Temporal Deformable Convolutional Encoder-Decoder Networks for Video Captioning[*]

Jingwen Chen,[1] Yingwei Pan,[2] Yehao Li,[1] Ting Yao,[2] Hongyang Chao,[1,3] Tao Mei[2]
[1]Sun Yat-sen University, Guangzhou, China
[2]JD AI Research, Beijing, China
[3]The Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education, Guangzhou, China
{chenjingwen.sysu, panyw.ustc, yehaoli.sysu, tingyao.ustc}@gmail.com, isschhy@mail.sysu.edu.cn, tmei@live.com

Abstract

It is well believed that video captioning is a fundamental but challenging task in both computer vision and artificial intelligence fields. The prevalent approach is to map an input video to a variable-length output sentence in a sequence to sequence manner via Recurrent Neural Network (RNN). Nevertheless, the training of RNN still suffers to some degree from vanishing/exploding gradient problem, making the optimization difficult. Moreover, the inherently recurrent dependency in RNN prevents parallelization within a sequence during training and therefore limits the computations. In this paper, we present a novel design — Temporal Deformable Convolutional Encoder-Decoder Networks (dubbed as TDConvED) that fully employ convolutions in both encoder and decoder networks for video captioning. Technically, we exploit convolutional block structures that compute intermediate states of a fixed number of inputs and stack several blocks to capture long-term relationships. The structure in encoder is further equipped with temporal deformable convolution to enable free-form deformation of temporal sampling. Our model also capitalizes on temporal attention mechanism for sentence generation. Extensive experiments are conducted on both MSVD and MSR-VTT video captioning datasets, and superior results are reported when comparing to conventional RNN-based encoder-decoder techniques. More remarkably, TDConvED increases CIDEr-D performance from 58.8% to 67.2% on MSVD.

Introduction

The recent advances in deep neural networks have convincingly demonstrated high capability in learning vision models particularly for recognition. The achievements make a further step towards the ultimate goal of video understanding, which is to automatically describe video content with a complete and natural sentence or referred to as video captioning problem. The typical solutions (Pan et al. 2016; Venugopalan et al. 2015a; 2015b; Yao et al. 2015; Yu et al. 2016) of video captioning are inspired by machine translation and equivalent to translating a video to a text. As illustrated in Figure 1 (a), the basic idea is to perform sequence to sequence learning, where a Convolutional Neural Network (CNN) or Recurrent Neural Network is usually exploited to

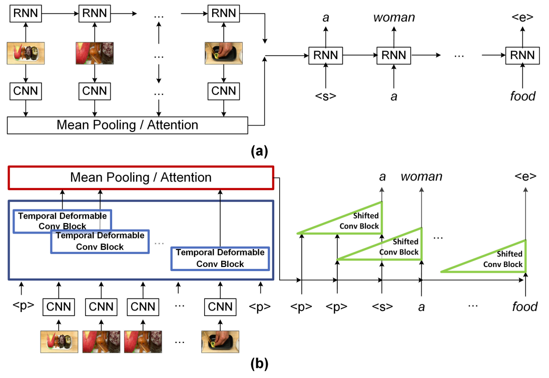


Figure 1: An illustration of (a) CNN or RNN plus RNN architecture and (b) our Temporal Deformable Convolutional Encoder-Decoder Networks, for video captioning.

encode a video and a decoder of RNN is utilized to generate the sentence, one word at each time step till the end. Regardless of different versions of CNN or RNN plus RNN video captioning framework, a common practice is the use of sequential RNN, which may result in two limitations. First, RNN powered by Long Short-Term Memory (LSTM) alleviates the issue of vanishing or exploding gradients to some extent, nevertheless the problem still exists. Moreover, the utilization of RNN on a long path between the start and end of a sequence will easily forget the long-range information. Second, the training process of RNN is inherently sequential due to the recurrent relation in time and thus prevents parallel computations in a sequence.

We propose to mitigate the aforementioned two issues via convolutions in an encoder-decoder architecture. Compared to the chain structure in RNN, feed-forward convolutions produce representations of fixed-length inputs and could easily capture long-term dependencies by stacking several layers on the top of each other. Unlike RNN in which the inputs are fed through a variable number of non-linearities and the number depends on the input length, the operation of convolution fixes the number of non-linearities and thus eases the optimization. Furthermore, feed-forward convolutions are without any recurrent functions and do not rely on the computation of the previous time step. Therefore, com-

[*]This work was performed at JD AI Research.

putations over all the input elements could be fully parallelized during training.

By consolidating the idea of leveraging convolutions in sequence-to-sequence learning for video captioning, we present a novel Temporal Deformable Convolutional Encoder-Decoder Networks (TDConvED) architecture, as conceptually shown in Figure 1 (b). Given a video, a standard CNN is utilized to extract visual features of sampled frames or clips. Then, the features of frames/clips are fed into convolutional block to produce intermediate states of each frame/clip with contexts. The context size is enlarged by stacking several convolutional blocks. Please also note that we endow the convolutional block in encoder with more power of devising temporal deformable convolutions by capturing dynamics in temporal extents of actions or scenes. Once all sampled frames/clips are encoded, the video-level representations are computed by mean pooling over all the intermediate states. Similarly, we employ feed-forward convolutional blocks that operate on the concatenation of video-level representations and the embeddings of past words to generate subsequent word, till the end token is predicted. In addition, sentence generation is further enhanced by additionally incorporating temporally attended video features. Since there are no recurrent connections and all ground-truth words are available at any time step, our TDConvED model could be trained in parallel.

