An Efficient Framework for Dense Video Captioning

Maitreya Suin, A. N. Rajagopalan
Indian Institute of Technology Madras
maitreyasuin21@gmail.com, raju@ee.iitm.ac.in

Abstract

Dense video captioning is an extremely challenging task since an accurate and faithful description of events in a video requires a holistic knowledge of the video contents as well as contextual reasoning of individual events. Most existing approaches handle this problem by first proposing event boundaries from a video and then captioning on a subset of the proposals. Generation of dense temporal annotations and corresponding captions from long videos can be dramatically source consuming. In this paper, we focus on the task of generating a dense description of temporally untrimmed videos and aim to significantly reduce the computational cost by processing fewer frames while maintaining accuracy. Existing video captioning methods sample frames with a predefined frequency over the entire video or use all the frames. Instead, we propose a deep reinforcement-based approach which enables an agent to describe multiple events in a video by watching a portion of the frames. The agent needs to watch more frames when it is processing an informative part of the video, and skip frames when there is redundancy. The agent is trained using actor-critic algorithm, where the actor determines the frames to be watched from a video and the critic assesses the optimality of the decisions taken by the actor. Such an efficient frame selection simplifies the event proposal task considerably. This has the added effect of reducing the occurrence of unwanted proposals. The encoded state representation of the frame selection agent is further utilized for guiding event proposal and caption generation tasks. We also leverage the idea of knowledge distillation to improve the accuracy. We conduct extensive evaluations on ActivityNet captions dataset to validate our method.

Introduction

A large share of today's internet traffic is video content. Many real-world problems from inventory management to self-driving cars depend on video which makes efficient video processing arguably the next horizon in computer vision. Producing robust video representation for solving different tasks such as action recognition (Simonyan and Zisserman 2014), generating textual descriptions (Donahue et al. 2015) and summaries (Pan et al. 2016), question answering (Jang et al. 2017) and so on, has proven to be much more challenging than learning deep image representations. This is in part due to the huge size of raw video streams and the presence of redundant information in the frames. Most of the existing frameworks, for every time step, need to pass the corresponding frame from the video through a Convolutional Neural Network (CNN) to get its feature representation. The redundancy in the frames makes it harder for CNNs to extract meaningful information. Also it becomes infeasible on low-end devices which have limitations of power, memory and computational speed as the amount of computation is proportional to the video length.

Compared to other video captioning datasets like Microsoft Video Description (MSVD) (Chen and Dolan 2011) and MSR Video to Text (MSRVTT) (Xu et al. 2016), ActivityNet Captions (Krishna et al. 2017) dataset have much longer videos containing multiple events with more descriptive sentences which makes the task of captioning computationally quite expensive and harder to optimize. Most of the existing works mainly focuses on increasing accuracy, while limited effort has been devoted to improving another critical aspect: efficiency (Zhang et al. 2016; Chen et al. 2018). State-of-the-art dense video captioning frameworks use the features from uniformly sampled frames, if not every single frame (Zhou et al. 2018; Mun et al. 2019). Few works in video action recognition (Bhardwaj, Srinivasan, and Khapra 2019) have tried to focus on efficiency using uniformly sampled frames. However, this assumes that information is evenly spread over time, which could include redundant frames that does not help the current task. Meanwhile, feeding all the frames is brute-force and introduces unnecessary computational burden. Frames which are not beneficial for the task at hand should be skipped as the analysis of even a single frame is computationally expensive due to the use of high-capacity backbone networks such as ResNet (He et al. 2016), InceptionNet (Szegedy et al. 2017), etc. One way of increasing efficiency is to design a faster feature extraction network or increase hardware capability. But, these are different dimensions of research which we do not address in this paper. Our focus is on how to adjust the computational complexity conditioned on the input video by selecting and processing a small number of informative frames. We propose an efficient network that dynamically selects informative frames conditioned on the input video for the task of dense video captioning inexpensively. Motivated by how humans generally approach to solve such a problem, we propose to formulate the frame selection task as a Markov decision processes. As there is no ground truth data available indicating which frames to pick, we use Reinforcement Learning (RL) algorithm to solve this problem. We use the performance score from different modules of the the dense captioning network as a reward and updates the agent using policy gradient method. It allows models to be optimized for long-term rewards which are useful in solving sequential decision-making problems. The workflow of our system is illustrated in Fig. 1. At every time step, the frame selection agent examines the history of frames and actions taken to decide where to look next. If the agent finds that the current frame does not increase performance, it can skip to a distant future. On the other hand, when finding informative frames it can slow down and observe closely to get a proper representation of the event. In the case of dense captioning, the quality of generated captions depends a lot on the performance of the proposal network as the captioning module describes only the events detected by it. Achieving reasonable accuracy on ActivityNet Captions dataset (Krishna et al. 2017) using only language score as a reward is difficult as the optimal performance demands the selected frames to cover all possible events in the video effectively. We empirically found that using only language score as a reward does not lead to satisfactory performance. Hence, we propose a carefully designed reward function which includes event proposal score and global representation score along with standard language score. The proposed reward encourages the agent to find a more informative frame which can improve the performance of event localization and corresponding caption generation directly. During training, the agent is optimized using policy gradient methods with a fixed maximum number of steps. The main contributions of the proposed approach are summarized as follows:

- We propose a novel framework for fast and efficient dense video captioning. The proposed frame-selection network effectively selects few informative frames that are capable of producing a similar score to that of using all the frames.
- We present a reinforcement-learning based method with novel rewards which drives both the event proposal and caption generation network to describe all the events effectively.
- We use knowledge distillation technique to improve the accuracy of our efficient model, which is pushed to match the output of the original network which uses all the frames.

