural network for video summarization. In *Proc. of Multimedia Conference (MM)*, 863–871. ACM.
- Zhao, B.; Li, X.; and Lu, X. 2018. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 7405–7414.
- Zhou, K., and Qiao, Y. 2018. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proc. of Association for the Advancement of Artificial Intelligence (AAAI)*.