Our main contribution is the proposal of the use of convolutions for sequence-to-sequence learning and eventually boosting video captioning. This also leads to the elegant view of how to design feed-forward convolutional structure in an encoder-decoder video captioning framework, and how to support parallelization over every element within a sequence, which are problems not yet fully understood.

Related Work

Video Captioning. The dominant direction in modern video captioning is sequence learning approaches (Pan et al. 2016; 2017; Chen et al. 2017; Venugopalan et al. 2015a; 2015b; Yao et al. 2015; Yu et al. 2016; Li et al. 2018) which utilize RNN-based architecture to generate novel sentences with flexible syntactical structures. For instance, Venugopalan *et al.* present a LSTM-based model to generate video descriptions with the mean pooled representation over all frames in (Venugopalan et al. 2015b). The framework is then extended by inputting both frames and optical flow images into an encoder-decoder LSTM in (Venugopalan et al. 2015a). Compared to mean pooling, Yao *et al.* propose to utilize the temporal attention mechanism to exploit temporal structure for video captioning (Yao et al. 2015). Later in (Zhao, Li, and Lu 2018), Zhao *et al.* design an object-aware tube feature for video captioning to enable attention on salient objects. Most recently, Hao *et al.* develop several deep fusion strategies to effectively integrate both visual and audio cues into sentence generation in (Hao, Zhang, and Guan 2018).

Sequence Learning with CNN. The most typical paradigm in sequence learning is RNN-based encoder-decoder structure which mainly capitalizes on RNN to model the probability of decoding an output given the previous outputs and all the inputs. Although the remarkable results have been observed on a number of sequential tasks (e.g., image/video captioning, machine translation), the inherent recursive characteristic inevitably limits the parallelization abilities and even raises vanishing/exploding gradient problems in the training stage. To tackle these barriers, there is an emerging trend of leveraging Convolutional Neural Network (CNN) for sequence learning in language modeling for NLP tasks (Meng et al. 2015; Bradbury et al. 2016; Kalchbrenner et al. 2016; Gehring et al. 2017). A convolutional encoder with a gated architecture has been designed in (Meng et al. 2015), which could pinpoint the parts of a source sentence that are relevant to the target word for machine translation. Recently, the first fully convolutional model for sequence learning is proposed in (Gehring et al. 2017) to design both encoder and decoder in the form of convolutions with CNN, which even outperforms strong recurrent models on machine translation task.

Summary. In short, our approach in this paper belongs to sequence learning method for video captioning. Unlike most of the aforementioned sequence learning models which mainly focus on generating sentence by solely depending on RNN-based decoder, our work contributes by exploiting a fully convolutional sequence learning architecture that relies on CNN-based encoder and decoder for video captioning. Moreover, we additionally explore the temporal deformable convolutions and temporal attention mechanism to extend and utilize temporal dynamics across frames/clips, and eventually enhance the quality of generated sentence.

Temporal Deformable Convolutional Encoder-Decoder Networks

Our Temporal Deformable Convolutional Encoder-Decoder Networks (TDConvED) architecture is devised to generate video descriptions by fully capitalizing on convolutional encoder and decoder. Specifically, a convolutional encoder is firstly leveraged to encode each frame/clip with the contextual information among multiple sampled frames/clips via temporal deformable convolution. The idea behind temporal deformable convolution is to augment the temporal samplings with additional offsets learnt from the inherent temporal dynamics of scenes/actions within frame/clip sequence. Next, conditioned on the video-level representation induced by the temporal deformable convolutional encoder (e.g., performing mean pooling over all encoded features of frames/clips), we employ a convolutional decoder to predict each output word by additionally exploiting the contextual relations across the previous output words. Furthermore, a temporal attention mechanism tailored to temporal deformable convolutional encoder-decoder structure is designed to further boost video captioning. An overview of TDConvED is shown in Figure 2.

Problem Formulation

Suppose a video V with N_v sampled frames/clips (uniform sampling) to be described by a textual sentence S, where $S = (w_1, w_2, \ldots, w_{N_s})$ is the word sequence consisting of N_s words. Let $\mathbf{v} = (v_1, v_2, \ldots, v_{N_v})$ denote the temporal sequence of frame/clip representations of the video V,

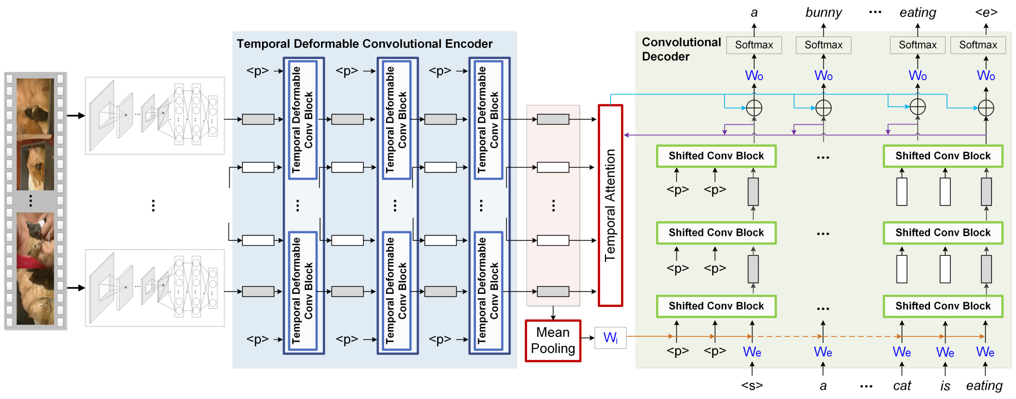


Figure 2: The overall architecture of our TDConvED that fully employs convolutions in both encoder and decoder networks for video captioning. A standard CNN is firstly utilized to extract visual features of sampled frames/clips. Then, the features of frames/clips are fed into a temporal deformable convolutional encoder with stacked temporal deformable convolutional blocks to produce intermediate states of each frame/clip with contexts. The video-level representations are computed by mean pooling over all the intermediate states. After that, a convolutional decoder with stacked shifted convolutional blocks operates on the concatenation of video-level representations and the embeddings of past words to generate the next word. In addition, a temporal attention mechanism tailored to such convolutional encoder-decoder structure is incorporated to boost video captioning.

where $v_i \in \mathbb{R}^{D_v}$ represents the D_v-dimensional feature of i-th frame/clip extracted by 2D/3D CNN.

