Video Summarization with DSN and Reinforcement Learning

1st Dat Nguyen Duy, 2st Minh Le Doan Phuc, 3st Viet Le The
VNUHCM - University of Information and Technology
Ho Chi Minh City, Vietnam
20520435@gm.uit.edu.vn, 20520243@gm.uit.edu.vn, 20520093@gm.uit.edu.vn

***Abstract*—Video summarization aims to facilitate large-scale video browsing by creating short, concise summaries that are diverse and representative of original videos. A summarization needs diversity and representativeness. This paper introduces the end-to-end deep summarization network (DSN) framework that have been developed based on reinforcement learning by adding a reward function to the model. DSN predicts each video frame a probability, which indicates how likely a frame is selected, then takes actions based on the probability distributions to select video frames for forming video summaries. Because labels are not required, our method can be fully unsupervised. Extensive experiments on two standard data sets show that our unsupervised method not only outperforms other state-of-the-art unsupervised methods, but also comparable to or even superior to most of published supervised approaches. And finally, we will introduce about how to set up the model and using some specific parameters to evaluate the result.**

I. Introduction

Because of the exponential growth of the number of online videos in recent years, research in video summarization has gained increasing attention, leading to various methods proposed to facilitate large-scale video browsing conveniently.

Recently, recurrent neural network (RNN), especially with the long short-term memory (LSTM) has been exploited to model the sequential patterns in video frames, as well as to tackle the end-to-end training problem. Zhang et al. proposed a deep architecture that combines a bidirectional LSTM network with a Determinantal Point Process (DPP) module that enhances diversity in summaries, referring to as DPP-LSTM. By using both video-level summaries and frame level importance scores, they were able to train DPP-LSTM with supervised learning. At test time, DPP-LSTM predicts both importance scores and outputs feature vectors, which are important factors for creating a DPP matrix. Due to the nature of DPP model, DPP-LSTM needs to be trained in two stages. Although DPP-LSTM has given good efficiency in some cases, supervised learning cannot completely explore the potential of deep networks with video summarization. This is because human's subjective opinions should be selected as the summary. Because of that, it is important and needed to select a better method that rely less on labels.

Mahasseni et al. (Mahasseni, Lam, and Todorovic 2017) developed a framework to train DPP-LSTM. DPP-LSTM selects keyframes and a discriminator network is used to decide whether a synthetic video constructed by the keyframes is real or not in the learning process. Because of this, it enforces DPP-LSTM to select more frames that are representative. Although this is an unsupervised framework, the training session is unstable due to the adversarial essence that causing this problem, which leads to failure. In the matter of increasing diversity, unless there are labels, DPP-LSTM cannot maximize from the DPP module otherwise. Because of an RNN-based encoder-decoder network following DPP-LSTM for video reconstruction, their framework requires multiple training stages, making it not efficient in practice.

In this paper, we construct video summarization as a sequential decision-making process and develop a deep summarization network (DSN) for video summarization job. DSN has an encoder-decoder architecture, where the encoder is a convolutional neural network (CNN) that extracts features on video frames and the decoder calculates probabilities based on which actions are sampled for selecting frames using bidirectional LSTM network. For our DSN training, we propose an end-to-end, reinforcement learning-based framework with a diversity-representativeness (DR) reward function and calculate generated summaries based on diversity and representativeness without depending on labels or user interactions.

The idea for the DR reward function is based on the properties of a high-quality video summary should have. The 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, which is basically the k-medoids problem. These two rewards lead to each other and work to create diverse and representative summaries. This is the first document that we apply reinforcement learning to unsupervised learning for video summarization in our research.

The goal of DSN is to maximize the expected rewards over time. There are two reasons why we use reinforcement learning (RL) to train our DSN. First, we use RNN as part of model and focus on the unsupervised setting. At each temporal step, receiving supervision signal is needed for RNN. But due to rewards are computed over the entire video sequence, they only can be collected after a sequence

[1] Bidirectional Long Short-Term Memory Network with a Determinantal Point Process

[2] Determinantal Point Process

[3] Recurrent Neural Network

finishes. Because of providing supervision from a reward that is only available in the end of video sequence, RL becomes an inevitable choice. Secondly, we predict that DSN can benefit more from RL. Thanks to iteratively enforcing the agent to take better and better actions, RL can optimize the action (frame-selection) mechanism of an agent. However, in a normal supervised/unsupervised learning, optimizing action mechanism is not particularly highlighted.

