Video Summarisation by Classification with Deep Reinforcement Learning

Kaiyang Zhou
k.zhou@qmul.ac.uk
Tao Xiang
t.xiang@qmul.ac.uk
Andrea Cavallaro
a.cavallaro@qmul.ac.uk

Computer Vision Group and
Centre for Intelligent Sensing,
School of Electronic Engineering and
Computer Science,
Queen Mary University of London,
London E1 4NS, UK

Abstract

Most existing video summarisation methods are based on either supervised or unsupervised learning. In this paper, we propose a reinforcement learning-based weakly supervised method that exploits easy-to-obtain, video-level category labels and encourages summaries to contain category-related information and maintain category recognisability. Specifically, We formulate video summarisation as a sequential decision-making process and train a summarisation network with deep Q-learning (DQSN). A companion classification network is also trained to provide rewards for training the DQSN. With the classification network, we develop a global recognisability reward based on the classification result. Critically, a novel *dense* ranking-based reward is also proposed in order to cope with the temporally delayed and sparse reward problems for long sequence reinforcement learning. Extensive experiments on two benchmark datasets show that the proposed approach achieves state-of-the-art performance.

1 Introduction

Video summarisation has traditionally been formulated as an unsupervised learning problem [2, 19, 33, 34, 39, 43, 44, 47, 48], with criteria to identify keyframes (or key-segments) handcrafted based on generic rules, such as diversity and representativeness. However, different types of video content may require different criteria or different combinations of them: for instance, summaries of *Eiffel Tower* videos should contain scenes with the tower, whereas summaries of *Making Sandwich* videos should focus on the key temporal stages of the task. How humans deploy these criteria based on the video content can be reflected through their summary annotations, which indicate whether each video frame or segment should be included in the summary. With the annotations, a supervised video summarisation model can be developed [5, 6, 7, 37, 41, 45], capturing implicitly the content-specific frame/segment selection criteria. However, its use for large-scale summarisation tasks is limited because summary annotations are expensive to collect and prone to bias due to the subjective nature of video summaries.

In this paper, a novel weakly-supervised video summarisation approach is proposed, which is content-specific but only requires video-level annotations in the form of video cat-

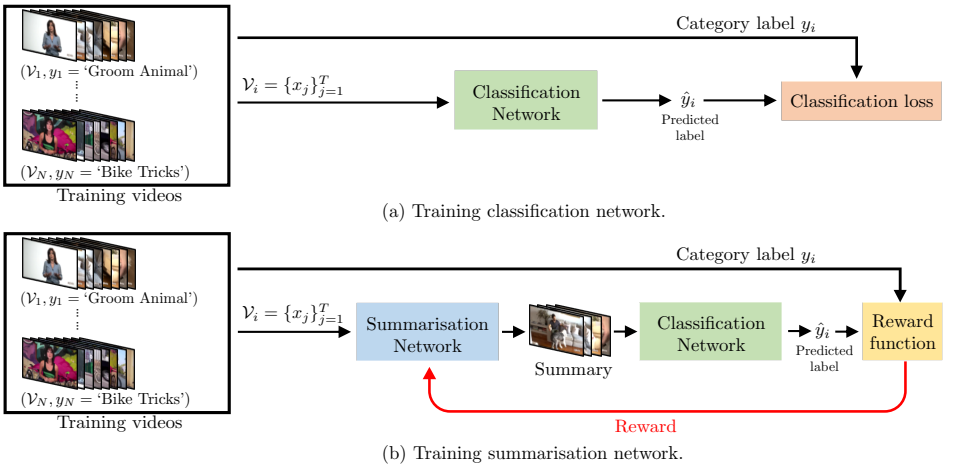


Figure 1: Framework overview. After training a classification network with category labels and freezing its weights (a), we train a summarisation network with deep Q-learning with the goal of generating summaries by removing redundant frames, while ensuring informative parts are maintained and the classification network can recognise them (b).

egory labels. These video-level labels are easy to obtain, making the approach much more scalable than the supervised alternatives. Our approach is motivated by the fact that category labels typically encapsulate strong semantic information about the video content. Maintaining the recognisability of the video after removing frames/segments to produce a summary can thus be considered as a top-level selection criterion. Such a criterion encapsulates various fine-grained, content-specific criteria deployed by humans. For example, to summarise videos labelled as *Groom Animal*, humans would select segments containing one or more people who are working on an animal to support the semantic meaning conveyed by the category label. We therefore propose to learn a video summarisation model that selects video frames/segments based on whether collectively they contribute the most to recognising the summarised video into its category label.