We evaluate our proposed efficient dense video captioning model on the ActivityNet captions dataset (Krishna et al. 2017), which is currently one of the largest dense video captioning dataset.

Related Work

Video Captioning: Recent works in video captioning mainly use CNNs (He et al. 2016) for encoding video frames, followed by a recurrent language decoder. Apart from the use of spatial attention in image captioning (Xu et al. 2015), temporal attention is also utilized for video captioning (Yao et al. 2015). Few efforts have been made for increasing efficiency in related fields like video classification (Wu et al. 2019; Fan et al. 2018). Our approach is closest to (Chen et al. 2018) where they use RL-based approach to select a few important frames from each video to caption a single event. However, what we address is far more complex as we have to select different events throughout videos which are comparatively longer and then caption those events accurately. We focus on dense video captioning in the context of the ActivityNet Captioning dataset, where the average length of a video is more than 6 times the length of other standard video captioning datasets. (Krishna et al. 2017) introduced a dense video captioning model that proposes event locations and captions each event. (Mun et al. 2019) modeled temporal dependency across events using event sequence generation network and used RL for caption generation. (Zhou et al. 2018) uses self and cross-module attention inspired by machine translation methods (Vaswani et al. 2017) and achieves superior performance. We also leverage attention modules for encoding video frames and caption generation.

Temporal Event Proposal: Many advancements have been made to localize events in a long untrimmed video. (Shou, Wang, and Chang 2016) propose and classify proposal candidates directly over video frames in a sliding window fashion, which is computationally expensive. (Li et al. 2018) incorporate temporal coordinate and descriptiveness regressions for precise localization of events. We incorporate RL algorithm for selecting informative frames from the video. Due to the varying nature of different event boundaries throughout the dataset, for further refining the selected frames in accurate temporal regions we use a simplified network derived from (Zhou, Xu, and Corso 2018) due to its good performance and efficiency in dense event proposals.

Approach

Here, we formulate frames selection as sequential decision-making problem which naturally fits into the reinforcement learning framework. The model can be viewed as an agent that interacts with a video sequence. At every time step t, the agent takes the current frame features as input and employs a history-aware observation network to encode the explored environment and then feeds that into the policy network to generate a proper action deciding where to watch next. The goal of the agent is to derive an effective frame selection strategy that achieves reasonable accuracy in event localization as well as captioning while using as few frames as possible. The rest of the network consists of self-attention based visual encoder and two decoders: event proposal decoder and captioning decoder. During training, conditioned on the state, the frame selection network picks a frame, uses the visual encoder to get a representation of all the selected frames till the current time step. This is then passed through the two decoders to produce event proposals and corresponding captions. The outputs of these decoders are used to determine the reward which along with the guide network's output steers the frame selection network to pick more and more informative frames. The currently selected frame updates the state representation and this process continues till

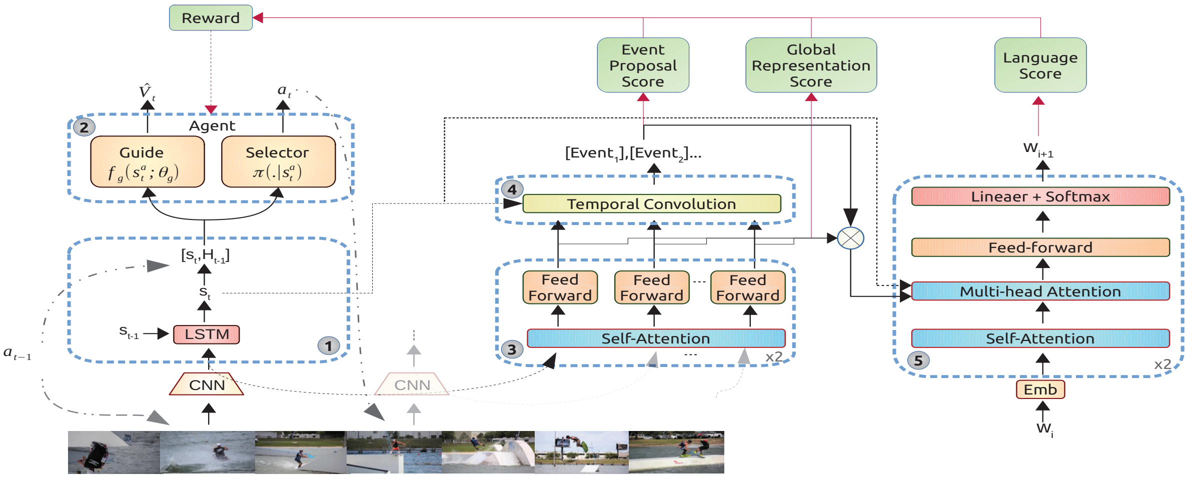


Figure 1: The overall architecture of the proposed network. Standard feature extractor CNN is used at the beginning for each selected frame. Then, the features are fed into the state encoder LSTM. The encoded state s_t after being concatenated with the historical actions H_{t-1} is fed to the agent. Features of all the frames selected by the agent are passed to the visual encoder for getting better representation. Further, it is passed through event proposal network and caption decoder which generates the output captions for the proposed event.

the end of the video or if it reaches the maximum number of time steps, whichever is earlier. All the modules are next described in detail.