Inspired by the recent successes of sequence to sequence solutions leveraged in image/video captioning (Venugopalan et al. 2015a; Yao et al. 2017a; 2017b; 2018), we aim to formulate our video captioning model in an encoder-decoder scheme. Instead of relying on RNN to compute intermediate encoder/decoder states, we novelly utilize a fully convolutional encoder-decoder architecture for sequence to sequence modeling in video captioning. Formally, a temporal deformable convolutional encoder is utilized to encode the input frame/clip sequence $\mathbf{v} = (v_1, v_2, \ldots, v_{N_v})$ into a set of context vectors $\mathbf{z} = (z_1, z_2, \ldots, z_{N_v})$, which are endowed with the contextual information among frames/clips sampled in a free-form temporal deformation along the input sequence. Next, we perform mean pooling over all the context vectors $\mathbf{z}$ to achieve the video-level representation, which is further injected into the convolutional decoder to generate target output sentence S by capturing the long-term contextual relations among input video and previous output words. Thus, the video sentence generation problem we exploit is formulated by minimizing the following energy loss:

$$E(\mathbf{v}, S) = -\log \Pr(S|\mathbf{v}), \tag{1}$$

which is the negative $\log$ probability of the correct textual sentence given the temporal sequence of input video. By applying chain rule to model the joint probability over the sequential words, the $\log$ probability of the sentence is measured by the sum of the $\log$ probabilities over each word, which can be written as

$$\log \Pr(S|\mathbf{v}) = \sum_{t=1}^{N_s} \log \Pr(w_t|\mathbf{v}, w_0, \ldots, w_{t-1}). \tag{2}$$

In contrast to the conventional RNN-based encoder-decoder architecture that models the above parametric distribution $\Pr(w_t|\mathbf{v}, w_0, \ldots, w_{t-1})$ with RNN, we employ convolutions to calculate it in a simple feed-forward manner at decoding stage. Since no recurrent dependency exists in the convolutional decoder and all grough-truth words are available at training, the joint probability $\Pr(S|\mathbf{v})$ can be calculated in parallel for all words. Derived from the idea of attention mechanism in sequence to sequence modeling (Yao et al. 2015), we also capitalize on temporal attention mechanism to focus on the frames/clips that are highly relevant to the output word for enriching the video-level representation. Such representation will be additionally injected into convolutional decoder to enhance video captioning.

Temporal Deformable Convolutional Encoder

An encoder is a module that takes the source sequence (i.e., frame/clip sequence of video) as input and produces intermediate states to encode the semantic content. Here we devise a temporal deformable convolutional block in the encoder of our TDConvED which applies temporal deformable convolution over the input sequence to capture the context across frames/clips sampled in a free-form temporal deformation, as depicted in Figure 3 (a). Such design of temporal deformable convolution improves conventional temporal convolution by capturing temporal dynamics on the natural basis of actions/scenes within the video. Meanwhile, the feed-forward convolutional structure in encoder enables the parallelization within the input sequence and allows fast computation. Furthermore, to exploit long-term dependency among input sequence at encoding stage, multiple temporal deformable convolutional blocks are stacked in encoder to integrate the contextual information from a larger number of temporal samplings in the input sequence.

Formally, consider the l-th temporal deformable convolutional block in encoder with the corresponding output sequence denoted as $\mathbf{p}^l = (p_1^l, p_2^l, \ldots, p_{N_v}^l)$, where $p_i^l \in \mathbb{R}^{D_r}$

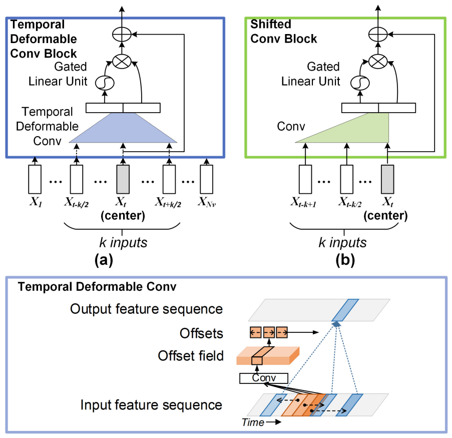


Figure 3: An illustration of (a) the temporal deformable conv block in encoder, (b) the shifted conv block in decoder, and (c) the temporal deformable convolution.