More specifically, we propose to utilise the video classification criterion elaborated above to guide the learning of a deep video summarisation network. We train the video summarisation network using reinforcement learning (RL) due to the following reasons. First, the classification of summaries can only be made at the end of videos whilst a decision/action needs to be made at every single frame on whether it should be included in the summary. This problem is thus naturally suited for RL. Second, the frame selections are inter-dependent in that the selection of one frame would have implications on the selection of others. The exploration-exploitation strategy of RL can better guide the summarisation network to capture the interdependencies among frames as different combinations of frames are explored.

Figure 1 shows the proposed framework. In order to provide rewards during the reinforcement learning of the summarisation network so that the video category recognisability is maintained, our framework includes a companion classification network. This network is a recurrent network learned with supervised classification loss. This classification network can judge whether a given video sequence contains sufficient information to be classified to a certain category. This judgement is then used as a supervision signal/reward to guide the learning of the summarisation network. Concretely, we formulate the judgement made by

the classifier as global recognisability reward and train the summarisation network with deep Q-learning [17, 32]. The summarisation network is thus termed as Deep Q-learning Summarisation Network (DQSN). Given a video, DQSN generates a summary by sequentially removing frames based on the prediction on future rewards. The classifier then classifies the summary and returns the global recognisability reward to DQSN, which is explicitly encouraged to produce summaries containing category-related information.

A well-known challenge in RL is the credit assignment problem, *i.e*. rewards are sparse or temporally delayed thus making it difficult to associate each action with a reward. With only the global recognisability reward, our DQSN also suffers from this problem as the single global reward can only be generated after a complete sequence of actions, which inevitably slows down the model convergence. The problem is particularly acute in our case due to the length of the sequences we are dealing with. We mitigate this problem by proposing a novel *dense* reward, which we call local relative importance reward. This reward gives each action a feedback by checking if the action changes the recognisability of the partial summary generated so far. Importantly, this reward is also obtained by the classification network, without requiring additional modules.

Contributions. (1) For the first time, a RL-based weakly supervised video summarisation framework is proposed, which requires only video-level category labels. (2) We overcome the notorious credit assignment problem in RL by introducing a novel dense reward. (3) To show the flexibility of our framework, we combine our weakly supervised rewards with those deployed in existing unsupervised approaches and demonstrate their complementarity. (4) We show that, on two widely-used benchmark datasets, namely TVSum [28] and CoSum [2], our approach not only outperforms unsupervised/weakly supervised alternatives but is also highly competitive against supervised approaches.

2 Related Work

Existing video summarisation approaches can be categorised as unsupervised, supervised or weakly supervised. Conventional unsupervised approaches cluster frames [19, 33, 34, 47, 48] or optimise hand-crafted objectives [2, 11, 12, 20, 28] to identify keyframes or key-segments. The selection criteria are usually generic (*e.g*. diversity [38] and representativeness [4, 39, 44]) and do not encode semantics. In contrast, supervised approaches aim to exploit semantics embedded in manually annotated summaries [5, 6, 27, 37, 41, 45]. In [14], keyframe labels are used to teach neural networks to skip unimportant frames. Since summary annotations are likely to contain biases and are expensive to collect, weak labels such as video category have been exploited to learn useful concepts to aid summarisation [21, 25].

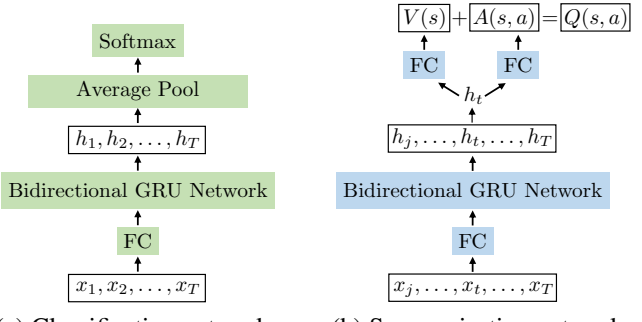
Since our approach is based on weakly-supervised learning of deep network, the most related video summarisation work is [21]. In [21], a 3D ConvNet is trained to predict categories for video clips. Important clips are identified by summing up back-propagated gradients from the true category probability. We significantly improve upon [21] by exploiting category labels with a RL formulation where the interdependencies between frames can be better explored. Our work is also related to [46] in that it is also based on RL. [46] proposes an unsupervised reward function to train a frame-selection network with policy gradient. Our method differs from [46] in that our global reward function is based on the recognisability of video summaries, while the reward in [46] encourages diversity and representativeness, which are in many cases too generic, as discussed above. Moreover, we use a local dense

reward that can compensate the temporally delayed global reward. Our experiments (see Sec. 4) show that our approach clearly outperforms those in [21, 46].

