Less Is More: Picking Informative Frames for Video Captioning

Yangyu Chen[1], Shuhui Wang[2⋆], Weigang Zhang[3] and Qingming Huang[1,2]

[1]University of Chinese Academy of Science, Beijing, 100049, China
[2]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, 100190, China
[3]Harbin Inst. of Tech, Weihai, 264200, China
yangyu.chen@vipl.ict.ac.cn, wangshuhui@ict.ac.cn,
wgzhang@hit.edu.cn, qmhuang@ucas.ac.cn
webpage: https://yugnaynehc.github.io/picknet

Abstract. In video captioning task, the best practice has been achieved by attention-based models which associate salient visual components with sentences in the video. However, existing study follows a common procedure which includes a frame-level appearance modeling and motion modeling on equal interval frame sampling, which may bring about redundant visual information, sensitivity to content noise and unnecessary computation cost. We propose a plug-and-play PickNet to perform informative frame picking in video captioning. Based on a standard encoder-decoder framework, we develop a reinforcement-learning-based procedure to train the network sequentially, where the reward of each frame picking action is designed by maximizing visual diversity and minimizing discrepancy between generated caption and the ground-truth. The rewarded candidate will be selected and the corresponding latent representation of encoder-decoder will be updated for future trials. This procedure goes on until the end of the video sequence. Consequently, a compact frame subset can be selected to represent the visual information and perform video captioning without performance degradation. Experiment results show that our model can achieve competitive performance across popular benchmarks while only $6\sim8$ frames are used.

1 Introduction

Human are born with the ability to identify useful information and filter redundant information. In biology, this mechanism is called sensory gating [6], which describes neurological processes of filtering out unnecessary stimuli in the brain from all possible environmental stimuli, thus prevents an overload of redundant information in the higher cortical centers of the brain. This cognitive mechanism is essentially consistent with a huge body of researches in computer vision [13].

As one of the strong evidences practicing on visual sensory gating, attention is introduced to identify the salient visual regions with high objectness and meaningful visual patterns of an image [21, 48]. It has also been established on videos that contains consecutive image frames. Existing study follows a common procedure which includes a frame-level appearance modeling and motion modeling on equal interval frame sampling, say, every 3 frames or 5 frames [29]. Visual features and motion features are

⋆ Corresponding author.

extracted on the selected frame subset one by one, and they are all fed into the learning stage. Similar to image, the video attention is recognized as a spatial-temporal saliency that identifies both salient objects and their motion trajectories [27]. It is also recognized as the word-frame association learned by sparse coding [41] or gaze-guided attention learning [45], which is a de-facto frame weighting mechanism. This mechanism also benefits many downstream tasks such as visual captioning and visual question answering for image and video [20, 43, 12].

(a) Equally sampled 30 frames from a video

(b) Informative frames

Fig. 1: An illustration of the temporal redundancy in video. Video always contains many redundant information. The whole video can be represented by a small portion of frames (b), while equally sampled frames still contain redundant information (a).

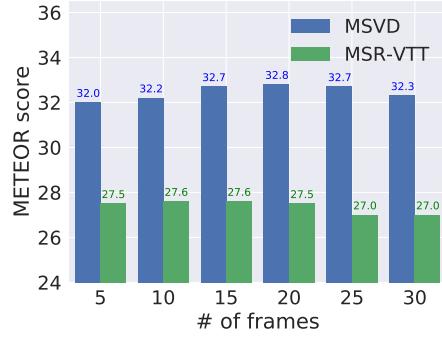


Fig. 2: The best METEOR score on the validation set of MSVD and MSR-VTT when using different number of equally sampled frames. The standard encoder-decoder model is used to generate captions.

Despite of the success on bridging vision and language achieved by existing attention-based methods, there still exists critical issues to be addressed as follows.

- **Frame selection perspective.** As shown in Figure 1(a), there are many frames with duplicated and redundant visual appearance information selected with equal interval frame sampling. This will also involve remarkable computation expenditures and less performance gain as the information from the input is not appropriately sampled. For example, it takes millions of floating point calculation to extract a frame-level visual feature for a moderate-sized CNN model. Moreover, there is no guarantee that all the frames selected by equal interval sampling contain meaningful information, so it tends to be more sensitive to content noise such as motion blur, occlusion and object zoom-out.
- **Downstream video captioning task perspective.** Previous attention-based models mostly identify the spatial layout of visual saliency, but the temporal redundancy existing in neighboring frames remains unsolved as all the frames are taken into consideration. This may lead to an unexpected information overload on the visual-linguistic correlation analysis model. For example, the dense-captioning-based strategy [27, 17, 14] can potentially describe images/videos in finer levels of detail by captioning many visual regions within an image/video-clip. With an

increasing number of frames, many highly similar visual regions will be generated and the problem will become prohibitive as the search space of sequence-to-sequence association becomes extremely large. We have conducted a preliminary study to investigate how many frames is enough for video captioning on two benchmarks. As shown in Figure 2, using more frames may not always lead to better performance, since sampling more frames may be prone to contain noisy information, and makes the training procedure more difficult.

– **Human perception perspective.** The vision-to-language technique can be applied to depict ambient information for human, such as describing the road conditions through voice broadcast for drivers. Based on existing video captioning methods, a naive way for generating such descriptions for endless visual streaming is to sample frames in every fixed time interval. However, it is problematic to determine an appropriate interval. If the interval is too long, some useful information may be missed and lead to wrong description. If the interval is too short, repeated descriptions will be generated as the visual content may not change largely, which is annoying for drivers as they focus on the change of surroundings. Therefore, it is necessary to explore a more appropriate strategy to capture informative frames and produce meaningful descriptions.