State-Encoder (SE)

The State-Encoder network interacts with the video at each time step and generates state $\boldsymbol{s}_t^a$ at time step t containing historical information about frames and actions a_t, which serves as the input to the frame-selection network and guide network. Using a recurrent neural network is one way to model the state representation where instead of using the direct observation, history of observations ($\mathbf{s}_t$) might give better information. Intuitively, along with the visual history, the awareness about preceding actions should help the agent to make better decisions, as for e.g., if for the past few time steps the agent is watching nearby frames, it should fast-forward in forthcoming time steps as the goal is to cover all the events effectively and not spend unnecessary time in attending to a single event in the fixed time frame. Motivated by this, we utilize the historical actions to form the final state $\boldsymbol{s}_t^a$. The internal state $\mathbf{s}_t$ is formed by a recurrent neural network f_r, which is parameterized by θ_r and updated over time by taking the external frame feature vector $\boldsymbol{i}_t$ as input as:

$$\boldsymbol{s}_t = f_r(\boldsymbol{s}_{t-1}, \boldsymbol{i}_t; \theta_r) \tag{1}$$

Similarly the final state is formed by concatenation as

$$\boldsymbol{s}_t^a = [\boldsymbol{h}_{t-1}^a, \boldsymbol{s}_t] \tag{2}$$

where $\boldsymbol{h}_{t-1}^a = \boldsymbol{\pi}_a(\cdot|\boldsymbol{s}_{t-n}^a;\theta_{fa}) : \boldsymbol{\pi}_a(\cdot|\boldsymbol{s}_{t-1}^a;\theta_{fa})$ indicates the probability of actions taken from step $t-n$ to $t-1$ (Action probabilities $\boldsymbol{\pi}_a$ are discussed in detail in Frame-Selection Network section). Although visual encoder could have been utilized to get the state description, estimating it at every time step is unnecessary and computationally demanding. Rather, we use a relatively lightweight LSTM network for the representation of the state and use the self-attention based visual encoder to get the ultimate representation after selecting all the frames.

Frame-Selection Network (FSN)

Frame-Selection Network accepts the encoded information ($\mathbf{s}_t^a$) and transforms it into a policy which provides the probability of different actions to be exercised at the following time step. While training, we sample from this policy to determine where to watch next. Empirically we found that the discrete action space $A = [-4s, -2s, +2s, +4s, +8s, +16s, +32s]$ works well for most of the videos, where + indicates going forward and - indicates going backward. The actions are scaled according to the video length. Videos have local patterns and if the agent recognizes an event, it should examine nearby to get a proper representation of the event. The agent can also go back in case it advances too much and misses some information. The FSN is made of a fully connected layer f_{fs} which is parameterized by θ_{fs},followed by a softmax. This network outputs a multinomial probability distribution i.e., the policy $\pi_a(\cdot|\mathbf{s}_t^a;\theta_{fs})$, which gives the probability distribution of the actions to be taken at the next time step. The action a_t is sampled from π_a during training. During the inference, we use maximum aposteriori estimation to choose next action,i.e.,

$$a_t = \arg\max_{a\in A}(\boldsymbol{\pi}_a(\cdot|\boldsymbol{s_t^a};\theta_{fs})) \tag{3}$$

Guide Network (GN)

As time progresses, FSN learns to generate better and better actions that determines which frames to inspect and the Guide gets more experienced at assessing those. Guide Network has a fully connected layer f_g, which is parameterized by θ_g and it produces an output $f_u(\boldsymbol{s}_t;\theta_g) = \hat{V}_t$ which is known as the value function (Sutton, Barto, and others 1998). The job of the guide is to approximate the value function, that is expected future rewards from the current state

$$V_t = \mathbb{E}_{\substack{\boldsymbol{s}^a_{t+1:T},\\ a_{t:T}}} \sum_{i=0}^{T-t} \gamma^i r_{t+i}, \tag{4}$$

where γ is the discount factor which is utilized in determining cumulative discounted rewards. The actual reward, acquired from empirical rollouts, is then compared to the value predicted by the guide and used to update frame-selection network parameters in the direction of performance improvement. The use of V_t also reduces the variance and thus helps to expedite the learning of the algorithm.