Beyond video summarisation, several computer vision problems such as image captioning [26, 35, 42], visual tracking [10, 29, 40] and sketch abstraction [18] have been formulated as decision-making processes using RL. A major challenge in RL is credit assignment, which causes difficulties to associating each action with a global sequence-level reward which is sparse and temporally delayed. A common countermeasure is to devise so-called intrinsic rewards, such as the curiosity reward [24], for intermediate states. In our case, the summarisation agent can only receive the reinforcement signal when it finishes a (long) video sequence, thus the reward is severely delayed and sparse. To overcome this issue, we propose a novel dense reward to provide prompt feedback to intermediate states.

3 Proposed Approach

Our approach combines two bidirectional recurrent networks with gated recurrent unit (GRU) [1]: a classification network (Sec. 3.1) and a summarisation network (Sec. 3.2). Both networks take as input image features extracted by a pretrained ConvNet. We first train the classification network using supervised classification loss and video-level category labels. Then, we apply the fixed classification network to classify the summaries generated by the summarisation network. The classification result is formulated as a reward function and the summarisation network is trained using deep Q-learning [17, 32].



(a) Classification network. (b) Summarisation network.

Figure 2: Network architectures. x_t represents frame features. h_t represents hidden states.

3.1 Classification Network

Fig. 2(a) shows the design of our classification network. The input features are first mapped to an embedding space via a fully connected (FC) layer. We use PReLU [9] as the nonlinear activation function throughout this paper. The embedded features are then processed by a bidirectional GRU (Bi-GRU) network where the outputs from both temporal directions are concatenated, followed by an average pooling layer. Finally, a FC layer with softmax function is mounted on the top to predict C probabilities corresponding to C categories. We train this network using cross entropy loss equipped with the label smoothing regulariser to

reduce overfitting [31]. Thus, the loss for a video can be expressed as

$$\mathcal{L} = -\sum_{k=1}^{C} q(k)\log p(k), \quad s.t. \quad q(k) = (1-\omega)\delta(k=y) + \frac{\omega}{C}, \tag{1}$$

where q is label, p is prediction, $\delta(\text{condition})$ is 1 if the condition is true otherwise 0, and ω is weight, which is fixed to 0.1 as suggested in [31].

3.2 Deep Q-Learning Summarisation Network

We cast video summarisation as a sequential decision-making process and develop a summarisation network to approximate the action-value function. We term the summarisation network trained with the deep Q-learning algorithm [17] as DQSN (parameterised by θ). From the reinforcement learning perspective, our framework can be described by a Markov Decision Process (MDP), formally defined as a tuple $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R},\gamma)$. $\mathcal{S}$ is a set of states experienced by DQSN. A state s_t at time t is defined by a sequence of video frames. $\mathcal{A}$ is the action space composed of two actions: 1 for keeping frame and 0 for discarding frame. $\mathcal{P}(s_{t+1}|s_t,a_t)$ is the transition probability from the current state s_t to the next state s_{t+1} after taking an action $a_t \in \mathcal{A}$. $\mathcal{R}(r_t|s_t,a_t,s_{t+1})$ is the reward for transition (s_t,a_t,s_{t+1}). $\gamma \in [0,1]$ is the discount factor used to reduce the effect of future rewards.

The sequential summarisation process is described as follows. At the first time step $t=1$, the state s_1 is composed of the entire sequence of frames of a video, *i.e.* $s_1 = \{x_j | j = 1,2,...,T\}$, but with an attention on x_1. DQSN processes s_1 and predicts action values $Q_\theta(s_1,a_1)$ for x_1. If $Q_\theta(s_1,a_1=1) > Q_\theta(s_1,a_1=0)$, we keep x_1 and update next state $s_2 = s_1$. If $Q_\theta(s_1,a_1=1) < Q_\theta(s_1,a_1=0)$, we remove x_1 and update $s_2 = s_1 \backslash \{x_1\}$. While updating the next state, we simultaneously shift the attention to x_2. A reward r_1 will be given based on (s_1,a_1,s_2). Iteratively, DQSN processes s_t and predicts action values for frame x_t. This process terminates when $t=T$ or the number of remaining frames reaches a threshold.