is the output of the temporal deformable convolution centered on the i-th frame/clip. Given the output sequence $\mathbf{p}^{l-1} = (p_1^{l-1}, p_2^{l-1}, \ldots, p_{N_v}^{l-1})$ of the $(l-1)$-th block, each output intermediate state p_i^l is achieved by feeding the subsequence of $\mathbf{p}^{l-1}$ into a temporal deformable convolution (kernel size: k) plus a non-linearity unit. Note that the temporal deformable convolution operates in a two-stage way, i.e., first measuring temporal offsets via an one-dimensional convolution for sampling frames/clips and then aggregating the features of sampled frames/clips, as depicted in Figure 3 (c). More specifically, let $X = (p_{i+r_1}^{l-1}, p_{i+r_2}^{l-1}, \ldots, p_{i+r_k}^{l-1})$ denote the subsequence of $\mathbf{p}^{l-1}$, where r_n is the n-th element of R and $R = \{-k/2, \ldots, 0, \ldots, k/2\}$. The one-dimensional convolution in the l-th temporal deformable convolutional block can be parameterized as the transformation matrix $W_f^l \in \mathbb{R}^{k \times kD_r}$ and the bias $b_f^l \in \mathbb{R}^k$, which takes the concatenation of k elements in X as input and produces a set of offsets $\Delta r^i = \{\Delta r_n^i\}_{n=1}^k \in \mathbb{R}^k$:

$$\Delta r^i = W_f^l[p_{i+r_1}^{l-1}, p_{i+r_2}^{l-1}, \ldots, p_{i+r_k}^{l-1}] + b_f^l, \tag{3}$$

where the n-th element Δr_n^i in Δr^i denotes the measured temporal offset for the n-th sample in the subsequence X. Next, we achieve the output of temporal deformable convolution by augmenting samples with temporal offsets via another one-dimensional convolution:

$$o_i^l = W_d^l[p_{i+r_1+\Delta r_1^i}^{l-1}, p_{i+r_2+\Delta r_2^i}^{l-1}, \ldots, p_{i+r_k+\Delta r_k^i}^{l-1}] + b_d^l, \tag{4}$$

where $W_d^l \in \mathbb{R}^{2D_r \times kD_r}$ is the transformation matrix in the one-dimensional convolution and $b_d^l \in \mathbb{R}^{2D_r}$ is the bias. As the temporal offset Δr_n^i is typically fractional, $p_{i+r_n+\Delta r_n^i}^{l-1}$ in Eq.(4) can be calculated via temporal linear interpolation:

$$p_{i+r_n+\Delta r_n^i}^{l-1} = \sum_s B(s, i + r_n + \Delta r_n^i) p_s^{l-1}, \tag{5}$$

where $i + r_n + \Delta r_n^i$ denotes an arbitrary position, s enumerates all integral positions in the input sequence $\mathbf{p}^{l-1}$, and $B(a, b) = max(0, 1 - |a - b|)$.

In addition, we adopt a gated linear unit (GLU) (Dauphin et al. 2017) as the non-linearity unit to ease gradient propagation. Thus given the output of temporal deformable convolution $o_i^l \in \mathbb{R}^{2D_r}$ which has twice the dimension of the input element, a GLU is applied over $o_i^l = [A, B]$ through a simple gating mechanism:

$$g(o_i^l) = A \otimes \sigma(B), \tag{6}$$

where $A, B \in \mathbb{R}^{D_r}$ and $\otimes$ is the point-wise multiplication. $\sigma(B)$ represents a gate unit that controls which elements of A are more relevant to the current context. Moreover, the residual connections (He et al. 2016) from input of the temporal deformable convolution block to the output of the block are added to enable deeper networks. Hence, the final output of the l-th temporal deformable convolutional block is measured as

$$p_i^l = g(o_i^l) + p_i^{l-1}. \tag{7}$$

To ensure the match between the length of output sequence of temporal deformable convolutional block and the input length, both the left and right sides of the input are padded with $k/2$ zero vectors. By stacking several temporal deformable convolutional blocks over the input frame/clip sequence, we obtain the final sequence of context vectors $\mathbf{z} = (z_1, z_2, \ldots, z_{N_v})$, where $z_i \in \mathbb{R}^{D_r}$ represents the contextually encoded feature of the i-th frame/clip.

Convolutional Decoder

The most typical way to implement the decoder for sequence learning is to adopt sequential RNN to produce the output sequence conditioned on the representations of the input sequence induced by the encoder. However, the inherent dependency in RNN inevitably limits the parallelization abilities of RNN-based decoder and even raises vanishing/exploding gradient problems for training. To alleviate the challenges in decoder of sequence learning, we devise a fully convolutional decoder by stacking several shifted convolutional blocks to capture long-term context across the encoded context vectors of input video and output words. Here, the center position of the convolution in shifted convolutional block is shifted with respect to the center of normal convolutions, as shown in Figure 3 (b).

In particular, given the encoded sequence of context vectors $\mathbf{z} = (z_1, z_2, \ldots, z_{N_v})$ for input video, we first perform mean pooling over all context vectors and achieve the video-level representation $\tilde{\mathbf{z}} = \frac{1}{N_v}\sum_{i=1}^{N_v} z_i$. Since no memory cells or autoregressive path exists in the convolutional decoder, the video-level representation $\tilde{\mathbf{z}}$ is transformed with a linear mapping W_i and concatenated with the embeddings of the input word at each time step, which will be set as the input of the first shifted convolutional block of the decoder. The transformation for addition of word embeddings and position embeddings is denoted as W_e. Consider the l-th shifted convolutional block in decoder with the corresponding output sequence $\mathbf{q}^l = (q_0^l, q_1^l, \ldots, q_{N_s}^l)$, where $q_t^l \in \mathbb{R}^{D_f}$ denotes the output of the convolution at time step t. More specifically, by taking the output sequence of the $(l-1)$-th shifted convolutional block as input, every subsequence of $\mathbf{q}^{l-1}$ is fed into an one-dimensional convolution (kernel size: k) plus a GLU and residual connections, resulting in the output intermediate state q_t^l at time step t. Different from the basic convolution, the corresponding subsequence of $\mathbf{q}^{l-1}$ is $X_q = (q_{t-k+1}^{l-1}, q_{t-k+2}^{l-1}, \ldots, q_t^{l-1})$ since no future information is available to decoder at decoding stage. Hence, the operation in each shifted convolutional block is given by

