University of Information Technology – Vietnam National University HCMC
Faculty of Computer Science

Artificial Intelligent

Video Summarization with DSN and Reinforcement Learning

Lecturer: PhD. Hoang Ngoc Luong

CS106.M21.KHTN

Members

1. Nguyen Duy Dat – 20520435

2. Le The Viet– 20520093

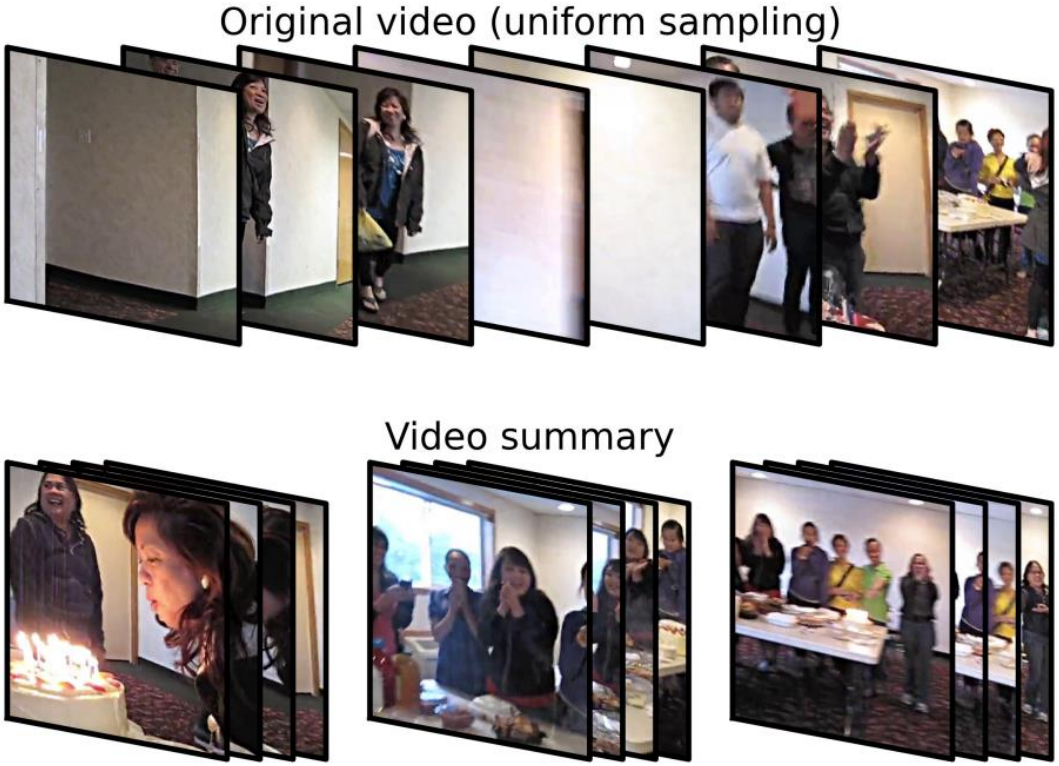
3. Le Doan Phuc Minh - 20520243

1. Introduction
2. Methods
3. Experiments

Introduction

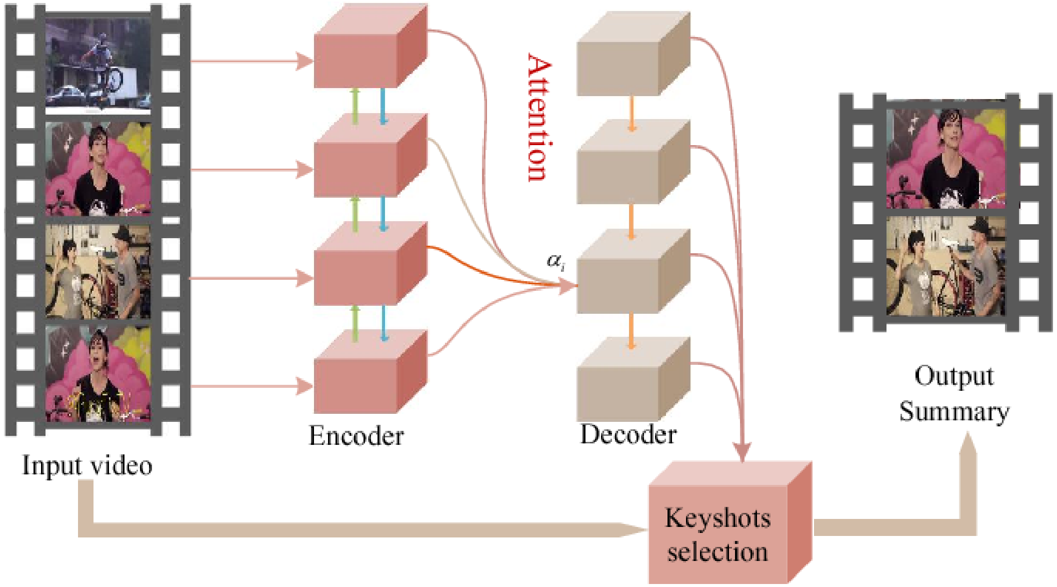
Introduction

Due to the exponential growth of the number of online videos in recent years, research in video summarization has gained increasing attention