The objective of DQSN is to take actions that maximise discounted future rewards $R_t = \sum_{t'=t}^{T} \gamma^{t'-1} r_{t'}$. According to the Bellman equation, DQSN can be trained with deep Q-learning [17] to regress R_t:

$$\mathcal{L}_Q = \mathbb{E}_{s_t,a_t,r_t,s_{t+1}}[(R_t - Q_\theta(s_t,a_t))^2], \quad s.t. \quad R_t = r_t + \gamma \max_{a_{t+1}} Q_{\theta^-}(s_{t+1},a_{t+1}), \tag{2}$$

where θ^- represents the parameters of a target network, which is identical to DQSN but is updated periodically. r_t is a hybrid reward, which is detailed in Sec. 3.2.1. In practice, applying a separate network to estimate the future rewards has been proven advantageous to stabilise the training [17].

Fig. 2(b) shows the network architecture of DQSN, whose bottom layers are identical to those of the classification network. The top layers of DQSN aim to predict action values $Q(s,a)$ for frame x_t. The bidirectional design allows the past and future information to be jointly captured. Inspired by [36], we design two streams to produce separate estimates of the value function $V(s)$ and the advantage function $A(s,a)$. $V(s)$ is a scalar that represents the quality of a state achieved by DQSN. $A(s,a)$ is composed of two scalars that provide relative measures of the importance for the two actions.

Separating $V(s)$ and $A(s,a)$ makes the learning of $Q(s,a)$ more efficient and more robust to numerical scale, as discussed in [36]. To overcome the lack of identifiability between $V(s)$

and $A(s,a)$ when learning $Q(s,a)$, we follow [36] and combine these two functions via

$$Q(s,a) = V(s) + \left(A(s,a) - \frac{1}{|\mathcal{A}|} \sum_{a'} A(s,a') \right) \quad s.t. \quad \mathcal{A} = \{0,1\}. \tag{3}$$

3.2.1 Reward Function

The reward term r_t in Eq. 2 is a combination of three rewards: global recognisability reward r_t^g, local relative importance reward r_t^l and unsupervised reward r_t^u.

Global Recognisability Reward. We propose to use the classification results of video summaries as a signal to guide the learning of DQSN. This reward is global because it is only available when a summarisation process finishes, *i.e.* $t = T$. Specifically, if a video summary can be recognised by the classification network, *i.e.* $\hat{y} = y$, we reward the summary with +1, otherwise we penalise the summary with -5[1]. Mathematically, this reward is formulated as

$$r_t^g = \delta(\hat{y} = y) - 5(1 - \delta(\hat{y} = y)) \quad s.t. \quad t = T. \tag{4}$$

Local Relative Importance Reward. To mitigate the credit assignment problem [24], we propose a novel local relative importance reward r_t^l, which evaluates the immediate result of removing a frame. The summarisation network can therefore obtain a prompt feedback on the quality of each action. For each transition (s_t, a_t, s_{t+1}), the classification network classifies s_t and s_{t+1}, resulting in ξ_t and ξ_{t+1}, which represent the rank of the true category. For example, if s_t is correctly recognised, $\xi_t = 1$. We introduce relative importance reward based on the change of rank caused by a_t. The intuition behind this reward is simple: if the rank is improved, we reward a_t; otherwise we penalise a_t. To encourage DQSN to remove as many (redundant) frames as possible, we further reward with +0.05 for intermediate states if $a_t = 0$. We formulate r_t^l as a function of hyperbolic tangent (tanh):

$$r_t^l = 0.05(1 - a_t) + h(\xi_t, \xi_{t+1}) \quad s.t. \quad h(\xi_t, \xi_{t+1}) = \tanh\left(\frac{\xi_t - \xi_{t+1}}{\eta} \right), \quad t < T, \tag{5}$$

where η is a scaling factor. $h(\xi_t, \xi_{t+1})$ measures the importance of $x_t | a_t = 0$ *relative* to previously removed frames. Note that this reward is only computed when $a_t = 0$ so it is computationally efficient.

