Video Summarization using Shot-level Relation-Aware Attention Network (VSRAN)

Anonymous WACV 2023 APPLICATIONS TRACK submission

Paper ID 1162

Abstract

Previous studies on video summarization mainly foucs on extracting key-frames, predicting key-frames based on extracted key-frames features and generate summary using average frame score within a shot. In addition to the complexity of key-frame extraction models, recent studies have raised doubts about the video summarization pipelines by generating video summaries using randomly selected key-frames and comparing them to the SOTA summaries. In this paper, we propose novel shot representation method which best represent motion and static feature of a shot. A relation-aware attention model are employed to fuse 2D and 3D shot features and directly predict shot score. Video summaries are generated based on predicted shot score and key-shot select methods. We verify the performance of the result and compare with SOTA model using two publicly available datasets - SumMe and CoSum.

1. Introduction

Given the rapid increase in the number of videos generated and shared in recent years, a need for models which can summarize videos and retrieve important information becomes more imminent. Considering a video as a sequence of semantically related shots which themselves are sequences of frames that are visually similar, various researchers have tried to represent videos as a combination of shots which contain key-frames[29][20][37][14][15]. There exist different kinds of video summarization e.g. movie, news, spot etc. studied by [31]. In general case, the SOTA methods take advantage of labeled datasets that provide frame-level importance scores such as TVSum[30], and trained to predict frame-level importance scores and extract key-frames[4][34][35]. Predicting frame-level importance scores and having shot boundaries, shot scores are estimated e.g. by averaging frame scores within a shot. Given shot scores and shot lengths, video summary is formed by selecting a subset of shots so that the total summary length is less than or equal to a given limit (e.g. 15% of the video length) and maximizing total summary score (knapsack problem). The effectiveness of the video summarization pipeline described above was challenged in [24] by showing that the randomly generated frame scores can lead to video summaries comparable to the SOTA summaries.

In this paper, we introduce a shot-based video summarization model which directly predicts shot scores based on a novel shot representation in contrast to the previously proposed models which predict frame scores based on individual frame visual features. Our model relies on shot representations that are fusion of static features which represent objects and patterns within the shot and motion features which represent the transition of frames and actions within shots. Learning relation among static features extracted from GoogleNet[32] and motion features extracted from 3D Resnet[12], and maintain the shot sequence order using positional embedding, our model predicts the shot importance scores that can be used for summary generation (see Figure 1). We demonstrate the effectiveness of our model on two publicly available datasets: SumMe[9] and CoSum[2]. Our contributions are four-fold:

1. We propose a video summarization model that predict the shot-level importance scores directly eliminating a need for frame-level importance score analysis. To the best of our knowledge, the proposed model is the first shot-level importance score prediction model.

2. We propose a method for extracting/fusing shot static and motion features

3. We propose and compare shot representation methods that best describe the motion and static feature of the shot.

4. The proposed model outperforms the state-of-the-art models on SumMe dataset in all summarization metrics suggested by [24] including F1-score, Kendall[17] and Spearman[41], and CoSum dataset in mAP metric.

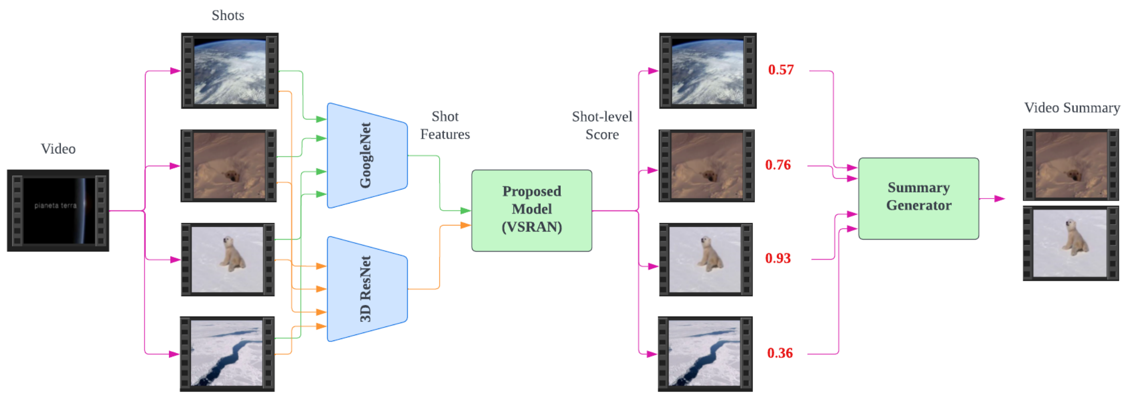

Figure 1: Overview architecture

2. Related work

State-of-the-art video summarization models have following pipeline study by Narwal et al.[23]. Give a video, video is segmented into shot based on its visual similarity. To select key-shots, frames score are estimated according to their importance and average those frames within a shot to get shot score. Summaries are formed using summarization technique e.g. 0/1 knapsack algorithm to get the final video summary. The most challenge part of this pipeline is to select the most important key-shots. To estimate the importance of shots, the state-of-the-art method can be categorize into two groups - supervised and unsupervised.