Zhang et al. - DPP-LSTM

DPP-LSTM: A deep architecture that combines a bidirectional long short-term memory LSTM network with a Determinantal Point Process (DPP) module that enhances diversity in summaries. Proposed by Zhang et al.



An example of LSTM Model in Video Summarization

Zhang et al. - DPP-LSTM

Although it have good efficiency, there are some problems:

- Two-stage training is required
- Supervised learning cannot completely explore the potential of deep networks

Mahasseni et al. - DPP-LSTM

In 2017, Mahasseni et al. developed a framework to train DPP-LSTM. This DPP-LSTM selects keyframes and uses a discriminator network to decide whether a synthetic video constructed by the keyframes is real or not while learning.

This not only make this method unsupervised, but it also enforces DPP-LSTM to select more frames that are representative.

Mahasseni et al. - DPP-LSTM

There are some problems:

- Training session is unstable due to the adversarial essence
- In the matter of increasing diversity, DPP-LSTM cannot maximize from the DPP module without labels
- Require multiple training stages (Not Efficient)

Architecture of DSN

Architecture of a deep summarization network (DSN):

- Encoder: extracts features on video frames using a convolutional neural network (CNN)
- Decoder: calculates probabilities based on which actions are sampled for selecting frames using bidirectional LSTM network

DSN Training

For DSN training, we proposed an end-to-end RL-based framework with a diversity representativeness (DR) reward function and calculate generated summaries based on diversity and representativeness.

Goal: Maximize the expected rewards over time

*RL: Reinforcement Learning

DR Reward Function

The DR reward function includes a diversity reward and a representativeness reward.

- The diversity reward measures how different the selected frames are when compared to other frames.
- The representativeness reward computes distances between frames and their nearest selected frames (the k-medoids problem)

Why Reinforcement Learning ?

- RNN is used as part of model and focus on the unsupervised setting. Due to the fact rewards only can be collected after computing over the entire video sequence, RL is an inevitable choice.

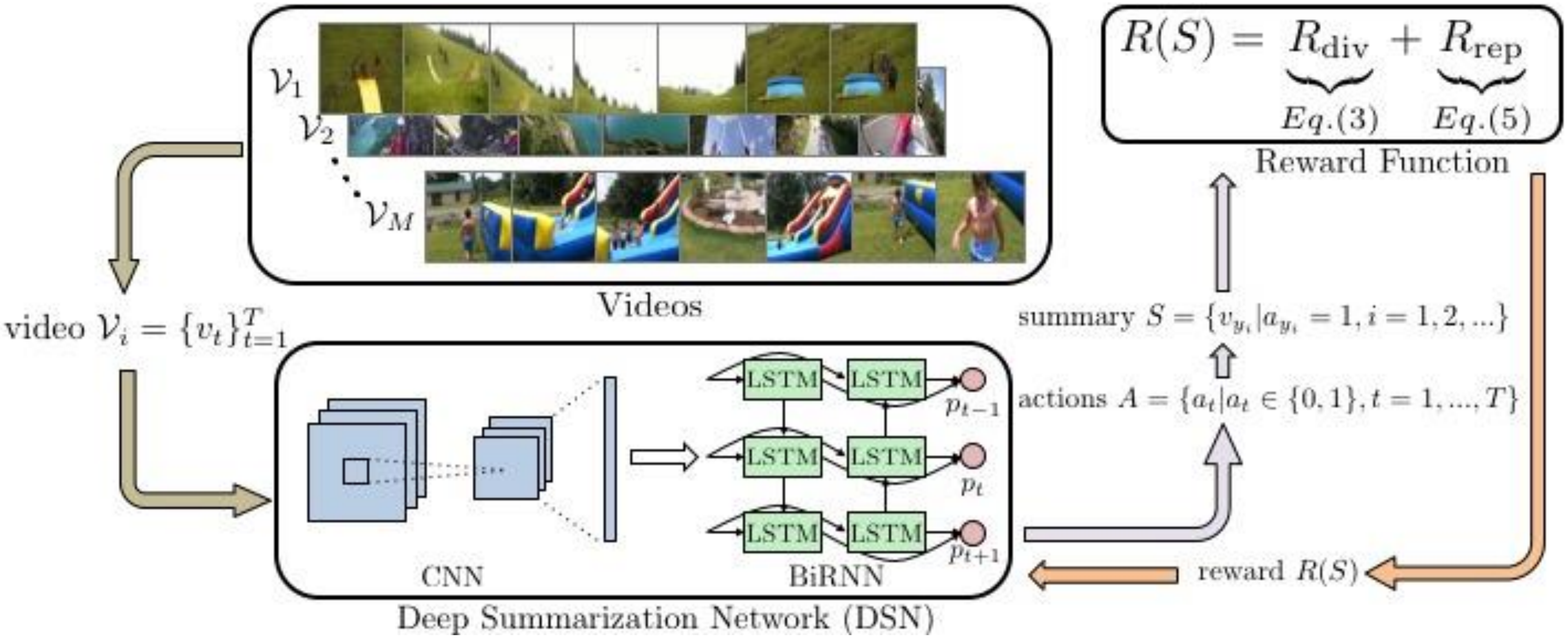
- Thanks to iteratively enforcing the agent to take better and better actions, RL can optimize the action (frame-selection) mechanism of an agent.