Thanks to the labelless training process, our method can be fully unsupervised. In the case labels are available, we extend unsupervised method to the supervised version by adding a supervised objective that directly maximizes the log-probability of selecting annotated keyframes. By learning the high-level concepts which have been encoded in labels, our DSN can detect globally important frames and export summaries that is close to human-annotated summaries.

In this research, we tested on two datasets, SumMe (Gygli et al. 2014) and TVSum (Song et al. 2015), to evaluate the quantity and quality of our method. The quantitative results show that our unsupervised method not only outperforms other state-of-the-art unsupervised alternatives but also comparable or even superior to most of published supervised methods. Impressively, the qualitative results illustrate that DSN trained with our unsupervised learning algorithm can detect important frames that is close to human selections.

II. Related Works

Video summarization. In recent years, research in video summarization has advanced significantly, leading to approaches of various characteristics. Lee et al. (Lee, Ghosh, and Grauman 2012) dentified people and important objects in video sumarization. Gygli et al. (Gygli et al. 2014) predicts the degree of interestingness of video frames and selected keyframes with the highest interestingness scores by learning a linear regressor for it. Gygli et al. (Gygli, Grabner, and Van Gool 2015) cast video summarization as a subset selection problem and optimized submodular functions with multiple objectives. Ejaz et al. (Ejaz, Mehmood, and Baik 2013) extracted keyframes of visual saliency by applying an attention-modeling technique. Zhang et al. (Zhang et al. 2016a) developed a nonparametric approach to transfer structures of known video summaries to new videos with similar topics. Auxiliary resources have also been exploited for aiding the process of summarization such as web images/videos (Song et al. 2015; Khosla et al. 2013; Chu, Song, and Jaimes 2015) and category information (Potapov et al. 2014). Most of these non-deep summarization methods independently process video frames. Because of that, they are all ignoring the inherent sequential patterns. Not to mention, non-deep summarization methods usually do not support end-to-end training, which causing longer test time. To address the aforementioned issues, we model video summarization via a deep RNN and propose a reinforcement learning-based framework. The deep RNN is used for capturing longterm dependencies in video frames and the RL-based framework for end-to-end network training end to end.

Reinforcement learning (RL). Thanks to its effectiveness in various tasks, RL has become an increasingly popular research area. Mnih et al. (Mnih et al. 2013) successfully approximated Q function with a deep CNN, allowing their agent to beat a human expert in some Atari games. Later, many researchers have applied RL algorithms to vision-related applications, image captioning (Xu et al. 2015) and person re-identification (Lan et al. 2017) are some of the examples. In the area of video summarization, our work is not the first to use RL. In the past, Song et al. (Song et al. 2016) has implemented RL to train a summarization network for category-specific keyframe selection. Keyframe-labels and category information of training videos are required for their learning framework. However, our work is significantly different when compared to the work of Song et al. and other RL-based work in the way that during the learning process, it does not require labels or user interactions. This is also attributed to our novel reward function. Therefore, our summarization method can be fully unsupervised. And this method is also more practical for largescale video summarization.

III. Proposed Approach (Methods)

We define video summarization as a sequential decision-making process. In particular, DSN will predict a probability for each video frames and make decisions on which frames to select based on the predicted probability distributions. This method is an end-to-end, reinforcement learning-based framework for training our DSN. We create a reward function that include 2 main components: reward for diversity and reward for representativeness. Sum of them will evaluate directly reward of framework when it's selected. Figure 1 illustrates the overall learning process.