2.1. Supervised Video Summarization

Given the input frame sequence, supervised video summarization approaches predict the corresponding summary with human-annotated ground truth. Fei et al. [5] declaim that image memorability can achieve a interesting video summary. The memorable summary is sensitive to faces, people or central objects. They trained their model with labeled human memorability score image dataset given by [18]. Predicting memorability score and entropy value for each frames. Video summary is constituted by selecting highest memorability score and entropy value. Beside memorability, some researcher define video summarization as a sequence to sequence problem. Leveraging the experience from natural language processing (NLP), previous methods utilized encoder-decoder architecture to model the temporal dependency according to the ground truth annotation. Zhang et al. [34] utilize the characteristic of bidirectional LSTM, which can consider whole video frame sequence forward and backward, to predict if a frame be a part of summary or not. Rochan et al. [28] proposed an alternative way using convolution neural networks to integrate video frames to generate summaries.

In order to address the gradient vanishing problem and understanding the structure of videos, Zhao et al. [37], [38] and Zhang et al. [35] start to take the structure of videos into account and introduce structure-adaptive model based on hierarchical LSTM to exploit video structures, merging shot information into model while training.

Vaswani et al.[33] shows self-attention perform well in translation problem. Video summarization on the other hand, can be considered as sequential problem which similar to NLP translation problem. Fajtl et al. [4] employ self-attention model to tackle the supervised video summarization problem. Original features are mapped into query, key and value. Attented feature are the weighted value of query and key. Regression network are attached followed by self-attention model.

2.2. Unsupervised Video Summarization

Unlike supervised video summary that require human annotation as label for training model, unsupervised video summarization do not involve label while training. Hannane et al.[11] proposed Mean Shift-based Keyframes for Video Summrization(MSKVS) algorithm. They represent frame using GFFV descriptor which is invariant to scale, illumination, noise and other external factors. After elimination redundant frames, mean shift algorithm are applied to extract key-frames of video and form the final summary. Parihar et al. [26] propose a pipeline for multiview video summarization. BIRCH algorithm are applied first to reduce unnecessary frames. Jaccard and Dice similarity are used to measure the similarity between frames and determinate shot boundaries. Multi-level K-means clustering is applied to extract keyframes of each key event in a video then form the final summary. Mahasseni et al. [21] proposed SUM-GAN for unsupervised video summarization. They claim good summary should reconstruct original video seamlessly. Inspired by Generative Adversarial

Networks(GAN) proposed by Goodfellow et al. [8], Mahasseni et al. apply GAN model in video summarization task. They use LSTM model to estimate the importance of frame. The importance of frame are multiply with frame features to generate summary. Another LSTM model is used to reconstruct video from summary. Finally, the third LSTM model is use to determine the input is from origin video or summary. Jung et al. [14][15] modify SUM-GAN frame importance part with bidirectional LSTM and self-attention with relative positional embedding. Unlike Mahasseni et al. fit frame sequence in time order, Jung et al. split frames into multiple chunks and strides. Chunks separate frames into equal length which are consider as local feature. Strides take frames with equal step which are consider as global feature. The frame score are weight sum of chunks, strides and the difference between frames. Apostolidis et al.[1] modified SUM_GAN proposed by [21] with stepwise SUM_GAN. Instead of update the whole weight simultaneously, Apostolidis et al. use four step to train their SUM_GAN model. In addition, they replace AutoEncoder with Variational AutoEncoder which learn the distribution of feature distribution and further add Attention block between encoder and decoder(called AAE). By adding attention block, correction between current encoder input and previous hidden state of decoder can be computed which benefit the video summarization.

3. Model

3.1. Shot Feature selection

Previous video summarization framework, fit frame-level feature to their model and estimate frame importance score. In this paper, on the other hand, using shot-level feature to predict shot importance score. Unlike frame-level prediction, there are extra preprocessing step to predict shot score - shot feature selection. We proposed three shot feature selection (Mean, Max Probability and Center) scheme to represent shot.