Testing Datasets

- SumMe (Gygli et al. 2014)
- TVSum (Song et al. 2015)

Methods

DSN Architecture


$R(S) = \underbrace{R_{\text{div}}}_{Eq.(3)} + \underbrace{R_{\text{rep}}}_{Eq.(5)}$
Reward Function
$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_M$
Videos
video $\mathcal{V}_i = \{v_t\}_{t=1}^{T}$
summary $S = \{v_{y_i} | a_{y_i} = 1, i = 1, 2, ...\}$
actions $A = \{a_t | a_t \in \{0, 1\}, t = 1, ..., T\}$
LSTM
LSTM
p_{t-1}
LSTM
LSTM
p_t
LSTM
LSTM
p_{t+1}
CNN
BiRNN
reward $R(S)$
Deep Summarization Network (DSN)

Deep Summarization Network (DSN)

Sigmoid function guarantee every pt has a value in the range of [0;1]:

$$p_t = \sigma\left(W h_t\right),$$

$$a_t \sim \mathrm{Bernoulli}\left(p_t\right),$$

σ: sigmoid function

W: the weight of the state hide ht

at ∈ {0, 1}

Diversity-Representativeness Reward Function

R(S) = R_div + R_rep

Diversity Reward

$$R_{\text{div}} = \frac{1}{|\mathcal{Y}|(|\mathcal{Y}|-1)} \sum_{t \in \mathcal{Y}} \sum_{\substack{t' \in \mathcal{Y} \\ t' \neq t}} d\left(x_t, x_{t'}\right),$$

$$\left[\begin{aligned} &\gamma = \{v_{y_i} | a_{y_i} = 1, i = 1, \ldots, |\gamma|\} \\ &d\left(x_t, x_{t'}\right) = 1 - \frac{x_t^T x_{t'}}{\left\|x_t\right\|_2 \left\|x_{t'}\right\|_2}. \\ &|t - t'| > \lambda, \end{aligned}\right.$$

Representativeness Reward (K-medoids)

$$R_{\text{rep}} = \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \min_{t' \in \mathcal{Y}} \left\|x_t - x_{t'}\right\|_2\right).$$

Training with Policy Gradient

Maximizing the expected rewards with a policy function πθ and parameters θ.

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

pθ (a1:T): probability distributions of the action sequence A

R(S): R_div + R_rep

Πθ: defined by our DSN

Training with Policy Gradient

Calculating the derivative of the expected reward function (Williams 1992):

$$\nabla_\theta J(\theta) = \mathbb{E}_{p_\theta(a_{1:T})}\left[R(S)\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta\left(a_t \mid h_t\right)\right]$$

at: the action

ht: the hidden state at time t

Training with Policy Gradient

Computing average of samples:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} R_n \nabla_\theta \log \pi_\theta \left(a_t \mid h_t\right)$$

Rn: reward got at the n-th episode

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

b: the moving average of rewards from the beginning to the present time

Regularization

Minimizing following expression:

$$L_{\text{percentage}} = \left\| \frac{1}{T} \sum_{t=1}^{T} p_t - \epsilon \right\|^2$$