Unsupervised Rewards. Similar to r_t^g, r_t^u is also computed globally. We employ the unsupervised diversity-representativeness (DR) reward proposed in [46],

$$r_t^u = \frac{1}{|\mathcal{Y}||\mathcal{Y}| - 1} \sum_{t \in |\mathcal{Y}|} \sum_{\substack{t' \in |\mathcal{Y}| \\ t' \neq t}} d(x_t, x_{t'}) + \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \min_{t' \in \mathcal{Y}} ||x_t - x_{t'}||_2\right), \tag{6}$$

where $\mathcal{Y} = \{y_i | a_{y_i} = 1, i = 1, ..., |\mathcal{Y}|\}$ contains indices of kept frames and $d(\cdot,\cdot)$ is cosine dissimilarity. The first term computes the dissimilarity among selected frames (or segments) while the second term evaluates how well the original video can be reconstructed by the summary. We show in Sec. 4 that the unsupervised DR reward is complementary to our weakly-supervised reward (r_t^g and r_t^l). Note that we give equal weights to r_t^g and r_t^u.

[1] We give stronger weight to penalty to encourage DQSN to produce summaries with high recognition accuracy, which was found effective in experiments.

3.2.2 Optimisation with Experience Replay and Double Q-Learning

We employ experience replay [15] for minibatch updates. At each time step, we store the experience as a tuple $e_t = (s_t, a_t, r_t, s_{t+1})$ into a replay memory $\mathcal{M}$ initialised with a fixed capacity. To update DQSN, we randomly sample minibatches of experiences from $\mathcal{M}$ with uniform distribution. We perform ε-greedy policy to select actions, *i.e.* we choose a random action with probability ε and an optimal action from DQSN with probability $1-\varepsilon$.

Q-learning is prone to overestimate action values [8], as the max operator uses the same function to select as well as to evaluate an action (Eq. 2). To alleviate this issue, we apply double Q-learning [8, 32], its improved version. Specifically, the current Q learner is employed to select the optimal action of the next state and this action is then evaluated using the target network, so Eq. 2 is substituted with

$$\mathcal{L}_Q = \mathbb{E}_{\{e_t\}\sim\mathcal{M}}[(R_t - Q_\theta(s_t,a_t))^2], \quad s.t. \quad R_t = r_t + \gamma Q_{\theta^-}(s_{t+1}, \arg\max_{a_{t+1}} Q_\theta(s_{t+1},a_{t+1})). \tag{7}$$

The gradients are computed based on Eq. 7, $\nabla_\theta \mathcal{L}_Q$. In practice, we replace the squared error loss with Huber loss, which is less sensitive to outliers. To optimise θ, we use Adam [13] and clip the norm of gradients at 5 to avoid exploding gradients [22].

Summary Generation. During testing, we select actions by $\arg\max_a Q(s,a)$. We score each frame with softmax normalised $Q(s,a=1)$. Shot-level scores are computed by averaging frame scores within the same shots and generate summaries by selecting shots with the highest scores but keeping the duration below a threshold, following [16, 41, 46]. Note that during testing, the video category labels are *not* required.

4 Experiments

We implement our model using PyTorch [23][2]. We use GoogLeNet [30] trained on ImageNet [3] to extract frame features followed by $\ell 2$ normalisation as the input to the classification and summarisation networks. The dimension of embedding space and hidden units of GRU is 256. The discount factor γ is 0.99; η in Eq. 5 is set to 0.15 via cross-validation; ε in the ε-greedy policy decreases exponentially from 1 and stops at 0.1. We set the capacity of $\mathcal{M}$ and minibatch of transitions to 6000 and 200, respectively. The learning rate is $1e-04$.

4.1 Datasets and Settings

Experiments are conducted on the widely used TVSum [28] and CoSum [2] datasets, which contain two sets of non-overlapping categories. TVSum contains 10 categories[3] each with 5 videos, whose length varies from 2 to 10 minutes. CoSum consists of 51 videos, whose length ranges from 1 to 12 minutes, covering 10 categories[4]. Both datasets were annotated by multiple persons so there are multiple human summaries for each video. For evaluation, we compute F-score for each pair of machine summary and human summary and average results for a single video. The overall results are obtained via 5-fold cross-validation, following [16, 41, 46]. We follow [2, 28] to obtain shot-based summaries for evaluation.

[2]Code and data will be released at https://github.com/KaiyangZhou.

[3]TVSum categories: Changing Vehicle Tire, Getting Vehicle Unstuck, Groom Animal, Making Sandwich, Parkour, Parade, Flash Mob Gathering, BeeKeeping, Bike Tricks, and Dog Show.