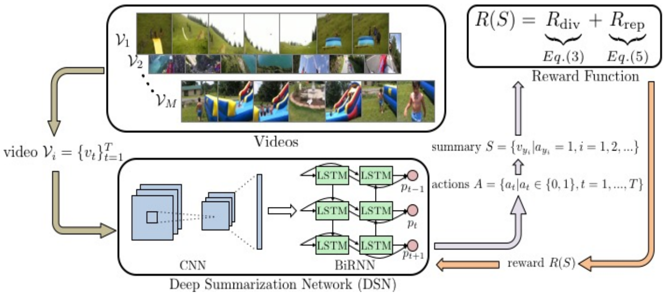
$R(S) = \underbrace{R_{\text{div}}}_{Eq.(3)} + \underbrace{R_{\text{rep}}}_{Eq.(5)}$
Reward Function
$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_M$
Videos
video $\mathcal{V}_i = \{v_t\}_{t=1}^T$
summary $S = \{v_{y_i} | a_{y_i} = 1, i = 1, 2, ...\}$
actions $A = \{a_t | a_t \in \{0, 1\}, t = 1, ...,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

Fig. 1. Deep summarization network (DSN) will be trained via reinforcement learning. DSN receives a video V_i and takes actions A (i.e., a sequence of binary variables) on which parts of the video are selected as the summary S. The feedback reward R(S) is computed based on the quality of the summary, i.e., diversity and representativenes

A. Deep Summarization Network (DSN)

DSN is formulated by the encoder-decoder framework. The encoder is a convolutional neural network (CNN) that extracts visual features from the input video frames. Specifically, CNN extracts the input video into a set of feature vectors $\{v_t\}_{t=1}^T$

with the length T. The decoder is a bidirectional recurrent neural network (BiRNN) topped with a fully connected (FC) layer. Using BiRNN as a decoder is the difference between this model and other models. We are particularly interested in the relationship between current frames and past and future frames. Therefore, BiRNN is chosen over RNN (which does not preserve frame-to-frame relevance in the long run) or one-way LSTM (which only cares about past frame relevance). The BiRNN takes as input the set of feature vectors $\{x_t\}_{t=1}^T$ and computes corresponding hidden states $\{h_t\}_{t=1}^T$. Each h_t is the concatenation of the forward hidden state h_t^f and the backward hidden state h_t^b (which is the combination of traversing from the beginning of the video onwards to the t frame and from the end of the video back to the next frame). The importance of frames close to frame t is emphasized. The FC layer that ends with the sigmoid function predicts a probability p_t for each frame, from which we sample with action a_t if t^{th} frame is selected. Sigmoid function guarantee that every p_t has a value in the range of [0;1]:

$$p_t = \sigma\left(W h_t\right), \tag{1}$$

$$a_t \sim \text{Bernoulli}\left(p_t\right), \tag{2}$$

where σ represents the sigmoid function, $a_t \in \{0, 1\}$ indicates whether the t^{th} frame is selected or not. W is the weight of the state hide h_t, the bias in Eq.(1) is ignored to make the expression more readable formula. The video after being summarized is a set of selected frame, $S = \{v_{y_i} \mid a_{y_i} = 1, i = 1, 2, \ldots\}$.

In practice, we use the GoogLeNet (Szegedy et al. 2015) for data preprocessing on ImageNet (Deng et al. 2009) as the CNN model to extract output features in original video. Real training network includes only BiRNN that receive input as features that have been extracted from CNN. The visual feature vectors $\{x_t\}_{t=1}^T$ are extracted from the penultimate layer of the GoogLeNet. During training, we only update the decoder (actually here is the 2-way LSTM network).

B. Diversity-Representativeness Reward Function

During training, DSN will receive a reward R(S) that evaluates the quality of the summaries created. The goal of DSN is to maximize the expected rewards throughout the process of generating more high-quality summaries. In general, a high-quality video summary includes diverse and representative of the original video. Therefore, the information of the whole video will be kept to the maximum extent. Finally, we set a new reward that evaluates the degree of diversity and representativeness. The proposed reward is the combination of two rewards R_{div} and R_{rep}, which we detail as follows.

Diversity reward: Measured by the difference between selected frames in the future space. Let the selected frames be $\gamma = \{v_{y_i} | a_{y_i} = 1, i = 1, ..., |\gamma|\}$. R_{div} is the mean of the pairwise dissimilarities among the selected frames:

$$R_{\text{div}} = \frac{1}{|\mathcal{Y}|(|\mathcal{Y}| - 1)} \sum_{t \in \mathcal{Y}} \sum_{\substack{t' \in \mathcal{Y} \\ t' \neq t}} d\left(x_t, x_{t'}\right), \tag{3}$$

where $d(\cdot, \cdot)$ is the dissimilarity function calculated by

$$d\left(x_t, x_{t'}\right) = 1 - \frac{x_t^T x_{t'}}{\left\|x_t\right\|_2 \left\|x_{t'}\right\|_2}. \tag{4}$$

In this problem, we set $d(x_t, x_{t'}) = 1$ if $|t - t'| > \lambda$, where λ controls the degree of temporal distance.