To deal with the above issues, we propose PickNet to perform informative frame picking for video captioning. Specifically, the base model for visual-linguistic association in video captioning is a standard encoder-decoder framework [2]. We develop a reinforcement-learning-based procedure to train the network sequentially, where the reward of each frame picking action is designed by considering both visual and textual cues. From visual perspective, we maximize the diversity between current picked frame candidate and the selected frames. From textual perspective, we minimize the discrepancy between the generated caption and the ground truth using current picked candidate. The rewarded candidate will be selected and the corresponding latent representation of encoder-decoder will be updated for future trials. This procedure goes on until the end of the video sequence. Consequently, a compact frame subset can be selected to represent the visual information and perform video captioning without performance degradation.

To the best of our knowledge, this is the first study on online task-driven frame selection for video captioning. Different from the previous work [46] that summarizing the video before video captioning, our method selects frames under **partially observed** settings and **do not need any auxiliary annotation or information**. It is very essential for real-world applications, since the video summarization annotations are subjective and expensive, and there is no trimmed video to summarize in real-world applications, but only endless visual streams. In fact, our framework can go beyond the encoder-decoder framework in video captioning task, and serves as a complementary building block for other state-of-the-art solutions. It can also be adapted by other task-specific objectives for video analysis. In summary, the merits of our PickNet include:

– **Flexibility.** We design a plug-and-play reinforcement-learning-based PickNet to pick informative frames for video captioning. A compact frame subset can be selected to represent the visual information and perform video captioning without performance degradation.

- **Efficiency.** The architecture can largely cut down the usage of convolution operations. It makes our method more applicable for real-world video processing.
- **Effectiveness.** Experiment shows that our model can achieve comparable or even better performance compared to state-of-the-art while only a small number of frames are used.

2 Related Works

2.1 Visual captioning

The visual captioning is the task translating visual contents into natural language. Early to 2002, Kojima *et al.* [16] proposed the first video captioning system for describing human behavior. From then on, a series of image and video captioning studies have been conducted. Early approaches tackle this problem using bottom-up paradigm [9, 18, 40, 8], which first generate descriptive words of an image by attribute learning and object recognition, then combine them by language models which fit predicted words to predefined sentence templates. With the development of neural networks and deep learning, modern captioning systems are based on CNN, RNN and the encoder-decoder architecture [35, 36].

An active branch of captioning is utilizing the attention mechanism to weigh the input features. For image captioning, the mechanism is typically in the form of spatial attention. Xu *et al.* [39] first introduced an attention-based model that automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. For video captioning, the temporal attention is added. Yao *et al.* [41] took into account both the local and global temporal structure of videos to produce descriptions, and their model is learned to automatically select the most relevant temporal segments given the text-generating RNN. However, the attention-based methods, especially temporal attention, are operated on full observed condition, which is not suitable in some real world applications, such as blind navigation. Our method do not require the global information of videos, which is more effective in these applications.

2.2 Frame selection

Selecting informative video frames is the most studied in the field of video summarization. This problem may be formulated as image searching. For example, Song *et al.* [32] considered images related to the video title that can serve as a proxy for important visual concepts, so they developed a co-archetypal analysis technique that learns canonical visual concepts shared between video and images, and used it to summarize videos. Other researchers use sparse learning to deal with this problem. Zhao *et al.* [47] proposed to learn a dictionary from given video using group sparse coding, and the summary video was then generated by combining segments that cannot be sparsely reconstructed using the learned dictionary.

Some video analysis task cooperates with frame selection mechanism. For example, in action detection, Yeung *et al.* [42] designed a policy network to directly predict the temporal bounds of actions, which decreased the cost of processing the whole video,

and improved the detection performance. However, the prediction made by this method is in the form of normalized global position, which requires the knowledge of the video length, making it unable to deal with real video streams. Different from the above methods, our model selects frames based on both semantic and visual information, and does not need to know the global length of video.

3 Method

Our method can be viewed as inserting the play-and-plug PickNet into the standard encoder-decoder for video captioning. The PickNet sequentially picks informative frames to generate a compact frame subset which properly represent the visual information of input video. And the encoder-decoder uses this subset to generate sentence description about the video.

3.1 Preliminary

Like most of video captioning methods, our model is built on the encoder-decoder-based sentence generator. In this subsection, we briefly introduce this building block.

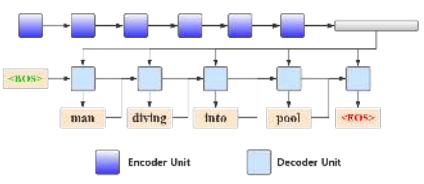


Fig. 3: The encode-decode procedure for video captioning.

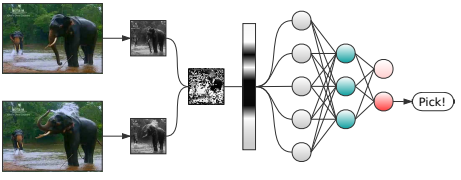


Fig. 4: The PickNet uses the flattened difference gray-scale image as input and produces a Bernoulli distribution to indicate picking the current frame or not.