[4]CoSum categories: Base Jump, Bike Polo, Eiffel Tower, Excavator River Crossing, Kids Playing in Leaves, MLB, NFL, Notre Dame Cathedral, Statue of Liberty, and Surfing.

4.2 Training Classification Network

The training splits of either datasets are not big enough to train our deep classification network well from scratch. Following the existing weakly-supervised summarisation method [21], we crawl additional video data of the same categories from YouTube for network pretraining. Specifically we search YouTube using the category names as queries. From the returned top-ranked videos, we filter out irrelevant videos using the following rules: (1) length not between 1 and 12 minutes; (2) contain multiple shots; (3) contain dynamic scenes; (4) no cartoons. This leads to 619 videos in total, roughly 30 videos for each category. We call this dataset *YouTube619* and use it *only* for pretraining the classification network. To test the classification accuracy on YouTube619, we randomly select 100 videos (5 per category) as test set and use the remaining 519 videos as training data to train our Bi-GRU network. We repeat such random split for 5 times and average the test accuracies, obtaining 89.6%. We then initialise the network with weights trained on YouTube619 and finetune on the target dataset. As a result, our Bi-GRU classifier with pretraining achieves 74% (TVSum) and 88% (CoSum), outperforming 66% (TVSum) and 72% (CoSum) obtained by the network trained from scratch.

4.3 Comparison with State-of-the-Art Methods

Method	Label	TVSum	CoSum
Uniform sampling	✗	15.5	20.4
K-medoids	✗	28.8	34.3
Dictionary selection [4]	✗	42.0	37.2
Online sparse coding [44]	✗	46.0	-
Co-archetypal [28]	✗	50.0	-
GAN [16]	✗	51.7	44.0
DR-DSN [46]	✗	57.6	47.8
LSTM [41]	frame-level	54.2	46.5
GAN [16]	frame-level	56.3	50.2
DR-DSN [46]	frame-level	58.1	54.3
Backprop-Grad [21]	video-level	52.7	46.2
DQSN (r^s)	video-level	57.9	50.1
DQSN ($r^s + r^u$)	video-level	58.1	51.7
DQSN ($r^s + r^l$)	video-level	58.2	52.0
DQSN (full model)	video-level	58.6	52.1

Table 1: Summarisation results (%) on TVSum and CoSum. $1^{st}/2^{nd}$ best in red/blue. Full model means $r^s + r^l + r^u$.

Table 1 compares our model, denoted as DQSN (full model), with the state-of-the-art on TVSum and CoSum. Our findings are summarised as follows. **(a)** *vs*. unsupervised: DQSN consistently outperforms all unsupervised approaches, often by large margins. These results suggest that the generic criteria employed in unsupervised learning are limited for not being able to adapt to different types of video content. Among them, the most competitive method is DR-DSN which is also RL-based (with only the generic DR reward). Comparing our DQSN (full model) with DR-DSN, the main difference is on introducing the weakly-supervised rewards (both local and global). Table 1 shows that by enforcing recognisability

on the generated summaries, our model significantly outperforms DR-DSN. In particular, the improvement margins are 1.0% and 4.3% on TVSum and CoSum, respectively. **(b)** *vs*. weakly supervised: We compare DQSN with the recently proposed method based on gradient back-propagation (Backprop-Grad) [21]. It can be seen that our model outperforms Backprop-Grad by 5.9% on both datasets. Both approaches employ recognisability as the frame selection criterion. The superiority of our model over Backprop-Grad can thus be explained by the fact that our RL-based framework can better capture the interdependencies among frames. **(c)** *vs*. supervised: Compared to the three supervised methods, LSTM [41], GAN [16] and DR-DSN [46], all of which require expensive frame-level annotation, our model is very competitive: it outperforms all three on TVSum; on CoSum, it is only slightly inferior to DR-DSN whilst beating the other two comfortably. Since our approach only uses video-level annotation, it is thus much more suited to large-scale applications.

4.4 Ablation Study

In this study we investigate how much different rewards contribute to the final model performance. Table 1 (bottom rows) compares DQSNs trained with different combinations of rewards (subscript t is omitted). We can see that $r^g + r^l$ clearly outperforms r^g on both datasets, strongly indicating the effectiveness of the local reward. By adding r^u, DQSNs are enhanced and achieve better F-scores (while $r^g + r^l + r^u$ still exhibits its advantage over $r^g + r^u$). This thus suggests that the unsupervised DR reward is complementary to our weakly-supervised reward and our approach is flexible enough to incorporate both.