ϵ: the percentage of frames is selected

Adding l2 regularization to avoid overfitting:

$$L_{\text{weight}} = \sum_{i,j} \theta_{i,j}^2$$

Optimization

Optimizing θ via stochastic gradient:

$$\theta = \theta - \alpha \nabla_{\theta} \left(-J + \beta_1 L_{\text{percentage}} + \beta_2 L_{\text{weight}} \right)$$

α: learning rate

B: hyperparameter balances the weight

Extension to Supervised Learning

Given the keyframes of video:

$$\mathcal{Y}^* = \{y_i^* \mid i = \ 1, \ldots, |\mathcal{Y}^*|\}$$

Maximizing the log-probability of selecting keyframes

$$L_{\text{MLE}} = \sum_{t \in \mathcal{Y}^*} \log p(t; \theta)$$

$$p_t = \sigma\left(W h_t\right),$$

Experiments

Setting up experiment

Dataset: TVsum, SumMe

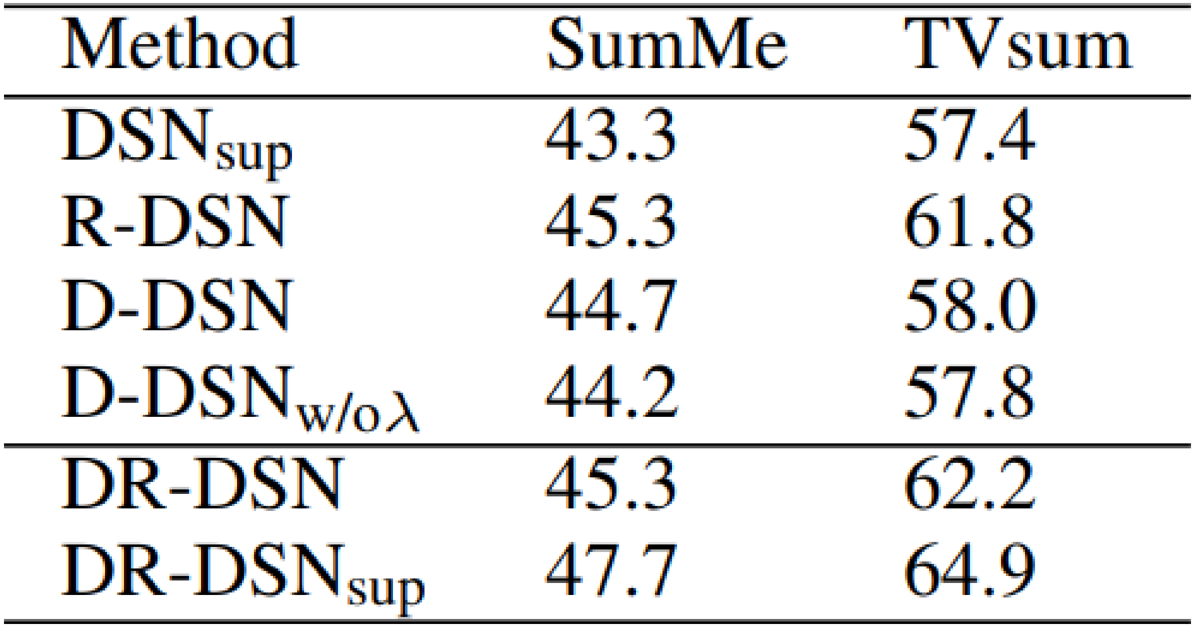
Evaluation metric: Fscore

Implementation Details:

- Downsample videos by 2 fps
- Temporal distance: 20
- Numbers of episodes to: 5
- Dimension of hidden state within the RNN cell: 256
- Epoch: 60

Quantitative Evaluation

Comparison with baseline:

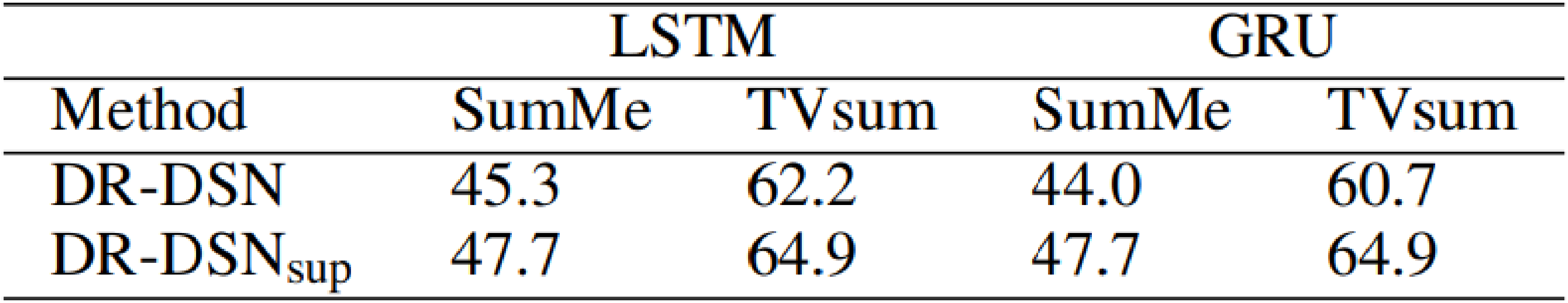


Method	SumMe	TVsum
DSN$_{\text{sup}}$	43.3	57.4
R-DSN	45.3	61.8
D-DSN	44.7	58.0
D-DSN$_{\text{w/o}\lambda}$	44.2	57.8
DR-DSN	45.3	62.2
DR-DSN$_{\text{sup}}$	47.7	64.9

Results (%) of different variants of our method on SumMe and TVSum

Quantitative Evaluation

Experiment with gated RNN units:

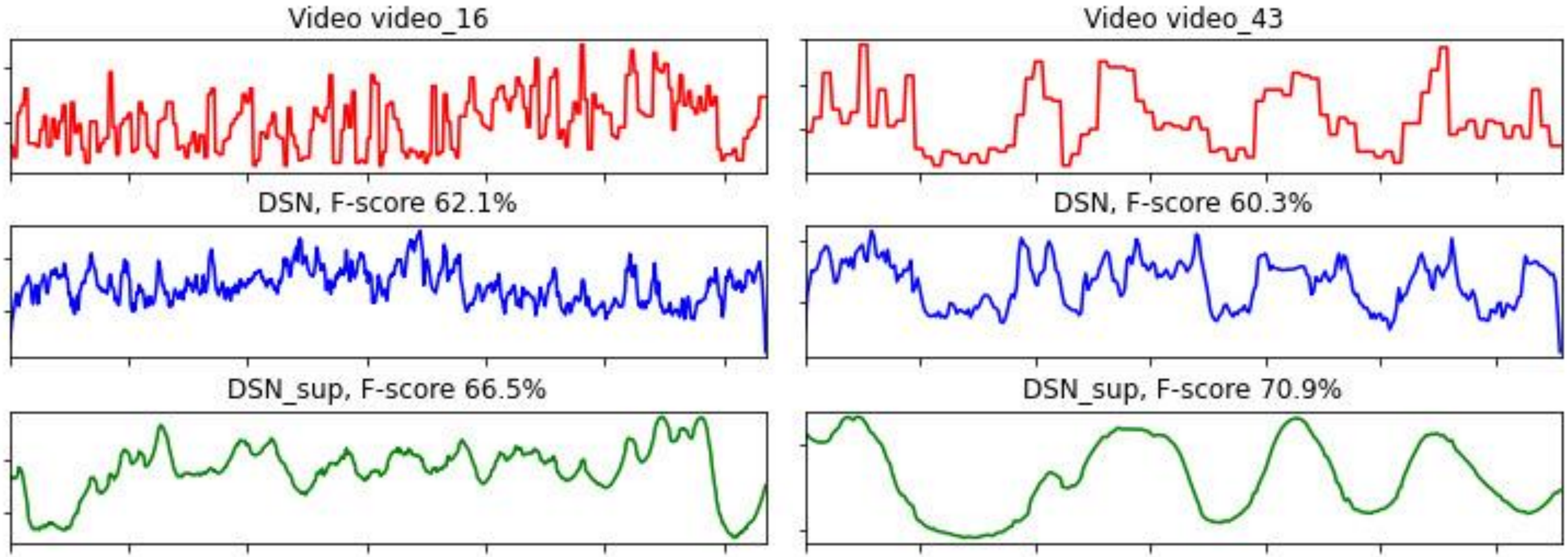


	LSTM		GRU	
Method	SumMe	TVsum	SumMe	TVsum
DR-DSN	45.3	62.2	44.0	60.7
DR-DSN$_{sup}$	47.7	64.9	47.7	64.9

Results (%) of using different gated recurrent units

Quantitative Evaluation

Predicted importance scores



Ground truth (top) and importance scores predicted by DR-DSN (middle) and DSNsup (bottom) of 2 video of TVsum dataset

Conclusion

After had an analyzation at every video summarization method and looked at their pros and cons, even though the F-score is around 60 points, we believe this model can be improved in the future.

By the time this report had been released, this model had been proven to be better than any other compared model.

Thank you for your attention

References

- 1801.00054v3.pdf (arxiv.org)
- https://medium.com/@sushmaparate28/supervised-video-summarization-using-deep-learning-39d023717ebf