Encoder. Given an input video, we use a recurrent video encoder which takes a sequence of visual features $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ as input and outputs a fixed size vector $\mathbf{v}$ as the representation of this video. The encoder is built on top of a Long Short-Term Memory (LSTM) [11] unit, which has been widely used for video encoding, since it is known to properly deal with long range temporal dependencies. Different from vanilla recurrent neural network unit, LSTM introduces a memory cell $\mathbf{c}$ which maintains the history of the inputs observed up to a time-step. The update operations on memory cell are controlled by input gate $\mathbf{i}_t$ that controls how the current input should be added into memory cell, forget gate $\mathbf{f}_t$ that controls what the current memory cell $\mathbf{c}_t$ will forget from the previous memory $\mathbf{c}_{t-1}$, and output gate $\mathbf{o}_t$ that controls how the current memory cell should be passed as output. These gates all take the combination of the frame feature $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$ as input, and the sigmoid activation is used to avoid gradient vanishing or exploding. The hidden state $\mathbf{h}$ and memory cell $\mathbf{c}$

are initialized to zero vector. And the last hidden state $\mathbf{h}_T$ is used as the final encoded video representation $\mathbf{v}$.

Decoder and sentence generation. Once the representation of the video has been generated, the recurrent decoder can employ it to generate the corresponding description. At every time-step of the decoding phase, the decoder unit uses the encoded vector $\mathbf{v}$, previous generated one-hot representation word $\mathbf{w}_{t-1}$ and previous internal state $\mathbf{p}_{t-1}$ as input, and outputs a new internal state $\mathbf{p_t}$. Like [2], our decoder unit is the Gated Recurrent Unit (GRU) [5], a simplified version of LSTM, which is good at language decoding. The output of GRU is modulated via two sigmoid gates: a reset gate $\mathbf{r}_t$ which determines how the previous internal state should be dropped to generate the next outputs, and an update gate $\mathbf{z}_t$ which controls how much information of the previous internal state should be preserved. A softmax function is applied on $\mathbf{p}_t$ to compute the probability of producing certain word at current time-step:

$$p_\omega(\mathbf{w}_t|\mathbf{w}_{t-1}, \mathbf{w}_{t-2}, ..., \mathbf{w}_1, \mathbf{v}) = \mathbf{w}_t^T \text{softmax}(W_p \mathbf{p}_t), \tag{1}$$

where W_p is used to project the output of the decoder to the dictionary space and ω denotes all parameters of the encoder-decoder. Also, the internal state $\mathbf{p}$ is initialized to zero vector. We use the greedy decode routine to generate every word. It means that at every time-step, we choose the word that has the maximal $p_\omega(\mathbf{w}_t|\mathbf{w}_{t-1}, \mathbf{w}_{t-2}, ..., \mathbf{w}_1, \mathbf{v})$ as the current output word. Specifically, we use a special token <BOS> as $\mathbf{w}_0$ to start the decoding, and when the decoder generates another special token <EOS>, the decoding procedure is terminated.

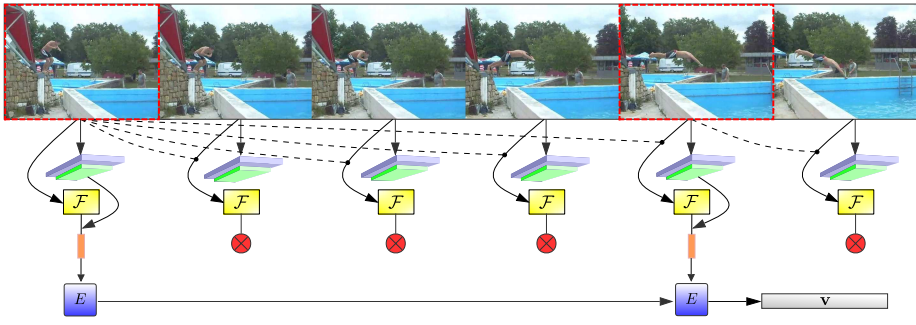


Fig. 5: A typical frame picking and encoding procedure of our framework. $\mathcal{F}$ denotes PickNet. E is the encoder unit and $\mathbf{v}$ is the encoded video representation. The design choice is the balance between processing time and computation cost. The system can simultaneously extract convolutional features and decide whether to pick the frame or not at each time-step. If it decides not to pick the frame at certain time-step, the convolutional neural network can stop early to save computation cost.

3.2 Our approach

Architecture. The PickNet aims to select informative video content without knowing the global information. It means that the pick decision can only be based on the current

observation and the history, which makes it more difficult than video summarization tasks. The more challenging issue is, we do not have supervision information to guide the learning of PickNet in video captioning tasks. Therefore, we formulate the problem as a reinforcement learning task, *i.e.*, given an input image sequence sampled from a video, the agent should select a subset of them under certain policy to retain video content as much as possible. Here, we use PickNet to produce the picking policy. Figure 4 shows the architecture of PickNet.

Considering the computation efficiency, we use a simple two-layer feedforward neural network as the prototype of PickNet. The network has two outputs, which indicate the probabilities to pick or drop the current observed frame. We model the frame picking process as the glance-and-compare operation. For each input frame $\mathbf{z}_t$, we first convert the colored image into grayscale image, and then resize it into a smaller image $\mathbf{g}_t$, which can be viewed as a "glance" of current frame. Then we subtract the current glance $\mathbf{g}_t$ by the glance of the last picked frame $\tilde{\mathbf{g}}$, to get a grayscale difference image $\mathbf{d}_t$; this can be seen as the "compare". Finally we flatten the 2D grayscale difference image into a 1D fixed size vector, and feed it to PickNet to produce a Bernoulli distribution that the pick decision is sampled from:

$$\mathbf{s}_t = W_2(\max(W_1 \text{vec}(\mathbf{d}_t) + \mathbf{b}_1, \mathbf{0})) + \mathbf{b}_2 \tag{2}$$

$$p_\theta(a_t|\mathbf{z}_t, \tilde{\mathbf{g}}) \sim \text{softmax}(\mathbf{s}_t), \tag{3}$$

where W_* are learned weight matrices and $\mathbf{b}_*$ are learned bias vectors.