Visual Encoder (VE)

Following the success of self-attention in natural language tasks for generating rich representation, we utilize it to get the final encoding of the frames picked by the agent. Feature representation of only the selected frames are fed to M number of encoding layers and each layer processes the output of the previous layer $\boldsymbol{E}_m$ as:

$$\boldsymbol{E}^{m+1} = \text{LN}(\text{FF}(\tilde{\boldsymbol{E}}^m) + \tilde{\boldsymbol{E}}^m) \tag{5}$$

$$\tilde{\boldsymbol{E}}^m = \text{LN}(\text{MHA}(\boldsymbol{E}^m, \boldsymbol{E}^m, \boldsymbol{E}^m) + \boldsymbol{E}^m) \tag{6}$$

where LN stands for layer normalization (Ba, Kiros, and Hinton 2016). MHA and FF denote the multi-head attention mechanism and the position-wise feed-forward network proposed by (Vaswani et al. 2017).

Additionally, following (Lei et al. 2018), we explore the use of object information in feature encoding by adding another semantically rich object embedding. The final few layers of an off-the-self light weight object detector trained with fixed backbone of ResNet-200 is used for extracting information about object classes present in a frame. We take average of the entire object embeddings of a frame to extract object representation ($\boldsymbol{V}_{ob}$). These features are then concatenated ($\boldsymbol{E}^m_{ob} = [\boldsymbol{E}^m, \boldsymbol{V}_{ob}]$) and passed to the event proposal and captioning network.

Event Proposal (EP) and Captioning Decoder (CD)

With encoded visual feature of all the selected frames, the event proposal network is used to recognize different event boundaries throughout the video. In the proposed approach, the job is much simpler. The frame selection network selects only a few frames covering the events beforehand. Due to the diverse nature of events in the videos like overlapping events, closely spaced events, etc. we further refine these frames in more precise temporal regions using a simpler temporal region proposal network derived from (Zhou, Xu, and Corso 2018) with the following modifications:

- We exclude the anchors containing frames less than a predefined threshold beforehand from processing as intuitively temporal regions with events should have an adequate number of frames to get proper representation. On the one hand, it drastically reduces the computational overhead by reducing the number of proposals thus making the task to be learned by the network more manageable. Also, it directly affects event proposal accuracy in case the frame selection network misses some event. Thus, it serves as a feedback to the agent and drives it to cover all the events adequately.
- Given the encoded features from the visual encoder, kernels of different sizes are slid over it to detect events. Along with this, to reason globally about the entire timeline when making predictions, we utilize the current state which contains holistic temporal information. Empirically we show that this improves event proposal accuracy.

These event proposals, along with the visual features are given to the captioning decoder. The output of this module are the captions to be predicted. Derived from the masked transformer architecture described in (Zhou et al. 2018), the N-layered captioning decoder generates the t^{th} word by performing the following operations:

$$\boldsymbol{Y}^{m+1}_{\leq t} = \text{LN}(\text{FF}(\Phi(\boldsymbol{Y}^n_{\leq t})) + \Phi(\boldsymbol{Y}^n_{\leq t})) \tag{7}$$

$$\Phi(\boldsymbol{Y}^n_{\leq t}) = \text{LN}(\text{MHA}([\Omega(\boldsymbol{Y}^n_{\leq t}), \boldsymbol{s}^t], \hat{\boldsymbol{E}}^n, \hat{\boldsymbol{E}}^n) + \Omega(\boldsymbol{Y}^n_{\leq t})) \tag{8}$$

$$\Omega(\boldsymbol{Y}^n_{\leq t}) = \text{LN}(\text{MHA}(\boldsymbol{Y}^n, \boldsymbol{Y}^n, \boldsymbol{Y}^n) + \boldsymbol{Y}^n) \tag{9}$$

$$p(\boldsymbol{w}_{t+1} | \boldsymbol{X}, \boldsymbol{Y}^N_{\leq t}) = \text{softmax}(\boldsymbol{W}^V \boldsymbol{y}^N_{t+1}) \tag{10}$$

where $\boldsymbol{y}^0_i$ represents word vector, $\boldsymbol{Y}^m_{\leq t} = \{\boldsymbol{y}^m_1, \ldots, \boldsymbol{y}^m_t\}$, $\boldsymbol{w}_{t+1}$ denotes the probability of each word in the vocabulary for time $t+1$, $\hat{\boldsymbol{E}}^m$ is the masked feature representation depending on the proposal. For detailed functioning of each block, we encourage the reader to refer to (Vaswani et al. 2017; Zhou et al. 2018). Some of the significant changes that we have adopted in the decoder design are:

- While determining language-conditioned image attention in Eq. (8), we additionally use the state value $\boldsymbol{s}_t$ along with self-attended language representation. Intuitively, $\boldsymbol{s}_t$ contains global knowledge about all the observed frames which will guide attention-heads to pick out more relevant visual information from the current proposal.
- As using fewer frames makes the job of dense captioning considerably harder for the network, we use the idea of knowledge distillation used in machine learning (Hinton, Vinyals, and Dean 2015; Romero et al. 2014). We incorporate an additional loss minimizing the difference between the probabilities produced by the captioning decoder which uses fewer frames and probabilities produced by the original decoder which uses all the frames. This enhances the efficiency of our captioning network as now the output from our efficient network utilizes much less information but seeks to match the output representation computed using all the frames.