3.1.1 Mean shot feature selection

Given a video with frame sequence f = $\{f_1, \ f_2,...,f_T\}$, kernel temporal segmentation (KTS) which proposed by [27] are used to determine the shot boundaries $S = \{s_1, s_2, ...s_M\}$. For static mean shot features selection $X^s_m = \{X^s_{m1}, X^s_{m2}, ...X^s_{mM}\}$, are the average of all frame features within a shot. The frame features are extracted using GoogleNet[32] pretrained with ImageNet dataset[3]. The frame is fed into the network and the feature is extracted from pool5 layer with dimension of 1×1024.

For motion mean shot features selection, pre-trained 3D Resnet[12] are used to extracted feature from non-overlapped 16-frames clips. Clip feature is extracted from pool5 layer of 3D Resnet with dimension 1×2048. Motion mean shot feature$X^m_m = \{X^m_{m1}, X^m_{m2}, ...X^m_{mM}\}$ are the average of clip features within a shot. For those clip less than 16 frames, zeros are padded to fill the gap.

Figure 2: Max Probability plot, orange line indicate the highest probability ratio, blue bars are interval between two standard deviation. This plot shows feature extraction model have higher confidence in selected feature.

3.1.2 Max probability feature selection

Max probability feature is the single feature which has highest classification probability after final classification layer in feature extraction model. Given a set of features in a shot X $\in R^{k\times dim}$ and feature probabilities p $\in R^{k\times n_{class}}$, we find the max probability among each feature. The max probability shot feature is the feature with highest probability. Algorithm 1 describe the algorithm for max probability feature selection.

For static feature, GoogleNet was original for image classification. The classification layer has 1000 output. Let P^s_i denoted the set of probability in a k frames shot ,where $P^s_i \in$R$^{k\times1000}$. $X^p_i = \{x^p_{si1}, x^p_{si2}, ...x^p_{sik}\}$ are the set of features in this shot. $P^s_{maxi} = Max(P^s_i) \in$R$^{1\times k}$. are the maximum probability among k frame. $Argmax$ is the location of those k frames probability. The static max probability feature of this shot is $x^p_{siArgmax} \in$R$^{1\times1024}$.

For motion feature, 3D Resnet was trained on motion classification. It has 400 output in its final layer. Let P^m_i $\in$R$^{k\times400}$ denote the set of probability in a k clips shot. $X^p_i = \{x^p_{mi1}, x^p_{mi2}, ...x^p_{mik}\}$ are the set of features in this shot.$P^m_{maxi} = Max(P^m_i) \in$R$^{1\times k}$. are the maximum probability among k clips. $Argmax$ is the location of those k clips probability. The motion max probability feature of this shot is $x^p_{miArgmax} \in$R$^{1\times1024}$.

Algorithm 1 Max Probability shot feature selection

Input: Shot features $X \in R^{k \times dim}$
Input: Feature probability $P \in R^{k \times n_{class}}$
Output: Max prob feature x_p

```
maxFeatureClasses ← []
for p_i ∈ P do
    maxClass ← max(p_i)
    maxFeatureClasses ← maxFeatureClasses ∪ maxClass
end for
maxIndex ← argmax(maxClass)
x_p ← X[maxIndex]
return x_p
```

3.1.3 Center shot feature selection

Center shot feature is the single feature closest to the features cluster center. Given a set of features in a shot X $\in R^{k \times dim}$, we first average those feature $x_m \in R^{1 \times dim}$. For each feature, calculate the Euclidean distance $Dist(,)$ from average mean feature x_m to that feature. The center feature $x_c \in R^{1 \times dim}$ is the feature closest to the mean feature. Algorithm 2 describe the algorithm for center shot feature selection.

For static center shot feature selection$X^s_c = \{X^s_{c1}, X^s_{c2}, ...X^s_{cM}\}$, all frame features from GoogleNet pool5 layer group into one cluster, the frame feature closest to the center is selected as the static center shot feature.

For motion center shot feature selection$X^m_c = \{X^m_{c1}, X^m_{c2}, ...X^m_{cM}\}$, non-overlapped 16-frames clip feature extracted from 3D Resnet also group into one cluster, the clip feature closest to the center is chose as the motion center shot feature.

Algorithm 2 Center shot feature selection

Input: Shot features $X \in R^{k \times dim}$
Output: Center feature x_c

```
Mean Feature x_m ← Mean(X) ∈ R^{1×dim}
miniDist = ∞
for x_i ∈ X do
    dist ← Dist(x_m, x_i)
    if dist < minDist then
        minDist ← dist
        Center feature x_c ← x_i
    end if
end for
return x_c
```

3.1.4 Uniform sampling shot feature selection

3D Resnet proposed by Hara et al. [12] required 16 frames per clip as input. In they research, they select frames from shot uniformly. Followed by [12], give a shot with n frames, shot motion feature is extract from the clip $c = \{f_0, f_{\lfloor \frac{1n}{16} \rfloor}, f_{\lfloor \frac{2n}{16} \rfloor} ... f_{\lfloor \frac{15n}{16} \rfloor}\}$ where f_i indicate the i-th frame in a shot. Clip c then fit into 3D Resnet to extract motion feature of this shot.