$$q_t^l = g(W_l^q[q_{t-k+1}^{l-1}, q_{t-k+2}^{l-1}, \ldots, q_t^{l-1}] + b_l^q) + q_t^{l-1}, \quad (8)$$

where $W_l^q \in \mathbb{R}^{2D_f \times kD_f}$ and $b_l^q \in \mathbb{R}^{2D_f}$ denote the transformation matrix and bias of the convolutional kernel in the l-th shifted convolutional block, respectively. Note that for each shifted convolutional block, only the left side of the input sequence is padded with $k-1$ zero vectors for matching the length between input and output sequence. Accordingly, by leveraging the convolutional decoder with stacked shifted convolutional blocks, we obtain the final output sequence $\mathbf{h} = (h_1, h_2, \ldots, h_{N_s})$ and each element $h_t \in \mathbb{R}^{D_f}$ denotes the output intermediate state at time step t.

Temporal Attention Mechanism

In many cases, the output word at decoding stage only relates to some frames/clips of the input video. As a result, utilizing one encoder to compress the whole video into a global feature (e.g., the aforementioned mean pooled video-level representation $\tilde{\mathbf{z}}$) would collapse the natural temporal structure within video and bring in the noises from irrelevant frames/clips, resulting in sub-optimal results. To dynamically pinpoint the frames/clips that are highly relevant to the output word and further incorporate the contributions of different frames/clips into producing video-level representation for predicting target word, a temporal attention mechanism is employed over the encoded sequence of context vectors $\mathbf{z} = (z_1, z_2, \ldots, z_{N_v})$ to boost video captioning. Specifically, at each time step t, the attention mechanism first generates a normalized attention distribution over all the encoded context vectors $\mathbf{z}$ depending on the current output intermediate state h_t of the convolutional decoder as

$$a_i^t = W_a \left[\tanh\left(W_z z_i + W_h h_t + b_a\right)\right], \lambda^t = softmax\left(a^t\right), \quad (9)$$

where a_i^t is the i-th element of a^t, $W_a \in \mathbb{R}^{1 \times D_a}$, $W_z \in \mathbb{R}^{D_a \times D_r}$ and $W_h \in \mathbb{R}^{D_a \times D_f}$ are transformation matrices. $\lambda^t \in \mathbb{R}^{N_v}$ denotes the normalized attention distribution and its i-th element λ_i^t is the attention probability of context vector z_i of the i-th frame/clip. Based on the attention distribution, we calculate the attended video-level feature $\hat{\mathbf{z}}_t$ by aggregating all the context vectors weighted by attention:

$$\hat{\mathbf{z}}_t = \sum_{i=1}^{N_v} \lambda_i^t z_i. \quad (10)$$

The attended video-level feature $\hat{\mathbf{z}}_t$ is further transformed with a linear mapping and combined with the output intermediate state h_t of convolutional decoder, which is employed to predict the next word through a softmax layer.

Inference

Similar to RNN based model, the inference of TDConvED is performed sequentially by generating one word at each time step. At the beginning of inference, the start sign word $< s >$ and the encoded sequence $\mathbf{z} = (z_1, z_2, \ldots, z_{N_v})$ of input video are jointly fed into the convolutional decoder to predict the output word distribution. The word with the maximum probability is popped out as the input word to the convolutional decoder at the next time step. This process continues until the end sign word $< e >$ is emitted or the pre-defined maximum sentence length is reached.

Experiments

We evaluate and compare our TDConvED with conventional RNN-based techniques by conducting video captioning task on two benchmarks, i.e., Microsoft Research Video Description Corpus (MSVD) (Chen and Dolan 2011) and Microsoft Research Video to Text (MSR-VTT) (Xu et al. 2016).

Datasets and Experimental Settings

MSVD. The MSVD dataset is the most popular video captioning benchmark and contains 1970 short videos clips from YouTube. Following the standard settings in previous works (Pan et al. 2016; Yao et al. 2015), we take 1200 videos for training, 100 for validation and 670 for testing.

MSR-VTT. MSR-VTT is a recently released large-scale benchmark for video captioning and consists of 10,000 web video clips from 20 well-defined categories. Following the official split, we utilize 6513, 497, and 2990 video clips for training, validation and testing, respectively.

Features and Parameter Settings. We uniformly sample 25 frames/clips for each video and each word in the sentence is represented as "one-hot" vector (binary index vector in a vocabulary). For visual representations, we take the output of 4,096-way fc7 layer from VGG or 2,048-way pool5 from ResNet (He et al. 2016) pre-trained on ImageNet (Russakovsky et al. 2015), and 4,096-way fc6 layer from C3D (Tran et al. 2015) pre-trained on Sports-1M (Karpathy et al. 2014) as frame/clip representations. In training phase, we add a start token $< s >$ to indicate the starting of sentence, an end token $< e >$ to denote the end of each sentence, and a padding token $< p >$ to keep the length of all textual sequences the same. For sentence generation in inference stage, we adopt the beam search and set the beam size as 5. The kernel size k of convolutions in encoder and decoder is set as 3. The convolutional encoder/decoder in our TDConvED consists of 2 stacked temporal deformable/shifted convolutional blocks. Both the dimensions of intermediate states in encoder and decoder, i.e., D_r and D_f, are set as 512. The dimension of the hidden layer for measuring attention distribution D_a is set as 512. The whole model is trained by Adam (Kingma and Ba 2015) optimizer. We set the initial learning rate as 10^{-3} and the mini-batch size as 64. The maximum training iteration is set as 30 epoches.