Representativeness reward: This reward measures how well the original video's content is represented. Defined as the K-medoids problem. The agent selects a set of medoids such that the mean of squared errors between video frames and their nearest medoids is minimal. Therefore, R_{rep} is defined:

$$R_{\text{rep}} = \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \min_{t' \in \mathcal{Y}} \left\|x_t - x_{t'}\right\|_2\right). \tag{5}$$

With this reward, the agent tends to choose frames that are close to the cluster centers in the future.

Diversity-representativeness reward: R_{div} and R_{rep} complement together to form the learning guide of DSN:

$$R(S) = R_{\text{div}} + R_{\text{rep}}\,. \tag{6}$$

During training, R_{div} and R_{rep} are similar in terms of importance level. In fact, it is not important to keep R_{div} and R_{rep} trivial. So, neither one prevails in the gradient calculation. When no frame is selected, the reward will be zero, that is, all action samples are 0.

C. Training with Policy Gradient

We want to maximize the expected rewards when summarizing videos through a policy function π_θ with parameters θ.

$$J(\theta) = \mathbb{E}_{p_\theta(a_{1:T})}[R(S)] \tag{7}$$

where $p_\theta\left(a_{1:T}\right)$ are the probability distributions of the action sequence A, and $R(S)$ is computed by R_{div} + R_{rep}. π_θ is defined by our DSN.

We calculate the derivative of the expected reward function ($J(\theta)$ by θ using Williams' formula (Williams 1992):

$$\nabla_\theta J(\theta) = \mathbb{E}_{p_\theta(a_{1:T})}\left[R(S) \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta\left(a_t \mid h_t\right)\right] \tag{8}$$

where a_t is the action and h_t is the hidden state from the BiRNN at time t.

The above formula will take a lot of resources to calculate the expectation, so we offer a more viable way that taking the sample with a certain number of times and then computing average.

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} R_n \nabla_\theta \log \pi_\theta\left(a_t \mid h_t\right) \tag{9}$$

where R_n is the reward got at the n^{th} episode.

The disadvantage of this formula is that it is difficult to converge because of large variance. So we subtract reward for one base b in the gradient calculation:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T} (R_n - b) \nabla_\theta \log \pi_\theta (a_t \mid h_t) \quad (10)$$

where b is computed as the moving average of rewards from the beginning to the present time.

D. Regularization

One thing that is easy to realize is that if we want high rewards, we just need to choose many frames as possible and the rewards are maximized when all the entire frame is selected. But this is not the approach that we aim for. We want to summarize a video into a shorter video, about 20-50 percent of length from original video duration. Inspired by (Mahasseni, Lam, and Todorovic 2017), we minimize the following expression during training,

$$L_{\text{percentage}} = \left\| \frac{1}{T} \sum_{t=1}^{T} p_t - \epsilon \right\|^2, \quad (11)$$

where ϵ determines the percentage of frames to be selected.

In addition, we also add the $\ell 2$ regularization term on the weight parameters θ to avoid overfitting

$$L_{\text{weight}} = \sum_{i,j} \theta_{i,j}^2. \quad (12)$$

E. Optimization

We optimize the policy function's parameters θ via stochastic gradient. By combing the gradients computed from Eq.(10), Eq.(11) and Eq.(12), θ is updated:

$$\theta = \theta - \alpha \nabla_\theta \left(-J + \beta_1 L_{\text{percentage}} + \beta_2 L_{\text{weight}} \right), \quad (13)$$

where α is learning rate, and β is hyperparameter to balance the weight.

In practice, we use Adam (Kingma and Ba 2014) as the optimization algorithm. The log-probability of actions is taken by the network that help high reward action to increase, while low rewards action is decreased.