3.2. Relation-Aware Attention

The architecture of the proposed model is shown in Figure3. The bottom layer extract shot-level static features $S^s = \{S^s_1, S^s_2, ..., S^s_M\} \in R^{M \times dim_s}$ and motion features $S^m = \{S^m_1, S^m_2, ..., S^m_M\} \in R^{M \times dim_m}$ from video as we mention in the section above. In the SumMe dataset, key-shots are given with start and end frame index. In the case where key-shots are not given, as is in the case for TVSum, we follow the previous works[4], [13], [21], [28], [35], [34], [39], and we apply kernel temporal segmentation (KTS), which is proposed by [27] for shot segmentation.

The middle part, which is the core part of the proposed model. The cross attention model learn the relationship between motion and static feature which is named as relation-aware attention. The static features and motion features are first embedded with sinusoid positional information. Motion features are fused with static features using cross attention mechanism. The attented feature z_i for i-th shot is calculated as eq.1.

$$z_i = Attention(x^s_i, X^m), i \in [1, M] \tag{1}$$

Based on the vanilla attention which is original proposed by Vaswani et al.[33]. Given then fed toward dropout layer and layer normalization to be the output of the relation-aware attention model, $Z = \{z_1, z_2, ..., z_M\} \in R^{M \times dim_s}$. Given the shot-level static feature $x^s_i \in R^{dim_s}$ and motion feature sequence $X^m \in R^{M \times dim_m}$, the attented feature is obtained as:

$$Attention(x^s_i, X^m) = \sum_{j=1}^{M} \alpha_{i,j}(W_v, x^m_j) \tag{2}$$

where α_{ij} denotes the attention weight for x^s_i and each x^m_j. W_Q, W_K, W_V denotes the weight matrix of the query, key and value.

$$\alpha_{ij} = \frac{exp((W_Q x^m_i)(W_K x^s_j)^T)}{\sum_{j=1}^{M} exp((W_Q x^m_i)(W_K x^s_j)^T)} \tag{3}$$

3.3. Positional Embedding

Positional embedding add sinusoidal positional information to input sequence with different frequency. Eq. 4 describe the positional embedding followed by [33].

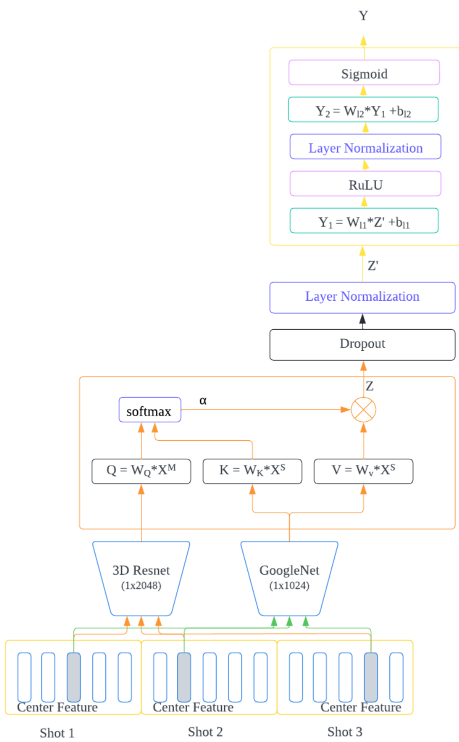

Figure 3: Overall architecture of ensemble network using relation-aware attention.

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/d_{model}}})$$
$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/d_{model}}}) \quad (4)$$

3.4. Regression Networks

The attented features $Z \in \mathrm{R}^{M \times dim_s}$ fed toward a dropout layer and layer normalization. $Z' \in R^{M \times dim_s}$ is the output of those procedure is then fed into regression network for final shot-level importance score prediction.

$$Z' = LayerNoem(Dropout(Z)) \quad (5)$$

The regression network consist of two linear transformations with ReLU activation, layer normalization in between and sigmoid activation behind.

$$y_t = Sigmoid(W_{l2} \cdot LayerNorm$$
$$(ReLU(W_{l1} \cdot z + b_{l1})) + b_{l2}), t \in [1, M] \quad (6)$$

where W_{l1}, W_{l2} and b_{l1} are weight matrices and bias for the regression network.