During training, we use stochastic policy, *i.e.*, the action is sampled according to Equation (3). When testing, the policy becomes determined, hence the action with higher probability is chosen. If the policy decides to pick the current frame, the frame feature will be extracted by a pretrained CNN and embedded into a lower dimension, then passed to the encoder unit, and the template will be updated: $\tilde{\mathbf{g}} \leftarrow \mathbf{g}_t$.

We force PickNet to pick the first frame, thus the encoder will always process at least one frame, which makes the training procedure more robust. Figure 5 shows how PickNet works with the encoder. It is worth noting that the input of PickNet can be of any other forms, such as the difference between optical flow maps, which may handle the motion information more properly.

Rewards. The design of rewards is very essential to reinforcement learning. For the purpose of picking informative video frames, we consider two parts for the reward: the language reward and visual diversity reward.

Language reward. First of all, the picked frames should contain rich semantic information, which can be used to effectively generate language description. In the video captioning task, it is natural to use the evaluated language metrics as the language reward. Here, we choose CIDEr [33] score. Given a set of picked frames $\mathcal{V}_i$ for video v_i and a collection of human generated reference sentences $S_i = \{s_{ij}\}$, the goal of CIDEr is to measure the similarity of the machine generated sentence c_i to a majority of how most people describe the video. So the language reward r_l is defined as:

$$r_l(\mathcal{V}_i, S_i) = \text{CIDEr}(c_i, S_i) \tag{4}$$

Visual diversity reward. Also, we want the picked frames that have good diversity in visual features. Using only language reward may miss some important visual information, so we introduce the visual diversity reward r_v. For all the selected frame features $\{\mathbf{x}_k \in \mathbb{R}^D\}$, we use the pairwise cosine distance to construct the visual diversity reward:

$$r_v(\mathcal{V}_i) = \frac{2}{N_p(N_p-1)} \sum_{k=1}^{N_p-1} \sum_{m>k}^{N_p} (1 - \frac{\mathbf{x}_k^{\mathbf{T}} \mathbf{x}_m}{\|\mathbf{x}_k\|_2 \|\mathbf{x}_m\|_2}), \tag{5}$$

where N_p is the number of picked frames, $\|\cdot\|_2$ is the 2-norm of a vector.

Picks limitation. If the number of picked frames is too large or too small, it may lead to poor performances in either efficiency or effectiveness. So we assign a negative reward to discourage this situations. Empirically, we set the minimum picked number N_{min} as 3, which stands for beginning, highlight and ending. The maximum picked number N_{max} is initially set as the $\frac{1}{2}$ of total frame number, and will be shrunk down along with the training process, until decreased to a minimum value τ.

In summary, we merge the two parts of reward, and the final reward can be written as

$$r(\mathcal{V}_i) = \begin{cases} \lambda_l r_l(\mathcal{V}_i, S_i) + \lambda_v r_v(\mathcal{V}_i) & \text{if} \quad N_{\text{min}} \leq N_p \leq N_{\text{max}} \\ R^- & \text{otherwise,} \end{cases} \tag{6}$$

where λ_* is the weighting hyper-parameters and R^- is the penalty.

3.3 Training

The training procedure is splitted into three stages. The first stage is to pretrain the encoder-decoder. We call it *supervision* stage. In the second stage, we fix the encoder-decoder and train PickNet by reinforcement learning. It is called *reinforcement* stage. And the final stage is the joint training of PickNet and the encoder-decoder. We call it *adaptation* stage. We use standard back-propagation to train the encoder-decoder, and REINFORCE [37] to train PickNet.

Supervision stage. When training the encoder-decoder, traditional method maximizes the likelihood of the next ground-truth word given previous ground-truth words using back-propagation. However, this approach causes the *exposure bias* [25], which results in error accumulation during generation at test time, since the model has never been exposed to its own predictions. In order to alleviate this phenomenon, the schedule sampling [3] procedure is used, which feeds back the model's own predictions and slowly increases the feedback probability during training. We use SGD with cross entropy loss to train the encoder-decoder. Given the ground-truth sentences $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m)$, the loss is defined as:

$$L_{\text{X}}(\omega) = -\sum_{t=1}^{m} \log(p_\omega(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ldots \mathbf{y}_1, \mathbf{v})), \tag{7}$$

where $p_\omega(\mathbf{y}_t | \mathbf{y}_{t-1}, \mathbf{y}_{t-2}, \ldots \mathbf{y}_1, \mathbf{v})$ is given by the parametric model in Equation (1).

Reinforcement stage. In this stage, we fix the encoder-decoder and treat it as the *environment*, which can produce language reward to reinforce PickNet. The goal of training is to minimize the negative expected reward:

$$L_R(\theta) = -\mathbb{E}\left[r(\mathcal{V}_i)\right] = -\mathbb{E}_{\mathbf{a}^s \sim p_\theta}\left[r(\mathbf{a}^s)\right], \tag{8}$$

where θ denotes all parameters of PickNet, p_θ is the learned policy parameterized by Equation (3), and $\mathbf{a}^s = (a_1^s, a_2^s, \ldots, a_T^s)$ is the action sequence, in which a_t^s is the action sampled from the learned policy at the time step t. s is a superscript to indicate a certain sampling **s**equence. $a_t^s = 1$ means frame t will be picked. The relation between $\mathcal{V}_i$ and $\mathbf{a}^s$ is:

$$\mathcal{V}_i = \{\mathbf{x}_t | a_t^s = 1 \wedge \mathbf{x}_t \in v_i\}, \tag{9}$$

i.e., $\mathcal{V}_i$ are the picked frames from input video v_i following the action sequence $\mathbf{a}^s$.