Reward Function

The reward function reflects how good are the selected frames for a particular video. Each decision at a given state is associated with an immediate reward to measure the decision made by the agent at the current time. Most of the previous works use the primary objective as the reward. However as dense video captioning requires multiple event detection and corresponding captioning throughout the video, this is a much more demanding task to solve. We introduce a reward function that not only helps increase the language reward but also enhances event proposal network's accuracy, which in turn affects the quality of the generated captions. Additionally, we use an additional global representation reward, which encourages the frame-selection network to pick frames from all the events in a video. The reward at time step t can then be expressed as:

$$r_t = p_t - \max_{t' \varepsilon [0,t-1]} p_{t'} \tag{11}$$

where p_t is defined as:

$$p_t = \alpha \cdot LS_t + \beta \cdot ES_t + \nu \cdot GS_t - \mu \cdot t \tag{12}$$

LS_t,RS_t,GS_t are language score, event proposal score and global representation score at time step t respectively. Along with this we use $\mu \cdot t$ to penalize the agent if it watches more frames without significantly increasing the accuracy. The selected frames should contain rich semantic information, which can be used to effectively detect events. For this purpose, we have used negative of loss value from event proposal network which will push the agent to pick informative frames throughout the video so that event proposal network can localize events more precisely and the loss decreases. For language reward, we have used METEOR score which depicts the similarity of the machine-generated sentence to a majority of how most people explain the video. We have observed that adding global representation score which is nothing but the difference in encoded information estimated using only the selected frames and all the frames, helps the agent to cover all the events in a video more accurately and consistently. The reward function in Eqn. 11 encourages the score at every time step to improve from historical ones, which forces the frame-selection network to select more and more informative frames.

Optimization

The loss for training our model has mainly two parts: reinforcement loss and standard event proposal and captioning loss (Zhou et al. 2018). The event proposal network loss consists of standard classification and regression loss, whereas cross-entropy loss is used for captioning. Reinforcement loss also has two parts. The guide network is trained with the following regression loss:

$$\mathcal{L}_G(\theta_v) = \frac{1}{2} ||\hat{V}_t - V_t||_2. \tag{13}$$

The frame-selection network is trained to maximize the expected future reward:

$$J_{FS}(\theta_{fs}) = \mathbb{E}_{a_t \sim \pi_a(\cdot | \boldsymbol{s}_t^a; \theta_{fs})} \sum_{t=0}^{T} r_t \tag{14}$$

Following (Sutton, Barto, and others 1998), we derive the expected gradient of J_{FS} as:

$$\nabla_\Theta J_{FS} = \mathbb{E} \left[\sum_{t=0}^{T} (R_t - \hat{V}_t) \nabla_\Theta \log \pi_\theta(\cdot | \boldsymbol{s}_t^a; \theta_{fs}) \right] \tag{15}$$

where R_t denotes the expected future reward and Θ contains all trainable parameters. For detailed analysis of policy gradient technique we encourage the reader to refer to (Sutton, Barto, and others 1998). Using knowledge distillation technique to make our efficient network to match the performance of the network that uses all the frames, we try to minimize the difference between the probabilities predicted by the two as:

$$L_{\text{dist}}(\theta_v) = d(\boldsymbol{P}_{\text{all}}, \boldsymbol{P}_{\text{selected}}) \tag{16}$$

where d denotes a distance metric such as KL divergence or squared error loss (Bhardwaj, Srinivasan, and Khapra 2019). The final loss can be written as:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{cap}} + \lambda_2 \mathcal{L}_{\text{event}} + \lambda_3 \mathcal{L}_{\text{dist}} + \lambda_4 \mathcal{L}_G - \lambda_5 J_{FS}$$

Experiments

Datasets and Experimental Settings

ActivityNet Captions (Krishna et al. 2017) is one of the largest datasets containing multiple annotated temporal event segments and corresponding natural language sentence describing those events. It contains almost 20,000 YouTube videos which include 10,024, 4,926 and 5,044 videos for training, validation and test splits, respectively. The average number of temporal events and corresponding descriptions is 3.65 per video, where the descriptions are 13.48 words long on average. Since the testing labels are not publicly available, we report performance on the validation set.

Implementation details

To extract inputs for the encoder, the videos are downsampled every 0.5s and features are computed from the "Flatten-673" layer in ResNet-200 (He et al. 2016). Most of the previous works (Zhou et al. 2018) use motion features like optical flow additionally to improve performance. However, we only use the appearance features in our model, because extracting motion features is very time-consuming, which deviates from the very purpose of reducing computational cost. We leverage Adam (Kingma and Ba 2014) with an initial learning rate of 0.001. We apply the well-known regularization technique Dropout (Srivastava et al. 2014) to regularize the training and prevent over-fitting.

Evaluation Metrics The captioning performance is measured with the most commonly used evaluation metrics: BLEU$\{3,4\}$ and METEOR. We use the performance evaluation tool provided by the 2019 ActivityNet Captions Challenge. The evaluation metric takes both proposal accuracy and captioning accuracy into account. The scores of the metrics are summarized via their averages based on tIoU thresholds of 0.3,0.5,0.7 and 0.9 given identified proposals and generated captions.