3.5. Key-Shot Selection

Previous work[4],[9], [13], [21], [28], [35], [34], [39] for key-shot selection aim to maximize the shot-level importance score in final summary. Given the predicted importance score y_t for each shot, the predicted summary is generated by solving the 0/1 knapsack problem as :

$$max \sum_{i=1}^{M} y_i u_i, s.t. \sum l_i u_i \leq L, u_i \in 0, 1 \quad (7)$$

where M is the number of shots, l_i is the length of the i-th shot, and L is the knapsack capacity, 15% of original video length followed by previous works [4],[9], [13], [21], [28], [35], [34], [39].

In this paper, we found that knapsack tent to select the shot with shorter period. It may fall to select the shot with high score and long period. To solve this issue, we introduce greedy algorithm for key-shot selection. Shot are first sorted based on its predicted score in descend order. Select the top shot until the summary length meet to length limitation. Algorithm 3 described the greedy key-shot selection.

Algorithm 3 Greedy Key-shot selection

Input: Shot scores $Score \in R^{1 \times M}$, Shots S
Summary length l
Output: Summary

```
1: Summary ← []
2: Sorted Index ← Argsort(Score)
3: for index ∈ Sorted index do
4:     Summary ← Summary ∪ Shots[index]
5:     if Length of Summary > l then
6:         return Summary
7:     end if
8: end for
```

4. Experiments

We evaluated the performance of the proposed model on two datasets, SumMe[9] and CoSum[2]. The overview of two datasets is show in Table 1. Note that previous works [4], [13], [28], [34], [39] for frame-level video summarization, all video are downsampled with sample rate $\frac{1}{15}$.

4.1. Hyperparameters

We train the proposed model with the initial learning rate of 1×10^{-5} and L2 regularization of 5×10^{-5} using Adam[19] as the optimizer. The batch size is set to 1, the number of epochs is 150 and dropout rate is set to 0.1. Mean squared error(MSE) are roles as loss. We apply k-fold cross validation with k set to 5 for SumMe and TvSum Dataset. For CoSum Dataset k is set to 4. All weight matrices are

Dataset	#Video	Annotation Type	#Annotation	#Video Length
SumMe	25	key-shots	15-18	$32 - 324_{(sec)}$
CoSum	51	key-shot	3	

Table 1: SumMe and CoSum Dataset

initialized with Xavier uniform[6] and biases are initialized as 0.1

4.2. Features

In this paper, we use two different kinds of feature, static feature and motion feature. The static feature is extracted from the pool5 layer of GoogleNet (Inception v1) [32] with dimension of 1024. GoogleNet model was pre-trained on ImageNet dataset[3].

The motion feature is extracted from the pool5 layer of 3D Resnet [12] with dimension of 2048. 3D Resnet model was pre-trained on Kinetics dataset [16].

4.3. Evaluation Metrics

Given a predicted summary S_{pred} and the corresponding ground truth summary S_{ground}, the precision and recall are compute as:

$$precision = \frac{|S_{pred} \cap S_{ground}|}{|S_{pred}|} \tag{8}$$

$$recall = \frac{|S_{pred} \cap S_{ground}|}{|S_{ground}|} \tag{9}$$

The F1-score is the harmonic mean of precision and recall written as follow:

$$F1score = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

Following the previous works [4], [21], [34], [39], for each video, the predicted summary is compared with each user annotated summary. Final result is reported as the average F1-score in TvSum Dataset and the maximum F1-score in SumMe Dataset. For CoSum Dataset, followed by [2], the ground truth summaries is a set of shots which selected by at least two annotators. The F1-score report in CoSum is compare predicted summary with ground truth summary.

In addition to F1-score metrics, Otani et al. [24] suggest used Kendall[17] and Spearman[41] coefficient to evaluate score prediction model. F1-score measure the overlap between predicted summary and annotated summary. It may be affect by key-shot selection algorithm e.g. knapsack. They shows that in frame-level importance score prediction, randomly generated score perform on par F1-score result compare with well-establish method. Kendall and Spearman on the other hands, measure mainly on score prediction. The kendall coefficient is calculated as follow:

$$kendall = \frac{n_c - n_d}{n} \tag{11}$$

where n_c, n_d denoted as number of concordant pairs and number of discordant pairs respectively, n denoted as total pairs.

The Spearman rank correlation coefficient[41] is calculated as follow:

$$\rho = 1 - \frac{6\sum_{i=0}^{M} d_i^2}{M(M^2-1)} \tag{12}$$

where d_i is the distance between two rank, M is number of shot in this case. The result of Kendall and Spearman in this paper are compare predict shot-level importance score with ground truth shot-level importance score.