We train PickNet by using REINFORCE algorithm, which is based on the observation that the gradient of a non-differentiable expected reward can be computed as follows:

$$\nabla_\theta L_R(\theta) = -\mathbb{E}_{\mathbf{a}^s \sim p_\theta}\left[r(\mathbf{a}^s)\nabla_\theta \log p_\theta(\mathbf{a}^s)\right]. \tag{10}$$

Using the chain rule, the gradient can be rewritten as:

$$\nabla_\theta L_R(\theta) = \sum_{t=1}^{n} \frac{\partial L_R(\theta)}{\partial \mathbf{s}_t}\frac{\partial \mathbf{s}_t}{\partial \theta} = \sum_{t=1}^{n} -\mathbb{E}_{\mathbf{a}^s \sim p_\theta} r(\mathbf{a}^s)(p_\theta(a_t^s) - \mathbf{1}_{a_t^s})\frac{\partial \mathbf{s}_t}{\partial \theta}, \tag{11}$$

where $\mathbf{s}_t$ is the input to the softmax function. In practice, the gradient can be approximated using a single Monte-Carlo sample $\mathbf{a}^s = (a_1^s, a_2^s, \ldots, a_n^s)$ from p_θ:

$$\nabla_\theta L_R(\theta) \approx -\sum_{t=1}^{n} r(\mathbf{a}^s)(p_\theta(a_t^s) - \mathbf{1}_{a_t^s})\frac{\partial \mathbf{s}_t}{\partial \theta}. \tag{12}$$

When using REINFORCE to train the policy network, we need to estimate a baseline reward b to diminish the variance of gradients. Here, the *self-critical* [26] strategy is used to estimate b. In brief, the reward obtained by current model under inferencing used at test stage, denoted as $r(\hat{\mathbf{a}})$, is treated as the baseline reward. Therefore, the final gradient expression is:

$$\nabla_\theta L_R(\theta) \approx -(r(\mathbf{a}^s) - r(\hat{\mathbf{a}}))\sum_{t=1}^{n}(p_\theta(a_t^s) - \mathbf{1}_{a_t^s})\frac{\partial \mathbf{s}_t}{\partial \theta}. \tag{13}$$

Adaptation stage. After the first two stages, the encoder-decoder and PickNet are well pretrained, but there exists a gap between them because the encoder-decoder use the full video frames as input while PickNet just selects a portion of frames. So we need a joint training stage to integrate this two parts together. However, the pick action is not differentiable, so the gradients introduced by cross-entropy loss can not flow into PickNet. Hence, we follow the approximate joint training scheme. In each iteration, the forward pass generates frame picks which are treated just like fixed picks when training the encoder-decoder, and the backward propagation and REINFORCE updates are performed as usual. It acts like performing dropout in time sequence, which can improve the versatility of the encoder-decoder.

4 Experimental Setup

4.1 Datasets

We evaluate our model on two widely used video captioning benchmark datasets: the Microsoft Video Description (MSVD) [4] and the MSR Video-to-Text (MSR-VTT) [38].

Microsoft Video Description (MSVD). The Microsoft Video Description is also known as YoutubeClips. It contains 1,970 Youtube video clips, each labeled with around 40 English descriptions collected by Amazon Mechanical Turks. As done in previous works [34], we split the dataset into three parts: the first 1,200 videos for training, then the followed 100 videos for validation and the remaining 670 videos for test. This dataset mainly contains short video clips with a single action, and the average duration is about 9 seconds. So it is very suitable to use only a portion of frames to represent the full video.

MSR Video-to-Text (MSR-VTT). The MSR Video-to-Text is a large-scale benchmark for video captioning. It provides 10,000 video clips, and each video is annotated with 20 English descriptions and category tag. Thus, there are 200,000 video-caption pairs in total. This dataset is collected from a commercial video search engine and so far it covers the most comprehensive categories and diverse visual contents. Following the original paper, we split the dataset in contiguous groups of videos by index number: 6,513 for training, 497 for validation and 2,990 for test.

4.2 Metrics

We employ four popular metrics for evaluation: BLEU [24], ROUGE$_L$ [19], METEOR [1] and CIDEr. As done in previous video captioning works, we use METEOR and CIDEr as the main comparison metrics. In addition, Microsoft COCO evaluation server has implemented these metrics and released evaluation functions[1], so we directly call such evaluation functions to test the performance of video captioning. Also, the CIDEr reward is computed by these functions.

4.3 Video preprocessing

First, we sample equally-spaced 30 frames for every video, and resize them into 224×224 resolution. Then the images are encoded with the final convolutional layer of ResNet152 [10], which results in a set of 2,048-dimensional vectors. Most video captioning models use motion features to improve performance. However, we **only use the appearance features** in our model, because extracting motion features is very time-consuming, which deviates from our purpose that cutting down the computation cost for video captioning, and the appearance feature is enough to represent video content when the redundant or noisy frames are filtered by our PickNet.

[1] https://github.com/tylin/coco-caption

4.4 Text preprocessing

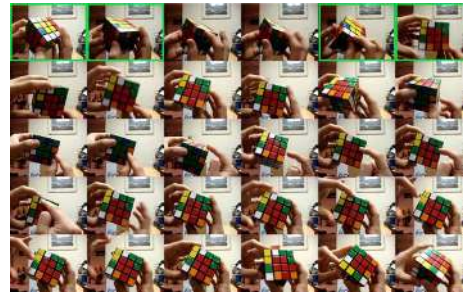
We tokenize the labeled sentences by converting all words to lowercases and then utilizing the word_tokenize function from NLTK toolbox to split sentences into words and remove punctuation. Then, the word with frequency less than 3 is removed. As a result, we obtain the vocabulary with 5,491 words from MSVD and 13,064 words from MSR-VTT. For each dataset, we use the one-hot vector (1-of-N encoding, where N is the size of vocabulary) to represent each word.