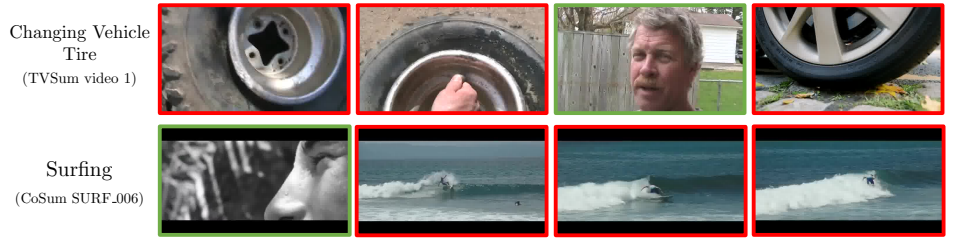


Figure 3: Example frames that downgraded (red) / improved (green) the rank of true category in classification when being removed.

Why Does Local Relative Importance Reward Help? To gain some insights into how the proposed dense local reward contributes, we show some frames that were removed and led to changes of the rank of true category in classification (see Fig. 3). In the first video, the frames containing the *Changing Tire* scene are important because removing them would downgrade the rank of the true category, whereas the frames containing the talking man are relatively unimportant. In the second video, *Surfing* frames are apparently more important than 'non-surfing' frames. These examples show that local frame relative importance measured by whether it triggers a rank change is indeed a good supervision signal for training the summarisation agent to take the correct action for keeping/removing a given frame.

4.5 Runtime Efficiency

We compare the summarisation time of Backprop-Grad [21] and DQSN under the same hardware condition[5] on TVSum + CoSum. Backprop-Grad runs at 3.21 second per video, while DQSN runs at 1.43 second per video, which is more than 2× faster. The reason is because Backprop-Grad performs forward and backward passes on each clip while DQSN only needs to do a forward pass. Moreover, DQSN consumes less memory as it does not save gradients as Backprop-Grad does.

4.6 Qualitative Results

Some example summaries are shown in Fig. 4. We observe that DQSN can extract category-related frames containing persons grooming the dogs and preserve well the temporal story-line. In contrast, Backprop-Grad tends to select repetitive scenes and fails to identify some important details such as the pink-clothed woman grooming the dog. DR-DSN achieves comparable performance to ours, but mistakenly selects irrelevant frames (*e.g.* frames containing dog food) to increase diversity. Such mistake is due to the fact the generic criteria used in DR-DSN are unable to adapt to the specific video content.

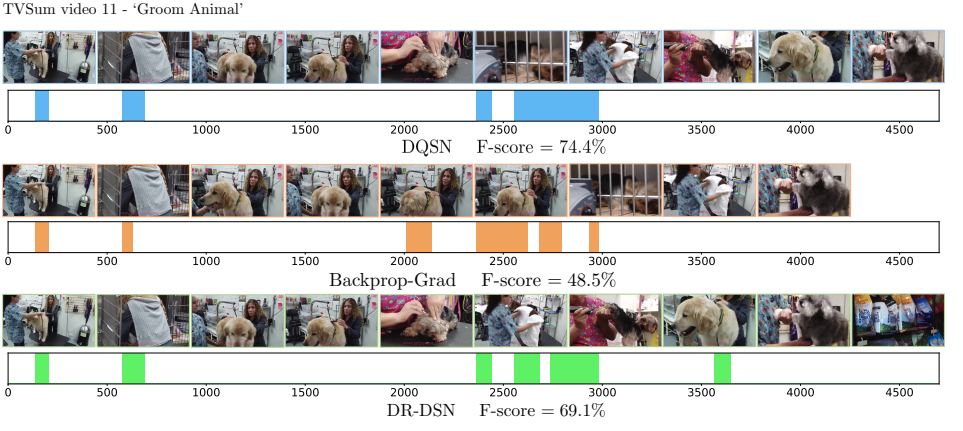


Figure 4: Sample summaries obtained by our DQSN, Backprop-Grad and unsupervised DR-DSN. The *x* axis is the timeline. Coloured segments represent summaries.