F. Extension to Supervised Learning

Given the keyframe of a video, $\mathcal{Y}^* = \{y_i^* \mid i = 1, \ldots, |\mathcal{Y}^*|\}$, we use Maximum Likelihood Estimation (MLE) to maximize the log-probability of selecting keyframes determined by $\mathcal{Y}^*, \log p(t;\theta)$ where $t \in \mathcal{Y}^* \cdot p(t;\theta)$ is computed from Eq. (1). The formula is defined as:

$$L_{\text{MLE}} = \sum_{t \in \mathcal{Y}^*} \log p(t;\theta). \quad (14)$$

G. Summary Generation

For a test video, we apply a trained DSN to predict the frame-selection probabilities as importance scores. We calculate the score of a scene by averaging the score of frames in the scene. For temporal segmentation, we use KTS proposed by (Potapov et al. 2014). To generate a summarization, we select the scene by maximizing the total scores while we ensure summarization's length that not exceed the 15 percent of video length.Maximization step is actually the 0/1 Knapsack problem, which is known as NP-hard. We get the results close to the optimal results through dynamic programming (Song et al.2015).

In addition to evaluating generated summaries in the Experiments part, we also do statistics with the original results of the DSN that exclude the influence of the generated summarization step. Through which we can better understand than DSN learn what.

IV. Experiment (Results)

A. Setting up experiment

Datasets. We evaluate our methods on SumMe (Gygli etal. 2014) and TVSum (Song et al. 2015). SumMe consists of 25 user videos covering various topics such as holidays and sports. Each video in SumMe ranges from 1to 6 minutes and is annotated by 15 to 18 persons, thusthere are multiple ground truth summaries for each video.TVSum contains 50 videos, which include the topics ofnews, documentaries, etc. The duration of each video variesfrom 2 to 10 minutes. Similar to SumMe, each video inTVSum has 20 annotators that provide frame-level importance scores. Following (Song et al. 2015; Zhang et al.2016b), we convert importance scores to shot-based summaries for evaluation.

Evaluation metric. For fair comparison with other approaches, we follow the commonly used protocol from(Zhang et al. 2016b) to compute F-score as the metric to assess the similarity between automatic summaries and groundtruth summaries. We also follow (Zhang et al. 2016b) to dealwith multiple ground truth summaries.

Implementation Details. Instead of using the source code in the author's article, we modified that source code based on the sample source for testing. However, the parameter settings remain unchanged. We also downsample videos by 2 fps and set the temporal distance to 20, still the length ratio $\in$ is 0.5. The quantity of episodes N to 5, as well the other hyperparameters, with α, $\beta 1$ and $\beta 2$ are optimized via cross-validation. We set the dimension of hidden state within the RNN cell to 256 throughout this paper.

B. Quantitative Evaluation

Comparison with baseline. We first start to set the main baseline model trained only with a reward, Rdiv and Rrep respectively, with the former is denoted by D-DSN while the latter is R-DSN. After calculating each reward separately, we combine them into a reward function and let the baseline model trained with this function as it used to, which is named DR-DSN. The model also extends to the supervised version called DR-DSNsup. Recall a lambda function above, the author validates the effectiveness of the -technique ignoring the distant similarity when computing Rdiv. We continue to take the D-DSN model trained without -technique, with the symbol as D-DSN$_{\text{w/o}\lambda}$. To demonstrate that DSN can get more benefit from reinforcement learning than from supervised learning, the DSN_{sup} model is generated by training with cross entropy loss using key frame annotation, where a regularization term is interpreted as a confidence penalty for a distributed output.

Table 1: Results (%) of different variants of our method on SumMe and TVSum

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

Table 1 reports the results of different variants of our method on SumMe and TVSum. We can see that DR-DSN clearly outperforms D-DSN and R-DSN on both datasets, which demonstrates that by using Rdiv and Rrep collaboratively, we can better teach DSN to produce high-quality summaries that are diverse and representative. Comparing the unsupervised model with the supervised one, we see that DR-DSN significantly outperforms DSNsup on the two datasets (45.3 vs. 43.3 on SumMe and 62.2 vs. 57.4 on TVSum), which justifies our assumption that DSN can benefit more from reinforcement learning than from supervised learning.