Experimental Results and Analysis

Frame Selection Network. To explore what policy the frame-selection network learns, we fix the maximum number of frames the agent can watch to (8,10,15,20,25) in this experiment (Table 1). Basically, when the number of frames

Table 1: Captioning results from ActivityNet Caption Dataset validation split. † indicates methods using additional modalities (e.g. optical flow) for video representation. We report BLEU, CIDER, METEOR score. Event proposal results are measured in Average Recall(AR)

Model	M	C	B4	B3	AR
(Mun et al. 2019)	**8.82**	**30.68**	0.93	**2.94**	**55.58**
(Zhou et al. 2018)†	4.98	9.25	1.15	2.42	52.95
Ours - All	6.21	13.82	**1.35**	2.87	53.4
Ours - 25	**6.23**	**13.78**	**1.35**	**2.88**	**52.94**
Uniform - 25 + δ	5.94	11.98	1.42	2.73	51.18
Random - 25 + δ	5.76	11.02	1.20	2.42	49.4
Ours - 20	**6.18**	**13.20**	**1.18**	**2.70**	**51.94**
Uniform - 20 + δ	5.90	11.88	1.30	2.32	50.5
Random - 20 + δ	5.48	11.03	1.14	2.08	48.62
Ours - 15	**5.70**	**11.68**	**1.10**	**2.54**	**50.06**
Uniform - 15 + δ	5.44	9.24	0.97	2.18	48.71
Random - 15 + δ	5.20	8.86	0.94	1.98	46.80
Ours - 10	**5.12**	**9.02**	**0.94**	**2.10**	**47.18**
Uniform - 10 + δ	4.46	7.78	0.88	1.94	46.05
Random - 10 + δ	4.22	7.39	0.75	1.80	43.95
Ours - 8	**4.98**	**8.89**	**0.92**	**2.08**	**46.32**
Uniform - 8 + δ	4.27	7.56	0.84	1.93	44.02
Random - 8 + δ	4.05	7.22	0.71	1.79	42.86

Table 2: Effect of different training strategies on the performance. L,E,G,K,S denotes language score, event proposal score, global representation score, knowledge distillation loss,use of state encoder representation(s_t) in EP and CD module respectively. Maximum number of frames to be watched is fixed at 15 for this experiment

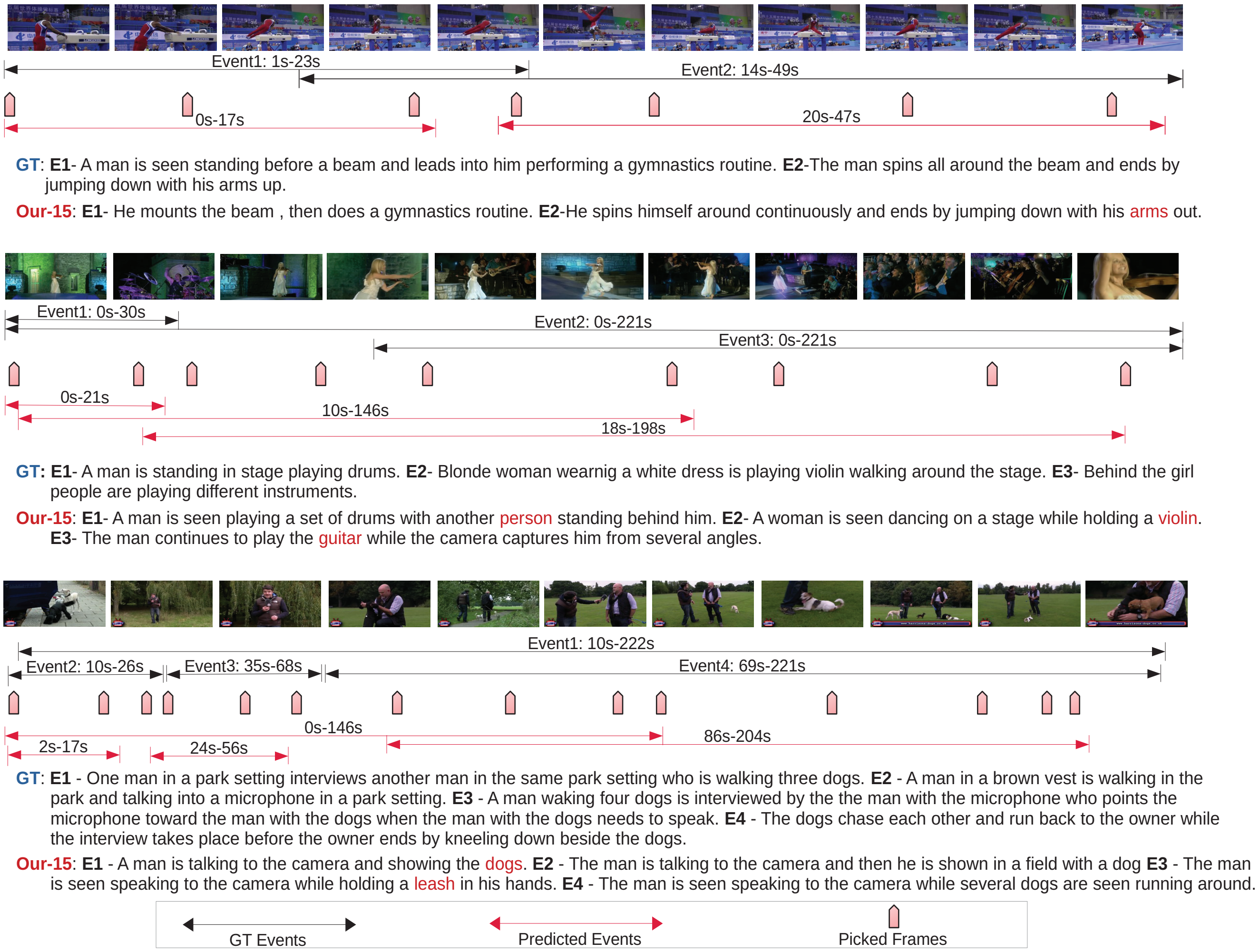
Method	M	C	B4	B3	AR
Ours (L+E+G+K+S)	**5.70**	**11.68**	**1.10**	**2.54**	**50.06**
Ours (L+E+G+S)	5.61	11.43	1.03	2.42	49.87
Ours (L+E+G+K)	5.14	10.49	0.98	2.20	48.92
Ours (L+E+K+S)	5.32	11.21	1.06	2.33	49.13
Ours (L+G+K+S)	5.07	11.14	0.98	2.14	47.86
Ours (L+K+S)	4.93	9.75	0.82	1.78	44.87