4.5 Implementation details

We use the validation set to tune some hyperparameters of our framework. The learning rates for three training stages are set to 3×10^{-4}, 3×10^{-4} and 1×10^{-4}, respectively. The training batchsize is 128 for MSVD and 256 for MSR-VTT, while each stage is trained up to 50 epoches and the best model is used to initialize the next stage. The minimum value of maximum picked frames τ is set to 7, and the penalty R^- is -1. To regularize the training and avoid over-fitting, we apply the well known regularization technique Dropout with retain probability 0.5 on the input and output of the encoding LSTMs and decoding GRUs. Embeddings for video features and words have size 512, while the sizes of all recurrent hidden states are empirically set to 1,024. For PickNet, the size of glance is 56×56, and the size of hidden layer is 1,024. The Adam [15] optimizer is used to update all the parameters.

5 Results and Discussion

Ours: a cat is playing with a dog
GT: a dog is playing with a cat



Ours: a person is solving a rubik's cube
GT: person playing with toy

Fig. 6: Example results on MSVD (left) and MSR-VTT (right). The green boxes indicate picked frames. (Best viewed in color and zoom-in. Frames are organized from left to right, then top to bottom in temporal order.)

Figure 6 gives some example results on the test sets of two datasets. As it can be seen, our PickNet can select informative frames, so the rest of our model can use these selected frames to generate reasonable descriptions. In short, two characteristics of picked frames can be found. The first characteristic is that the picked frames are concise

Model	BLEU4	ROUGE-L	METEOR	CIDEr	Time
Previous Work					
LSTM-E [23]	45.3	-	31.0	-	5x
p-RNN [44]	49.9	-	32.6	65.8	5x
HRNE [22]	43.8	-	33.1	-	33x
BA [2]	42.5	-	32.4	63.5	12x
Baseline Models					
Full	44.8	68.5	31.6	69.4	5x
Random	35.6	64.5	28.4	49.2	2.5x
k-means (k=6)	45.2	68.5	32.4	70.9	1x
Hecate [31]	43.2	67.4	31.7	68.8	1x
Our Models					
PickNet (V)	46.3	69.3	32.3	75.1	1x
PickNet (L)	49.9	69.3	32.9	74.7	1x
PickNet (V+L)	**52.3**	**69.6**	**33.3**	**76.5**	1x

Table 1: Experiment results on MSVD. All values are reported as percentage(%). L denotes using language reward and V denotes using visual diversity reward. k is set to the average number of picks $\bar{N_p}$ on MSVD. ($\bar{N_p} \approx 6$)

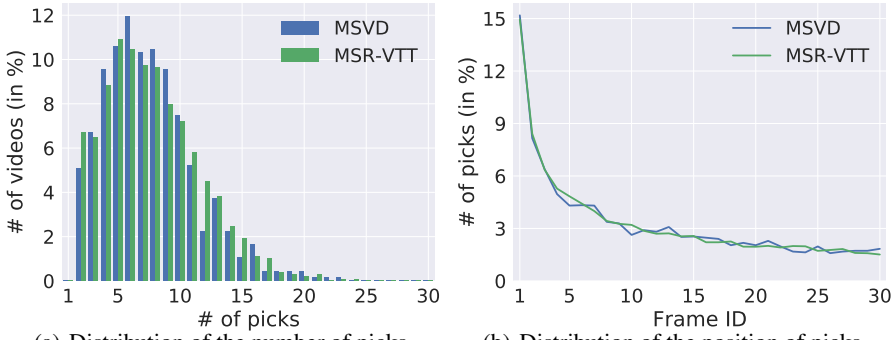
Model	BLEU4	ROUGE-L	METEOR	CIDEr	Time
Previous Work					
ruc-uva [7]	38.7	58.7	26.9	45.9	4.5x
Aalto [28]	39.8	59.8	26.9	45.7	4.5x
DenseVidCap [27]	41.4	61.1	28.3	48.9	10.5x
MS-RNN [30]	39.8	59.3	26.1	40.9	10x
Baseline Models					
Full	36.8	59.0	26.7	41.2	3.8x
Random	31.3	55.7	25.2	32.6	1.9x
k-means (k=8)	37.8	59.1	26.9	41.4	1x
Hecate [31]	37.3	59.1	26.6	40.8	1x
Our Models					
PickNet (V)	36.9	58.9	26.8	40.4	1x
PickNet (L)	37.3	58.9	27.0	41.9	1x
PickNet (V+L)	39.4	59.7	27.3	42.3	1x
PickNet (V+L+C)	41.3	59.8	27.7	44.1	1x

Table 2: Experiment results on MSR-VTT. All values are reported as percentage(%). C denotes using the provided category information. k is set to the average number of picks $\bar{N_p}$ on MSR-VTT. ($\bar{N_p} \approx 8$)

and highly related to the generated descriptions, and the second one is that the adjacent frames may be picked to represent action. In order to demonstrate the effectiveness of our framework, we compare our approach with some state-of-the-art methods on the two datasets, and analyze the learned picks of PickNet in consequent sections.