Evaluation Metrics. We adopt three common metrics: BLEU@4 (Papineni et al. 2002), METEOR (Banerjee and Lavie 2005), and CIDEr-D (Vedantam, Zitnick, and Parikh

Table 1: Performances of TDConvED and other RNN-based approaches on MSVD, where B@4, M and C are short for BLEU@4, METEOR and CIDEr-D. All values are reported as percentage (%). The short name in the brackets indicates the features, where G, V, C, O, R and M denotes GoogleNet, VGGNet, C3D, Optical flow, ResNet and motion feature learnt by 3D CNN on hand-crafted descriptors, respectively.

Model	M	C	B@4
MP-LSTM (V)	29.2	53.3	37.0
MP-LSTM (V+C)	29.7	55.1	39.4
MP-LSTM (R)	32.5	71.0	50.4
S2VT (V+O)	29.8	-	-
S2VT (V+C)	30.0	58.8	42.1
TA (G+M)	29.6	51.7	41.9
TA (V+C)	29.9	56.1	40.9
TA (R)	33.3	72.0	51.3
LSTM-E (V+C)	31.0	-	45.3
GRU-RCN (G)	31.6	68.0	43.3
BAE (R+C)	32.4	63.5	42.5
p-RNN (V+C)	32.6	65.8	49.9
MAMRNN (G)	32.9	-	-
PickNet-VL (R)	33.1	76.0	46.1
M^3 (G+C)	33.3	-	52.8
TF (R)	33.4	-	-
TDConvED$_1$ (V)	29.4	53.6	41.0
TDConvED$_2$ (V)	30.6	58.1	44.1
TDConvED (V)	30.8	58.2	45.2
TDConvED$_1$ (V+C)	31.7	61.6	45.9
TDConvED$_2$ (V+C)	32.5	65.3	47.8
TDConvED (V+C)	32.7	67.2	49.8
TDConvED$_1$ (R)	32.9	72.3	48.3
TDConvED$_2$ (R)	33.3	74.6	51.2
TDConvED (R)	33.8	76.4	53.3

2015). All the metrics are computed using the API [1] released by Microsoft COCO Evaluation Server (Chen et al. 2015).

Compared Approaches. To empirically verify the merit of TDConvED, we compared our method with several conventional LSTM based models, including a CNN Encoder plus a LSTM Decoder model (**MP-LSTM**) (Venugopalan et al. 2015b), LSTM based Encoder-Decoder model (**S2VT**) (Venugopalan et al. 2015a) and Temporal Attention based LSTM network (**TA**) (Yao et al. 2015). We also compared TDConvED with state-of-the-art methods: **GRU-RCN** (Ballas et al. 2015), **LSTM-E** (Pan et al. 2016), **p-RNN** (Yu et al. 2016), **MM** (only visual input) (Ramanishka et al. 2016), **MA-LSTM** (Xu et al. 2017), **BAE** (Baraldi, Grana, and Cucchiara 2017), **hLSTMat** (Song et al. 2017), **MAMRNN** (Li, Zhao, and Lu 2017), **TF** (Zhao, Li, and Lu 2018), **M^3** (Wang et al. 2018), **PickNet-VL** (Chen et al. 2018) and **MCNN+MCF** (Wu and Han 2018). In addition to TDConvED, two different versions, named as TDConvED$_1$ and TDConvED$_2$, are also taken into comparisons. The former extends the basic CNN plus RNN model (MP-LSTM) by simply replacing the RNN-based decoder with our designed convolutional decoder, and the latter is more similar to TDConvED that only excludes temporal attention mechanism.

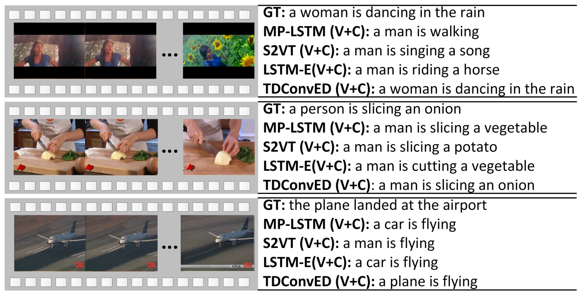
[1]https://github.com/tylin/coco-caption

Table 2: Performance of TDConvED and other RNN-based approaches on MSR-VTT, where B@4, M and C are short for BLEU@4, METEOR and CIDEr-D scores. All values are reported as percentage (%). The short name in the brackets indicates the frame/clip features, where G, C, R and A denotes GoogleNet, C3D, ResNet and Audio feature.