allowed to be watched increases, the accuracy improves. However, the rate at which it increases gradually goes down as we make more frames available to watch (Figure 3). This indicates that increasing the number of watched frames may not always significantly improve the performance. This experiment supports the claim that there are redundant frames in a video and those frames can be safely skipped without much decrease in the accuracy. To compensate for the extra computation in our proposed method compared to static frame selection methods, we watch extra δ frames in those cases. Also, for fair comparison we keep the training strategies same wherever applicable while using all, uniformly sampled or randomly sampled frames.

Computational advantage: Note that the computational cost increases linearly with the number of frames watched. This is because the most expensive operation is extracting features with CNNs. For ResNet-200, around 15 GFLOPs are required to compute features. The proposed modifications in our approach requires additional 1.8 GFLOPS compared to (Zhou et al. 2018). We show that there is a scope of improving the efficiency without sacrificing on the accuracy much. Although the frame selection network requires additional computations, the accuracy surpasses those of static frame selection methods. The computational saving is more than 90% compared to using all the frames while giving equivalent performance. When compared to static frame sampling methods, we achieve similar or more accuracy with almost 30% less computations on average.

Analyses of learned policies To verify if the learning by the frame-selection network is meaningful, we visualize the selected frames for different videos containing diverse events in Figure 2. We fix the maximum number of frames the agent can watch to 15. Although some of the previous work in video classification implemented a separate network to decide when to stop (Fan et al. 2018), for the present task multiple events are spread throughout the video and we found that adaptive stopping harms the agent's ability to cover all the events effectively. Forced by the penalty used in reward in Eq. 12, the agent tries to watch less frames without sacrificing the accuracy. Note that increasing the penalty is different from reducing the maximum number of frames. Training with penalty will make the agent capable of dynamically judging the difficulty level of a video as it is expected to keep a balance between accuracy and penalty. We observe that, for relatively simple events (e.g., "A man is standing in stage playing drums.") the frame selection network selects very few frames from the corresponding temporal region and fast-forwards quickly; while for some complicated scenarios (e.g., "The dogs chase each other and run back to the owner while the interview takes place before the owner ends by kneeling down beside the dogs."), it tends to take additional time steps to understand the events (Figure 2). To further analyze the effect of penalty, we fix maximum number of frames to $\{8,10,15,20,25\}$ and vary the value of μ. When μ increases, both computational complexity and performance drops as shown in Figure 3. However our frame-selection network still outperforms static frame selection strategies for similar complexity.

For a better understanding of each term used in the reward function, we analyze the effect of those separately (Table 2). Earlier for video classification (Wu et al. 2019) or captioning task (Chen et al. 2018), the main objective like classification accuracy or language score has been used as the primary reward. But, for a more difficult job like dense video captioning on ActivityNet captions dataset, we found it to be insufficient. We predict that as different events are spread throughout the video, and describing those events requires accurate temporal region proposal, only using language reward results in sub-optimal performance. Experimentally, we found that the language reward improves the quality of picked frames inside an event by selecting more diverse frames, which helps the captioning module to describe better. However, it fails to cover all the events consistently. Event proposal score, which depicts the accuracy of the output from the event proposal network, makes the selected frames to cover event boundaries more accurately. We also observe that adding global representation score makes the agent cover all significant events consistently. This score



GT: **E1**- A man is seen standing before a beam and leads into him performing a gymnastics routine. **E2**-The man spins all around the beam and ends by jumping down with his arms up.