We follow [2] used mean average precision(mAP) to evaluate the performance of our model in CoSum dataset.

5. Result

We compared our shot-level video summarization with other supervised state-of-the-art frame-level video summarization model.

5.1. Summary based evaluation

F1-score eq. 10 is widely used to evaluate the quality of video summarization model. It reflect the proportion of overlapping between ground truth summaries and predicted summary. Table 2 shows the result of SOTA and our model summary performance in SumMe dataset. In standard setting, we use one dataset and involved full-fold training. In transfer setting, model are trained using all videos in TvSum, OVP and YouTube datasets and evaluate all videos in SumMe dataset. In augment setting, model are augmented using all videos in TvSum, OVP and Youtube then trained with full-fold in SumMe dataset.

5.2. Shot score prediction based evaluation

SOTA model predict frame importance scores based on frame features. Video summaries are formed by selecting shot that are able to maximize the shot score using dynamic programming algorithm e.g. 0/1 knapsack. This pipeline are challenged by researcher [24]. In addition to F1-score that measure the quality of machine summaries, they suggest Kendall [17] and Spearman [41] coefficient metrics to

Method	Standard	Transfer	Augment
Random Frame level	41.0		
Random Shot level	34.7		
SUM-GAN[14]	41.7	43.6	
VASNet[4]	43.4	42.5	41.9
RSGN[36]	45.0	45.7	44.0
Clip-it[22]	52.5	**54.7**	50.0
Proposed VSRAN	**57.7**	45.5	**54.7**
ground truth	64.7		

Table 2: F1-score comparsion on SumMe dataset with state-of-the-art

Method	Kendall	Spearman
Random	0.0	0.0
DR-DSN[40]	0.047	0.048
RSGN[36]	0.083	0.085
Proposed VSRAN	**0.104**	**0.123**

Table 3: Kendall and Spearman coefficient comparison on SumMe dataset with state-of-the-art methods

evaluate the performance of machine score prediction. Table 3 shows the comparison of Kendall and Spearman coefficient in SumMe dataset.

This result confirm the question from [24] that frame-level score estimate is not required for generate summaries and our result is comparable to other frame-level video summarization model.

5.3. CoSum evaluation

CoSum dataset proposed by Chu et al. [2], they suggest mean average precision(mAP) as evaluation metric. To be fair comparsion, we follow [2] using mAP as evaluation metric to evaluate out proposed VSRAN model on CoSum dataset. According to [2], CoSum dataset has 10 categories that query from Youtube. Table 4 show our VSRAN mAP result on CoSum dataset amount 10 categories. Table 5 shows the mAP result of state-of-the-art methods and our proposed model on CoSum dataset.

6. Ablation Study

In this paper, we study different shot feature selection, key-shot selection method and Effect on positional embedding.

6.1. Ablation study: Shot feature selection

For shot feature selection, we use mean, max probability, uniform sampling and center shot feature selection which we introduced in section3.1. We use SumMe dataset and all metrics including F1-score for measuring video summary and Kendall and Spearman for measuring score prediction to show the effect of shot feature selection. Table 6 shows the performance of different shot feature selection method. Based on the result shows in table 6, we can see center feature has the best performance in every evaluation metrics. Uniform sampling feature selection selecting the frames in equal step. When it deal with longer shot, transition of frames vanish due to long time period. Mean feature selection average features which destroy the feature. Center feature on the other hand, select the feature closest to the center, which generally describe the transition of frames and objects in the shot.

6.2. Ablation study: Key-shot selection

In this section, we will compare two kinds of key shot selection algorithm. For knapsack algorithm, it will select the shots which can generate highest score in summary. Because of the length limitation problem, knapsack tent to select the shot with short period. Greedy algorithm select the shot from the highest score till it meet the length limitation. It may not guarantee the summary has highest score, but shot with higher score will be included in summary. Since SumMe dataset provide user summary for each videos that are able to directly evaluate the performance of final machine summary and CoSum dataset is able to generate ground truth summary based the suggestion given by [2], we use SumMe and CoSum dataset to show the effect of two key-shot selection methods. Table 7 shows the F1-score result in SumMe and CoSum dataset.

Based on table 7, greedy algorithm perform better than knapsack. The reason of this result is that, knapsack may skip the shot with higher score but longer period. Figure 4 show that, some shot with higher score knapsack do not pick it. Because when selecting those longer shot, the summary may not has highest score. In order to maximize the summary score, shorter shot is the better choose.