By adding the supervision signals of LMLE (Eq. (14)) to DR-DSN, the summarization performances are further improved (2.4% improvements on SumMe and 2.7% improvements on TVSum). This is because labels encode the highlevel understanding of the video content, which is exploited by DR-DSN$_{\text{sup}}$ to learn more useful patterns. The performances of R-DSN are slightly better than those of D-DSN on the two datasets, which is because diverse summaries usually contain redundant information that are irrelevant to the video subject. We observe that the performances of D-DSN are better than those of D-DSN$_{\text{w/o}\lambda}$ that does not consider temporally distant frames. When using the -technique in training, around 50% $\sim$ 70% of the distance matrix was set to 1 (varying across different videos) at the early stage. As the training epochs increased, the percentage went up too, eventually staying around 80% $\sim$ 90%. This makes sense because selecting temporally distant frames can lead to higher rewards and DSN is encouraged to do so with the diversity reward function.

Comparison with the author's result. As can be seen from the tables, it is obvious from our chart that TVSum F-score shows a significant score in all type of DSN experiment, while the figure for SumMe shows an opposite trend. Comparing our data with the original result in both type (SumMe and TVSum) using DSN model to train, there is a big difference in both SumMe results and TVsum result, our data is higher than author data in 6 different DSN methods, although there is no considerable differences in the result. Because the model train uses random function to choose random epoch, our results is still objective.

Table 2: The author's result

Method	SumMe	TVsum
DSN$_{\text{sup}}$	38.2	54.5
R-DSN	40.7	56.9
D-DSN	40.5	56.2
D-DSN$_{\text{w/o}\lambda}$	39.3	55.7
DR-DSN	41.4	57.6
DR-DSN$_{\text{sup}}$	42.1	58.1

Table 3: Results (%) of using different gated recurrent units and TVSum

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

Experiment with different gated RNN units. We experiment withd LSTM and GRU (Cho et al. 2014), and find that LSTM-based models consistently beat GRU-based models (see Table 3). This may be interpreted as that the memory mechanism in LSTM has a higher degree of complexity, thus allowing more complex patterns to be learned.

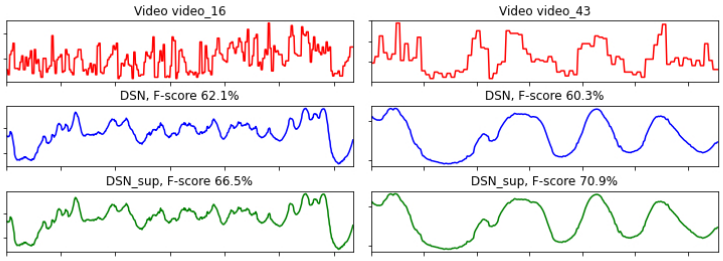

Fig. 2. Ground truth (top) and importance scores predicted by DR-DSN (middle) and DSNsup (bottom) of 2 video of TVsum dataset.

Predicted importance scores. We visualize the raw predictions by DR-DSN and DSNsup in Fig. 3. By comparing predictions with ground truth, we can better understand in more depth how well DSN has learned. It is worth highlighting that the curves of importance scores predicted by the unsupervised model resemble those predicted by the supervised model in several parts. More importantly, these parts coincide with the ones also considered as important by humans. This strongly demonstrates that reinforcement learning with our diversity-representativeness reward function can well imitate the human-learning process and effectively teach DSN to recognize important frames.

V. Conclusion

In this report, we have been able to build a video summarization model using Deep Summarization Network (DSN) and Reinforcement Learning (RL) with multi-respersentative rewards function, which is inspired by Kaiyang Zhou (Kaiyang Zhou al et. 2017). After we 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. At least by the time this report had been released, this model had been proven to be better than any other compared model.

VI. Acknowledgment

We would like to express our special thanks of gratitude to our teacher Dr. Ngoc Hoang Luong who gave us the golden opportunity to do this wonderful project on the topic Video Summarization with DSN and Reinforcement Learning. We also want to send our appreciation to Mr. Sylvain Marchienne who has modified the DSN code.

References

G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

K. Elissa, “Title of paper if known,” unpublished.

R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.