5.1 Comparison with the state-of-the-arts

We compare our approach on MSVD with four state-of-the-art approaches for video captioning: LSTM-E [23], p-RNN [44], HRNE [22] and BA [2]. LSTM-E uses a visual-semantic embedding to generate better captions. p-RNN use both temporal and spatial attention. BA uses a hierarchical encoder while HRNE uses a hierarchical decoder to describe videos. All of these methods use motion features (C3D or optical flow) and extract visual features frame by frame. Besides, we report the performance of our baseline models, which include using all the sampled frames, and using some straightforward picking strategies. In order to compare our PickNet with general picking policies, we conduct trials that pick frames by randomly selecting and k-means clustering, respectively. Specially, to compare with video summarization methods, we choose Hecate [31] to produce frame level summarization and use it to generate captions. For analyzing the effect of different rewards, we conduct some ablation studies on them. As it can be noticed in Table 1, our method improves plain techniques and achieves the state-of-the-art performance on MSVD. This result outperforms the most recent state-of-the-art method by a considerable margin of $\frac{76.5-65.8}{65.8} \approx 16.3\%$ on the CIDEr metric. Further, we try to compare the time efficiency among these approaches. However, most of state-of-the-art methods do not release executable codes, so the accurate performance may not be available. Instead, we estimate the running time by the complexity of visual feature extractors and the number of processed frames. Thanks to the PickNet, our captioning model is $5\sim33$ times faster than other methods.



(a) Distribution of the number of picks. (b) Distribution of the position of picks.

Fig. 7: Statistics on the behavior of our PickNet.

On MSR-VTT, we compare four state-of-the-art approaches: ruc-uva [7], Aalto [28], DenseVidCap [27] and MS-RNN [30]. ruc-uva incorporates the encoder-decoder with two new stages called early embedding which enriches input with tag embeddings, and late reranking which re-scores generated sentences in terms of their relevance to a specific video. Aalto first trains two models which are separately based on attribute and motion features, and then trains a evaluator to choose the best candidate generated by the two captioning model. DenseVidCap generates multiple sentences with regard to video segments and uses a winner-take-all scheme to produce the final description. MS-RNN uses a multi-modal LSTM to model the uncertainty in videos to generate diverse captions. Compared with these methods, our method can be simply trained in end-to-end fashion, and does not rely upon any auxiliary information. The performance of these approaches and that of our solution is reported in Table 2. We observe that our approach is able to achieve competitive result even **without utilizing attribute information**, while other methods take advantage of attributes and auxiliary information sources. Also, our model is the **fastest** among the compared methods. For fairly demonstrating the effectiveness of our method, we embed the provided category information into our language model, and better accuracy can be achieved (PickNet (V+L+C) in Table 2). It is also worth noting that the PickNet can be easily integrated with the compared methods, since none of them incorporated with frame selection algorithm. For example, DenseVidCap generates region-sequence candidates based on equally sampled frames. It can alternatively utilize PickNet to reduce the time for generating candidates by cutting down the number of selected frames.

5.2 Analysis of learned picks

We collect statistics on the properties of our PickNet. Figure 7 shows the distributions of the number and position of picked frames on the test sets of MSVD and MSR-VTT. As observed in Figure 7(a), in the vast majority of the videos, less than 10 frames are picked. It implies that in most case only $\frac{10}{30} \approx 33.3\%$ frames are necessary to be encoded for captioning videos, which can largely reduce the computation cost. Specifically, the average number of picks is around 6 for MSVD and 8 for MSR-VTT. Looking at the distributions of position of picks in Figure 7(b), we observe a pattern

of *power law distribution*, *i.e.*, the probability of picking a frame is reduced as time goes by. It is reasonable since most videos are single-shot and the anterior frames are sufficient to represent the whole video.

5.3 Captioning for streaming video

a cat is playing → a rabbit is playing → a rabbit is being petted
→ a person is petting a rabbit $\times 3$

Fig. 8: An example of online video captioning.

One of the advantage of our method is that it can be applied to streaming video. Different from offline video captioning, captioning for streaming video requires the model to tackle with unbounded video and generate descriptions immediately when the visual information has changed, which meets the demand of practical applications. For this online setting, we first sample frames at 1fps, and then sequentially feed the sampled frames to PickNet. If certain frame is picked, the pretrained CNN will be used to extract visual features of this frame. After that, the encoder will receive this feature, and produce a new encoded representation of the video stream up to current time. Finally, the decoder will generate a description based on the encoded representation. Figure 8 demonstrates an example of online video captioning with the picked frames and corresponding descriptions. As it is shown, the descriptions will be more appropriate and more determined as the informative frames are picked.

6 Conclusion

In this work, we design a plug-and-play reinforcement-learning-based PickNet to select informative frames for the task of video captioning, which achieves promising performance on effectiveness, efficiency and flexibility on popular benchmarks. This architecture can largely cut down the usage of convolution operations by picking only $6\sim8$ frames for a video clip, while other video analysis methods usually require more than 40 frames. This property makes our method more applicable for real-world video processing. The proposed PickNet has a good flexibility and could be potentially employed to other video-related applications, such as video classification and action detection, which will be further addressed in our future work.

7 Acknowledgment

This work was supported in part by National Natural Science Foundation of China: 61672497, 61332016, 61620106009, 61650202 and U1636214, in part by National Basic Research Program of China (973 Program): 2015CB351802 and in part by Key Research Program of Frontier Sciences of CAS: QYZDJ-SSW-SYS013.