Model	M	C	B@4
MP-LSTM (R)	25.4	35.8	34.1
MP-LSTM (G+C+A)	25.6	38.1	35.7
S2VT (R)	25.8	36.7	34.4
S2VT (G+C+A)	26.0	39.1	36.0
TA (R)	24.9	34.5	33.2
TA (G+C+A)	25.1	36.7	34.8
LSTM-E (R)	25.7	36.1	34.5
LSTM-E (G+C+A)	25.8	38.5	36.1
hLSTMat (R)	26.3	-	38.3
MA-LSTM (G+C+A)	26.5	41.0	36.5
M^3 (V+C)	26.6	-	38.1
MM (R+C)	27.0	41.8	38.3
MCNN+MCF (R)	27.2	42.1	38.1
PickNet-VL (R)	27.2	42.1	38.9
TDConvED$_1$ (R)	26.8	40.7	37.1
TDConvED$_2$ (R)	27.2	41.9	39.0
TDConvED (R)	27.5	42.8	39.5

Experimental Results and Analysis

Quantitative Analysis. Table 1 shows the performances of different models on MSVD. It is worth noting that the reported performances of different approaches are often based on different frame/clip representations. For fair comparisons, our proposed models are evaluated on three commonly adopted frame/clip representations, i.e., output from VGG, the concatenation of features from VGG and C3D, and output from ResNet. Overall, the results across three evaluation metrics with the same input frame/clip representations consistently indicate that our proposed TDConvED achieves better results against three conventional RNN-based models including non-attention models (MP-LSTM and S2VT) and attention-based approach (TA). Specifically, by equipping decoder with convolutions, TDConvED$_1$ (V+C) makes the relative improvement over MP-LSTM (V+C) which adopts RNN-based decoder by 6.7%, 11.8% and 16.5% in METEOR, CIDEr-D and BLEU@4, respectively. S2VT which exploits the context among the input frame/clip sequence through RNN-based encoder improves MP-LSTM, but the performances are still lower than our TDConvED$_2$. The result indicates the merit of leveraging temporal deformable convolutions in TDConvED$_2$ than RNN in encoder-decoder architecture for video captioning. Moreover, by additionally incorporating temporal attention mechanism into MP-LSTM model, TA leads to a performance boost. Similar in spirit, TDConvED improves TDConvED$_2$ by further taking temporal attention into ac-


GT: a woman is dancing in the rain
MP-LSTM (V+C): a man is walking
S2VT (V+C): a man is singing a song
LSTM-E(V+C): a man is riding a horse
TDConvED (V+C): a woman is dancing in the rain

GT: a person is slicing an onion
MP-LSTM (V+C): a man is slicing a vegetable
S2VT (V+C): a man is slicing a potato
LSTM-E(V+C): a man is cutting a vegetable
TDConvED (V+C): a man is slicing an onion

GT: the plane landed at the airport
MP-LSTM (V+C): a car is flying
S2VT (V+C): a man is flying
LSTM-E(V+C): a car is flying
TDConvED (V+C): a plane is flying

Figure 4: Sentence generation results on MSVD dataset. The videos are represented by sampled frames and the output sentences are generated by 1) GT: ground truth sentences, 2) MP-LSTM, 3) S2VT, 4) LSTM-E and 5) our TDConvED.

count. There is a performance gap between TDConvED and TA. Though both runs involve the utilization of temporal attention, they are fundamentally different in the way that TA explores the attention distribution over original frame/clip features depending on the hidden state of RNN-based decoder, and TDConvED is by measuring attention distribution over the contextually encoded frame/clip features conditioned on the intermediate state of convolutional decoder. This again verifies the effectiveness of convolutional encoder-decoder architecture. In addition, compared to other state-of-the-art techniques (e.g., BAE, p-RNN, PickNet-VL, M^3, and TF), our TDConvED achieves better results in terms of all the three evaluation metrics. The results basically indicate that exploring convolutions in both encoder and decoder networks is a promising direction for video captioning.

The performance comparisons on MSR-VTT are summarized in Table 2. Our TDConvED consistently outperforms other baselines. Similar to the observations on MSVD, TDConvED$_1$ exhibits better performance than the basic CNN plus RNN model, MP-LSTM, by replacing RNN with convolutions in decoder. Moreover, TDConvED$_2$ which additionally utilizing temporal deformable convolutional encoder to encode input frame/clip sequence performs better than TDConvED$_1$, and larger degree of improvement is attained when employing temporal attention mechanism in convolutional encoder-decoder structure by TDConvED.

Qualitative Analysis. Figure 4 shows a few sentence examples generated by different methods and human-annotated ground truth. It is easy to see that all of these automatic methods can generate somewhat relevant sentences, while TDConvED can predict more relevant keywords by exploiting long-range contextual relations via temporal deformable convolutional encoder-decoder structure and temporal attention mechanism. Compared to "walking" in the sentence generated by MP-LSTM, "dancing" in TDConvED is more precise to describe the first video. Similarly, the object "onion" presents the second video more exactly.

Effect of the Number of Convolutional Blocks. In order to investigate the effect of the number of stacked convolutional blocks in our TDConvED, two additional experiments are conducted by varying the number of stacked convolutional blocks in encoder and decoder, respectively. As the encoder in TDConvED$_1$ is to perform mean pooling over the features of frames/clips and irrespective of convolutions, we first examine the impact of the convolutional block number in decoder of TDConvED$_1$. Table 3(a) reports the comparison when the number ranges from 1 to 3. The best performances are achieved when the number of stacked convolutional blocks is 2. Next, by fixing the number of convolutional blocks in decoder as 2, Table 3(b) summarizes the performances of TDConvED$_2$ (V+C) when the convolutional block number in encoder varies from 1 to 3. The results show that the setting with 2 stacked temporal deformable convolutional blocks in encoder outperforms others.

Table 3: The effect of the number of stacked conv blocks (i.e., #blocks) in (a) decoder and (b) encoder on MSVD.

(a) Decoder in TDConvED$_1$

#blocks	M	C
1	31.6	57.9
2	31.7	61.6
3	31.4	59.8

(b) Encoder in TDConvED$_2$

#blocks	M	C
1	32.2	64.3
2	32.5	65.3
3	32.3	62.8

Table 4: Comparison of training time between TDConvED and MP-LSTM on MSVD (Nvidia K40 GPU).