5 Conclusion

We presented a RL-based approach DQSN for video summarisation, which uses video-level category labels. A global recognisability reward was formulated to guide the learning of DQSN. Critically, a novel dense reward was proposed to mitigate the credit assignment problem in RL. Compared with unsupervised and supervised learning, our objective function can capture semantics while using only easy-to-obtain, video-level labels. Experimental results showed that our approach not only outperforms unsupervised/weakly supervised alternatives but is also highly competitive with supervised approaches.

[5] We used a GeForce GTX 1080 GPU.

References

[1] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

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

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *CVPR*, 2012.

[5] Boqing Gong, Wei-Lun Chao, Kristen Grauman, and Fei Sha. Diverse sequential subset selection for supervised video summarization. In *NIPS*, 2014.

[6] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

[7] Michael Gygli, Helmut Grabner, and Luc Van Gool. Video summarization by learning submodular mixtures of objectives. In *CVPR*, 2015.

[8] Hado V Hasselt. Double q-learning. In *NIPS*, 2010.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[10] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, 2017.

[11] Aditya Khosla, Raffay Hamid, Chih-Jen Lin, and Neel Sundaresan. Large-scale video summarization using web-image priors. In *CVPR*, 2013.

[12] Gunhee Kim, Leonid Sigal, and Eric P Xing. Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In *CVPR*, 2014.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[14] Shuyue Lan, Rameswar Panda, Qi Zhu, and Amit K. Roy-Chowdhury. Ffnet: Video fast-forwarding via reinforcement learning. In *CVPR*, 2018.

[15] Long-Ji Lin. *Reinforcement learning for robots using neural networks*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1993.

[16] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *CVPR*, 2017.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *NIPS Workshop*, 2013.

[18] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *CVPR*, 2018.

[19] Mayu Otani, Yuta Nakashima, Esa Rahtu, Janne Heikkilä, and Naokazu Yokoya. Video summarization using deep semantic features. In *ACCV*, 2016.

[20] Rameswar Panda and Amit K. Roy-Chowdhury. Collaborative summarization of topic-related videos. In *CVPR*, 2017.

[21] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K Roy-Chowdhury. Weakly supervised summarization of web videos. In *ICCV*, 2017.

[22] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop*, 2017.

[24] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2017.

[25] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *ECCV*, 2014.

[26] Zhou Ren, Xiaoyu Wang, Ning Zhang, Xutao Lv, and Li-Jia Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*, 2017.

[27] Mrigank Rochan and Yang Wang. Learning video summarization using unpaired data. *arXiv preprint arXiv:1805.12174*, 2018.

[28] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *CVPR*, 2015.

[29] James Supancic III and Deva Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *ICCV*, 2017.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[32] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

[33] Jingya Wang, Xiatian Zhu, and Shaogang Gong. Video semantic clustering with sparse and incomplete tags. In *AAAI*, 2016.

[34] Jingya Wang, Xiatian Zhu, and Shaogang Gong. Discovering visual concept structure with sparse and incomplete tags. *AI*, 2017.

[35] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018.

[36] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. In *ICML*, 2016.

[37] Huawei Wei, Bingbing Ni, Yichao Yan, Huanyu Yu, and Yang Xiaokang. Video summarization via semantic attended networks. In *AAAI*, 2018.

[38] Jia Xu, Lopamudra Mukherjee, Yin Li, Jamieson Warner, James M Rehg, and Vikas Singh. Gaze-enabled egocentric video summarization via constrained submodular maximization. In *CVPR*, 2015.

[39] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *ICCV*, 2015.

[40] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*, 2017.

[41] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *ECCV*, 2016.

[42] Li Zhang, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-critic sequence training for image captioning. In *NIPS Workshop*, 2017.

[43] Yujia Zhang, Michael Kampffmeyer, Xiaodan Liang, Dingwen Zhang, Min Tan, and Eric P Xing. Dtr-gan: Dilated temporal relational adversarial network for video summarization. *arXiv preprint arXiv:1804.11228*, 2018.

[44] Bin Zhao and Eric P Xing. Quasi real-time summarization for consumer videos. In *CVPR*, 2014.

[45] Bin Zhao, Xuelong Li, and Xiaoqiang Lu. Hsa-rnn: Hierarchical structure-adaptive rnn for video summarization. In *CVPR*, 2018.

[46] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *AAAI*, 2018.

[47] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. Video synopsis by heterogeneous multi-source correlation. In *ICCV*, 2013.

[48] Xiatian Zhu, Chen Change Loy, and Shaogang Gong. Learning from multiple sources for video summarisation. *IJCV*, 2016.