References

1. Banerjee, S., Lavie, A.: Meteor: an automatic metric for mt evaluation with improved correlation with human judgments. In: ACL. pp. 65–72 (2005)
2. Baraldi, L., Grana, C., Cucchiara, R.: Hierarchical boundary-aware neural encoder for video captioning. In: CVPR. pp. 3185–3194 (2017)
3. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: NIPS. pp. 1171–1179 (2015)
4. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL. pp. 190–200 (2011)
5. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: EMNLP. pp. 1724–1734 (2014)
6. Cromwell, H.C., Mears, R.P., Wan, L., Boutros, N.N.: Sensory gating: A translational effort from basic to clinical science. Clinical EEG and Neuroscience **39**(2), 69–72 (2008)
7. Dong, J., Li, X., Lan, W., Huo, Y., Snoek, C.G.M.: Early embedding and late reranking for video captioning. In: ACM Multimedia. pp. 1082–1086 (2016)
8. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollar, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., Zweig, G.: From captions to visual concepts and back. In: CVPR. pp. 1473–1482 (2015)
9. Farhadi, A., Hejrati, M., Sadeghi, M.A., Young, P., Rashtchian, C., Hockenmaier, J., Forsyth, D.: Every picture tells a story: Generating sentences from images. In: ECCV. pp. 15–29 (2010)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Hochreiter, S., Schmidhuber, J.J.J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
12. Hori, C., Hori, T., Lee, T.Y., Sumi, K., Hershey, J.R., Marks, T.K.: Attention-based multimodal fusion for video description. In: ICCV. pp. 4203–4212 (2017)
13. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on pattern analysis and machine intelligence **20**(11), 1254–1259 (1998)
14. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: CVPR. pp. 4565–4574 (2016)
15. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: ICLR (2015)
16. Kojima, A., Tamura, T., Fukunaga, K.: Natural language description of human activities from video images based on concept hierarchy of actions. IJCV **50**(2), 171–184 (2002)
17. Krause, J., Johnson, J., Krishna, R., Fei-Fei, L.: A hierarchical approach for generating descriptive image paragraphs. In: CVPR. pp. 3337–3345 (2017)
18. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: CVPR. pp. 1601–1608 (2011)
19. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: ACL (2004)
20. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical co-attention for visual question answering. In: NIPS. pp. 289–297 (2016)
21. Mnih, V., Heess, N., Graves, A., Kavukcuoglu, K.: Recurrent models of visual attention. In: NIPS. pp. 2204–2212 (2014)
22. Pan, P., Xu, Z., Yang, Y., Wu, F., Zhuang, Y.: Hierarchical recurrent neural encoder for video representation with application to captioning. In: CVPR. pp. 1029–1038 (2016)
23. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. In: CVPR. pp. 4594–4602 (2016)

24. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318 (2002)
25. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. In: ICLR (2016)
26. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: CVPR. pp. 1179–1195 (2017)
27. Shen, Z., Li, J., Su, Z., Li, M., Chen, Y., Jiang, Y.G., Xue, X.: Weakly supervised dense video captioning. In: CVPR. pp. 5159–5167 (2017)
28. Shetty, R., Laaksonen, J.: Frame-and segment-level features and candidate pool evaluation for video caption generation. In: ACM Multimedia. pp. 1073–1076 (2016)
29. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS. pp. 568–576 (2014)
30. Song, J., Guo, Y., Gao, L., Li, X., Hanjalic, A., Shen, H.T.: From deterministic to generative: Multi-modal stochastic rnns for video captioning. arXiv (2017)
31. Song, Y., Redi, M., Vallmitjana, J., Jaimes, A.: To click or not to click: Automatic selection of beautiful thumbnails from videos. In: CIKM. pp. 659–668 (2016)
32. Song, Y., Vallmitjana, J., Stent, A., Jaimes, A.: Tvsum: Summarizing web videos using titles. In: CVPR. pp. 5179–5187 (2015)
33. Vedantam, R., Zitnick, C.L., Parikh, D.: Cider: consensus-based image description evaluation. In: CVPR. pp. 4566–4575 (2015)
34. Venugopalan, S., Rohrbach, M., Darrell, T., Donahue, J., Saenko, K., Mooney, R.: Sequence to sequence - video to text. In: ICCV. pp. 4534–4542 (2015)
35. Wang, B., Ma, L., Zhang, W., Liu, W.: Reconstruction network for video captioning. In: CVPR. pp. 7622–7631 (2018)
36. Wang, J., Jiang, W., Ma, L., Liu, W., Xu, Y.: Bidirectional attentive fusion with context gating for dense video captioning. In: CVPR. pp. 7190–7198 (2018)
37. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)
38. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR. pp. 5288–5296 (2016)
39. Xu, K., Ba, J.L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: neural image caption generation with visual attention. In: ICML. pp. 2048–2057 (2015)
40. Yang, Y., Teo, C.L., Daumé III, H., Aloimonos, Y.: Corpus-guided sentence generation of natural images. In: EMNLP. pp. 444–454 (2011)
41. Yao, L., Cho, K., Ballas, N., Paí, C., Courville, A.: Describing videos by exploiting temporal structure. In: ICCV. pp. 4507–4515 (2015)
42. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. In: CVPR. pp. 2678–2687 (2016)
43. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: CVPR. pp. 4651–4659 (2016)
44. Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W.: Video paragraph captioning using hierarchical recurrent neural networks. In: CVPR. pp. 4584–4593 (2016)
45. Yu, Y., Choi, J., Kim, Y., Yoo, K., Lee, S.H., Kim, G., Kim, G.: Supervising neural attention models for video captioning by human gaze data. In: CVPR. pp. 6119–6127 (2017)
46. Zeng, K., Chen, T., Niebles, J.C., Sun, M.: Title generation for user generated videos. In: ECCV. pp. 609–625 (2016)
47. Zhao, B., Xing, E.P.: Quasi real-time summarization for consumer videos. In: CVPR. pp. 2513–2520 (2014)
48. Zheng, H., Fu, J., Mei, T.: Look closer to see better : Recurrent attention convolutional neural network for fine-grained image recognition. In: CVPR. pp. 4476–4484 (2017)