Model	#param	training time	B@4	M
MP-LSTM (V)	11.15 M	2370 s/epoch	37.0	29.2
TDConvED (V)	16.79 M	1070 s/epoch	45.2	30.8

Comparisons on Training Time. Table 4 details the training time of our TDConvED and MP-LSTM. TDConvED is benefited from parallelization for training and the results clearly demonstrate the advantage of TDConvED at training speed, which is about 2.2× faster than MP-LSTM.

Conclusions

We have presented Temporal Deformable Convolutional Encoder-Decoder Networks (TDConvED) architecture, which explores feed-forward convolutions instead of RNN in sequence-to-sequence learning for video captioning. Particularly, we delve into the problem from the viewpoint of stacking several convolutional blocks to capture long-term contextual relationships for sequence learning. To verify our claim, we have built an encoder-decoder framework that encodes each frame/clip with contexts to intermediate states through multiple stacked convolutional blocks and then operates convolutional blocks on the concatenation of video features and the embeddings of past words to generate the next word. Moreover, through a new design of temporal deformable convolution, the encoder is further endowed with more power of capturing dynamics in temporal extents. As our encoder-decoder structure only involves feed-forward convolutions, the model allows parallel computation in a sequence at training. Extensive experiments conducted on both MSVD and MSR-VTT datasets validate our proposal and analysis. More remarkably, the improvements of our model are clear when comparing to RNN-based techniques.

Acknowledgments

This work is partially supported by NSF of China under Grant 61672548, U1611461, 61173081, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

References

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. C. 2015. Delving deeper into convolutional networks for learning video representations. *arXiv preprint arXiv:1511.06432*.

Banerjee, S., and Lavie, A. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *ACL*.

Baraldi, L.; Grana, C.; and Cucchiara, R. 2017. Hierarchical boundary-aware neural encoder for video captioning. In *CVPR*.

Bradbury, J.; Merity, S.; Xiong, C.; and Socher, R. 2016. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*.

Chen, D., and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

Chen, X.; Fang, H.; Lin, T.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Chen, S.; Chen, J.; Jin, Q.; and Hauptmann, A. 2017. Video captioning with guidance of multimodal latent topics. In *ACM MM*.

Chen, Y.; Wang, S.; Zhang, W.; and Huang, Q. 2018. Less is more: Picking informative frames for video captioning. *CoRR* abs/1803.01457.

Dauphin, Y. N.; Fan, A.; Auli, M.; and Grangier, D. 2017. Language modeling with gated convolutional networks. In *ICML*.

Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. In *ICML*.

Hao, W.; Zhang, Z.; and Guan, H. 2018. Integrating both visual and audio cues for enhanced video caption. In *AAAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Kalchbrenner, N.; Espeholt, L.; Simonyan, K.; van den Oord, A.; Graves, A.; and Kavukcuoglu, K. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Li, F. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Li, Y.; Yao, T.; Pan, Y.; Chao, H.; and Mei, T. 2018. Jointly localizing and describing events for dense video captioning. In *CVPR*.

Li, X.; Zhao, B.; and Lu, X. 2017. MAM-RNN: multi-level attention model based RNN for video captioning. In *IJCAI*.

Meng, F.; Lu, Z.; Wang, M.; Li, H.; Jiang, W.; and Liu, Q. 2015. Encoding source language with convolutional neural network for machine translation. In *ACL*.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

Pan, Y.; Yao, T.; Li, H.; and Mei, T. 2017. Video captioning with transferred semantic attributes. In *CVPR*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Ramanishka, V.; Das, A.; Park, D. H.; Venugopalan, S.; Hendricks, L. A.; Rohrbach, M.; and Saenko, K. 2016. Multimodal video description. In *ACM MM*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. Imagenet large scale visual recognition challenge. *IJCV*.

Song, J.; Gao, L.; Guo, Z.; Liu, W.; Zhang, D.; and Shen, H.-T. 2017. Hierarchical LSTM with adjusted temporal attention for video captioning. In *IJCAI*.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Vedantam, R.; Zitnick, C. L.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R. J.; Darrell, T.; and Saenko, K. 2015a. Sequence to sequence - video to text. In *ICCV*.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R. J.; and Saenko, K. 2015b. Translating videos to natural language using deep recurrent neural networks. In *NAACL*.

Wang, J.; Wang, W.; Huang, Y.; Wang, L.; and Tan, T. 2018. M^3: Multimodal memory modelling for video captioning. In *CVPR*.

Wu, A., and Han, Y. 2018. Multi-modal circulant fusion for video-to-language and backward. In *IJCAI*.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

Xu, J.; Yao, T.; Zhang, Y.; and Mei, T. 2017. Learning multimodal attention LSTM networks for video captioning. In *ACM MM*.

Yao, L.; Torabi, A.; Cho, K.; Ballas, N.; Pal, C. J.; Larochelle, H.; and Courville, A. C. 2015. Describing videos by exploiting temporal structure. In *ICCV*.

Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2017a. Incorporating copying mechanism in image captioning for learning novel objects. In *CVPR*.

Yao, T.; Pan, Y.; Li, Y.; Qiu, Z.; and Mei, T. 2017b. Boosting image captioning with attributes. In *ICCV*.

Yao, T.; Pan, Y.; Li, Y.; and Mei, T. 2018. Exploring visual relationship for image captioning. In *ECCV*.

Yu, H.; Wang, J.; Huang, Z.; Yang, Y.; and Xu, W. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*.

Zhao, B.; Li, X.; and Lu, X. 2018. Video captioning with tube features. In *IJCAI*.