Our-15: **E1**- He mounts the beam , then does a gymnastics routine. **E2**-He spins himself around continuously and ends by jumping down with his arms out.

GT: E1- A man is standing in stage playing drums. **E2**- Blonde woman wearnig a white dress is playing violin walking around the stage. **E3**- Behind the girl people are playing different instruments.

Our-15: **E1**- A man is seen playing a set of drums with another person standing behind him. **E2**- A woman is seen dancing on a stage while holding a violin. **E3**- The man continues to play the guitar while the camera captures him from several angles.

GT: **E1** - One man in a park setting interviews another man in the same park setting who is walking three dogs. **E2** - A man in a brown vest is walking in the park and talking into a microphone in a park setting. **E3** - A man waking four dogs is interviewed by the the man with the microphone who points the microphone toward the man with the dogs when the man with the dogs needs to speak. **E4** - The dogs chase each other and run back to the owner while the interview takes place before the owner ends by kneeling down beside the dogs.

Our-15: **E1** - A man is talking to the camera and showing the dogs. **E2** - The man is talking to the camera and then he is shown in a field with a dog **E3** - The man is seen speaking to the camera while holding a leash in his hands. **E4** - The man is seen speaking to the camera while several dogs are seen running around.

Figure 2: Visualization of picked frames and generated captions.

is highest when the agent does not miss any event so that the difference in encoded information is lowest. We further analyze the effect of knowledge distillation loss on the accuracy. Forcing the efficient network to mimic the performance of the original network which uses all the frames increases the learning capability of our network. Experimental results are shown in Table 2. Utilization of encoded state representation s_t further improves the performance (Table 2).

Conclusions

In this paper, we presented an efficient framework for dense captioning of videos. We utilize reinforcement-learning to derive a frame-selection policy with an aim to reduce the overall computational cost. We proposed a novel reward function which helps the agent to learn optimal policy in the comparatively difficult task of dense video captioning. We also use knowledge distillation technique to improve the performance. The proposed architecture has good flexibility and could be potentially employed to other complex video-related tasks which will be further addressed in our future work.

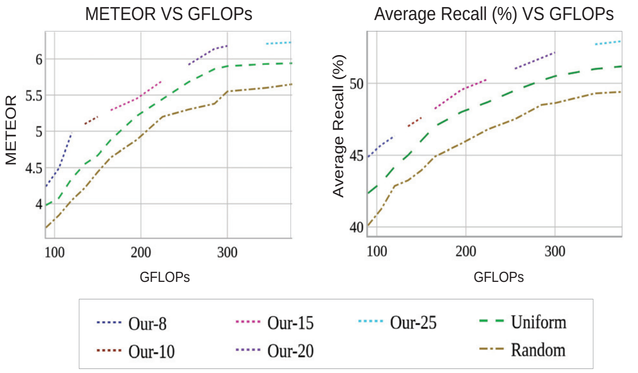


Figure 3: METEOR Score and Average Recall(%) vs. computational cost. We compare our method with static frame selection methods where the number denotes the maximum frames the agent can watch.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bhardwaj, S.; Srinivasan, M.; and Khapra, M. M. 2019. Efficient video classification using fewer frames. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 354–363.

Chen, D. L., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 190–200. Association for Computational Linguistics.

Chen, Y.; Wang, S.; Zhang, W.; and Huang, Q. 2018. Less is more: Picking informative frames for video captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 358–373.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2625–2634.

Fan, H.; Xu, Z.; Zhu, L.; Yan, C.; Ge, J.; and Yang, Y. 2018. Watching a small portion could be as good as watching all: Towards efficient video classification. In *IJCAI*, volume 2, 6.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2758–2766.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, 706–715.

Lei, J.; Yu, L.; Bansal, M.; and Berg, T. L. 2018. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*.

Li, Y.; Yao, T.; Pan, Y.; Chao, H.; and Mei, T. 2018. Jointly localizing and describing events for dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7492–7500.

Mun, J.; Yang, L.; Ren, Z.; Xu, N.; and Han, B. 2019. Streamlined dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6588–6597.

Pan, P.; Xu, Z.; Yang, Y.; Wu, F.; and Zhuang, Y. 2016. Hierarchical recurrent neural encoder for video representation with application to captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1029–1038.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Shou, Z.; Wang, D.; and Chang, S.-F. 2016. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1049–1058.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, 568–576.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Introduction to reinforcement learning*, volume 2. MIT press Cambridge.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wu, Z.; Xiong, C.; Ma, C.-Y.; Socher, R.; and Davis, L. S. 2019. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1278–1287.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, 2048–2057.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C.; Larochelle, H.; and Courville, A. 2015. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, 4507–4515.

Zhang, B.; Wang, L.; Wang, Z.; Qiao, Y.; and Wang, H. 2016. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2718–2726.

Zhou, L.; Zhou, Y.; Corso, J. J.; Socher, R.; and Xiong, C. 2018. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8739–8748.

Zhou, L.; Xu, C.; and Corso, J. J. 2018. Towards automatic learning of procedures from web instructional videos. In *Thirty-Second AAAI Conference on Artificial Intelligence*.