7. Conclusion

Given the similarities among frames within video shots, we proposed a model for video summarization that predict shot scores in contrast to previous studies which predict

Category	Base	Bike	Effiel	Kid	MLB	NFL	Notre	Statue	Surf
VSRAN	0.66	0.73	1	0.91	0.647	0.9	1	0.688	0.713

Table 4: CoSum Category result

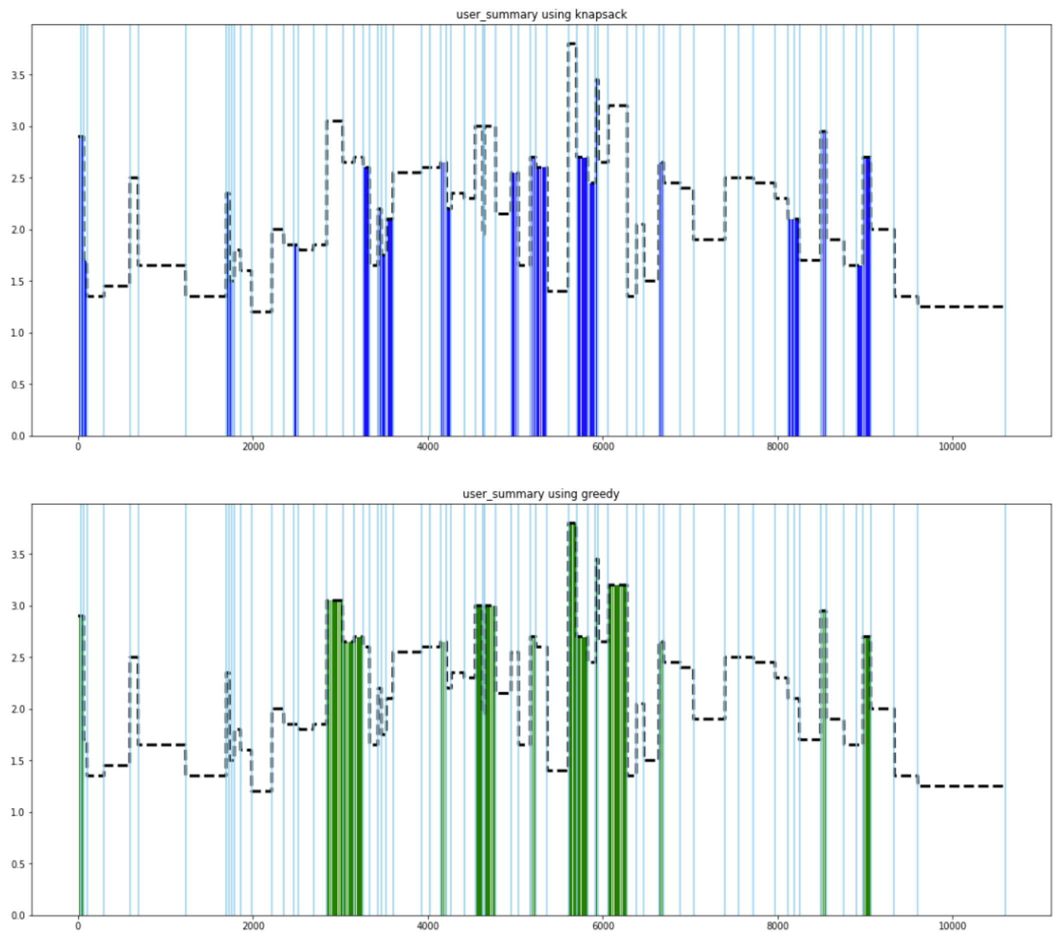

Figure 4: Compare the shot pick by knapsack and greedy. The black dash line indicate the predict shot score. The blue bars are the shot selected by knapsack algorithm and the green bars are the shots selected by greedy algorithm

Method	mAP-top5	mAP-top15
KTS [27]	0.684	0.686
seqDPP [7]	0.692	0.709
SubMod [10]	0.735	**0.745**
DeSumNet [25]	0.721	0.736
Proposed VSRAN	**0.792**	0.676

Table 5: CoSum result compare to SOTA methods

Feature selection	F1-score	Kendall	Spearman
Uniform Sampling	44.5	0.097	0.115
Mean	49.6	0.084	0.1
Max Probability	56.3	0.068	0.08
Center	**57.7**	**0.104**	**0.123**

Table 6: Ablation study on shot feature selection

Key-Shot selection	SumMe[9]	CoSum[2]
Knapsack	53.1	52.1
Greedy	**57.7**	**54.8**

Table 7: Ablation study on Key-shot selection

frame scores. Taking advantage of pre-trained 2D CNN and 3D CNN models, we compared different shot representation variants and proposed a method for shot feature extraction. The comprehensive evaluation of the model on SumMe and CoSum datasets and the ablation studies demonstrated that shot-level importance score can lead to the SOTA summaries. Furthermore, the comparison of different summary generation methods verified the concerns raised previously on the performance of the Knapsack summary generator compared to greedy summary generators. As shown in one of the ablation studies, we are collecting datasets from veriety of fields to analyze the performance of the generic summarizer on larger datasets and eventually augment the training data and improve model performance.

References

[1] Evlampios Apostolidis, Eleni Adamantidou, Alexandros I Metsai, Vasileios Mezaris, and Ioannis Patras. Unsupervised video summarization via attention-driven adversarial learning. In *International Conference on multimedia modeling*, pages 492–504. Springer, 2020. 3

[2] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *CVPR*, 2015. 1, 5, 6, 7, 8

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 3, 6

[4] Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In *Asian Conference on Computer Vision*, pages 39–54. Springer, 2018. 1, 2, 4, 5, 6, 7

[5] Mengjuan Fei, Wei Jiang, and Weijie Mao. Memorable and rich video summarization. *Journal of Visual Communication and Image Representation*, 42:207–217, 2017. 2

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 6

[7] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. *Advances in neural information processing systems*, 27, 2014. 8

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[9] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014. 1, 5, 8

[10] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3090–3098, 2015. 8

[11] Rachida Hannane, Abdessamad Elboushaki, and Karim Afdel. Mskvs: Adaptive mean shift-based keyframe extraction for video summarization and a new objective verification approach. *Journal of Visual Communication and Image Representation*, 55:179–200, 2018. 2

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017. 1, 3, 4, 6

[13] Zhong Ji, Kailin Xiong, Yanwei Pang, and Xuelong Li. Video summarization with attention-based encoder–decoder networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1709–1717, 2019. 4, 5

[14] Yunjae Jung, Donghyeon Cho, Dahun Kim, Sanghyun Woo, and In So Kweon. Discriminative feature learning for unsupervised video summarization. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 33, pages 8537–8544, 2019. 1, 3, 7

[15] Yunjae Jung, Donghyeon Cho, Sanghyun Woo, and In So Kweon. Global-and-local relative position embedding for unsupervised video summarization. In *European Conference on Computer Vision*, pages 167–183. Springer, 2020. 1, 3

[16] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 6

[17] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945. 1, 6

[18] Aditya Khosla, Akhil S Raju, Antonio Torralba, and Aude Oliva. Understanding and predicting image memorability at a large scale. In *Proceedings of the IEEE international conference on computer vision*, pages 2390–2398, 2015. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Dong Liu, Nagendra Kamath, Subhabrata Bhattacharya, and Rohit Puri. Adaptive context reading network for movie scene detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9):3559–3574, 2020. 1

[21] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017. 2, 3, 4, 5, 6

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

[23] Pulkit Narwal, Neelam Duhan, and Komal Kumar Bhatia. A comprehensive survey and mathematical insights towards video summarization. *Journal of Visual Communication and Image Representation*, 89:103670, 2022. 2

[24] Mayu Otani, Yuta Nakashima, Esa Rahtu, and Janne Heikkila. Rethinking the evaluation of video summaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7596–7604, 2019. 1, 6, 7

[25] Rameswar Panda and Amit K Roy-Chowdhury. Collaborative summarization of topic-related videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7083–7092, 2017. 8

[26] Anil Singh Parihar, Joyeeta Pal, and Ishita Sharma. Multi-view video summarization using video partitioning and clustering. *Journal of Visual Communication and Image Representation*, 74:102991, 2021. 2

[27] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014. 3, 4, 8

[28] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 347–363, 2018. 2, 4, 5

[29] Daniel Rotman, Yevgeny Yaroker, Elad Amrani, Udi Barzelay, and Rami Ben-Ari. Learnable optimal sequential grouping for video scene detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1958–1966, 2020. 1

[30] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015. 1

[31] MU Sreeja and Binsu C Kovoor. Towards genre-specific frameworks for video summarisation: A survey. *Journal of Visual Communication and Image Representation*, 62:340–358, 2019. 1

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

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4

[34] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016. 1, 2, 4, 5, 6

[35] Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 383–399, 2018. 1, 2, 4, 5

[36] Bin Zhao, Haopeng Li, Xiaoqiang Lu, and Xuelong Li. Reconstructive sequence-graph network for video summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2793–2801, 2021. 7

[37] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hierarchical recurrent neural network for video summarization. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 863–871, 2017. 1, 2

[38] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7405–7414, 2018. 2

[39] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 4, 5, 6

[40] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 7

[41] Daniel Zwillinger and Stephen Kokoska. *CRC standard probability and statistics tables and formulae*. Crc Press, 1999. 1, 6