Adaptive Feature Abstraction for Translating Video to Text

Yunchen Pu,[†][*] **Martin Renqiang Min,**[‡] **Zhe Gan,**[†] **Lawrence Carin**[†]

[†]Department of Electrical and Computer Engineering, Duke University
{yunchen.pu, zhe.gan, lcarin}@duke.edu
[‡]Machine Learning Group, NEC Laboratories America
renqiang@nec-labs.com

Abstract

Previous models for video captioning often use the output from a specific layer of a Convolutional Neural Network (CNN) as video features. However, the variable context-dependent semantics in the video may make it more appropriate to adaptively select features from the multiple CNN layers. We propose a new approach to generating adaptive spatiotemporal representations of videos for the captioning task. A novel attention mechanism is developed, which adaptively and sequentially focuses on different layers of CNN features (levels of feature "abstraction"), as well as local spatiotemporal regions of the feature maps at each layer. The proposed approach is evaluated on three benchmark datasets: YouTube2Text, M-VAD and MSR-VTT. Along with visualizing the results and how the model works, these experiments quantitatively demonstrate the effectiveness of the proposed adaptive spatiotemporal feature abstraction for translating videos to sentences with rich semantics.

Introduction

Videos represent among the most widely used forms of data, and their accurate characterization poses an important challenge for computer vision and machine learning. Generating a natural-language description of a video, termed video captioning, is an important component of video analysis. Inspired by the successful encoder-decoder framework used in machine translation (Bahdanau, Cho, and Bengio 2015; Cho et al. 2014; Sutskever, Vinyals, and Le 2014) and image caption generation (Karpathy and Li 2015; Kiros, Salakhutdinov, and Zemel 2014; Mao et al. 2015; Pu et al. 2016; Vinyals et al. 2015; Gan et al. 2017a; 2017b), most recent work on video captioning (Donahue et al. 2015; Pan et al. 2016b; Venugopalan et al. 2015a; 2015b; Yao et al. 2015; Yu et al. 2016) employs a two-dimensional (2D) or three-dimensional (3D) Convolutional Neural Network (CNN) as an encoder, mapping an input video to a compact feature-vector representation. A Recurrent Neural Network (RNN) is typically employed as a decoder, unrolling the feature vector to generate a sequence of words of arbitrary length.

Despite achieving encouraging success in video captioning, previous models suffer important limitations. Often the

---

[*]Most of this work was done when the first author was a summer intern at NEC Laboratories America.

rich video content is mapped to a single feature vector for caption generation; this approach is prone to missing detailed and localized spatiotemporal information. To mitigate this, one may employ methods to focus attention on local regions of the feature map, but typically this is done with features from a selected (usually top) CNN layer. By employing features from a fixed CNN layer, the algorithm is limited in its ability to model rich, context-aware semantics that requires focusing on different feature abstraction levels. As investigated in Mahendran and Vedaldi; Zeiler and Fergus (2015; 2014), the feature characteristics/abstraction is correlated with the CNN layers: features from layers at or near the top of a CNN tend to focus on global (extended) visual percepts, while features from lower CNN layers provide more local, fine-grained information. It is desirable to select/weight features from different CNN layers adaptively when decoding a caption, selecting different levels of feature abstraction by sequentially emphasizing features from different CNN layers. In addition to focusing on features from different CNN layers, it is also desirable to emphasize local spatiotemporal regions in feature maps at particular layers.

To realize these desiderata, our proposed decoding process for generating a sequence of words dynamically emphasizes different levels (CNN layers) of 3D convolutional features, to model important coarse or fine-grained spatiotemporal structure. Additionally, the model adaptively attends to different locations within the feature maps at particular layers. While some previous models use 2D CNN features to generate video representations, our model adopts features from a deep 3D convolutional neural network (C3D). Such features have been shown to be effective for video representation, action recognition and scene understanding (Tran et al. 2015), by learning the spatiotemporal features that can provide better appearance and motion information. In addition, the proposed model is inspired by the recent success of attention-based models that mimic human perception (Mnih et al. 2014; Xu et al. 2015).

The proposed model, dual Adaptive Feature Representation (dualAFR), involves comparing and evaluating different levels of 3D convolutional feature maps. However, there are three challenges that must be overcome to directly compare features between layers: (*i*) the features from different C3D levels have distinct dimensions, undermining the direct use of a multi-layer perceptron (MLP)

based attention model (Bahdanau, Cho, and Bengio 2015; Xu et al. 2015); (*ii*) the features represented in each layer are not spatiotemporally aligned, undermining our ability to quantify the value of features at a specific spatiotemporal location, based on information from *all* CNN layers; and (*iii*) the semantic meaning of feature vectors from the convolutional filters of C3D varies across layers, making it difficult to feed the features into the same RNN decoder.

To address these issues, one may use either pooling or MLPs to map different levels of features to the similar semantic-space dimension. However, these approaches either lose feature information or have too many parameters to generalize well. In our approach, we employ convolution operations to achieve spatiotemporal alignment among C3D features from different levels and attention mechanisms to dynamically select context-dependent feature abstraction information.

The principal contributions of this paper are as follows: (*i*) A new video-caption-generation model is developed by dynamically modeling context-dependent feature abstractions; (*ii*) new attention mechanisms are developed to adaptively and sequentially emphasize different levels of feature abstraction (CNN layers), while also imposing attention within local spatiotemporal regions of the feature maps at each layer; (*iii*) 3D convolutional transformations are introduced to achieve spatiotemporal and semantic feature consistency across different layers; (*iv*) the proposed model is demonstrated to outperform other multi-level feature based methods, such as hypercolumns (Hariharan et al. 2015), and achieves state-of-the-art performance on several benchmark datasets using diverse automatic metrics and human evaluations.

Related Work

Recent work often develops a probabilistic model of the caption, conditioned on a video. In Donahue et al.; Venugopalan et al.; Venugopalan et al.; Yu et al.; Pan et al. (2015; 2015a; 2015b; 2016; 2016a) the authors performed video analysis by applying a 2D CNN pretrained on ImageNet, with the top-layer output of the CNN used as features. Given the *sequence* of features extracted from the video frames, the video representation is then obtained by a CRF (Donahue et al. 2015), mean pooling (Venugopalan et al. 2015b), weighted mean pooling with attention (Yu et al. 2016), or via the last hidden state of an RNN encoder (Venugopalan et al. 2015a). In Yao et al. (2015), the 2D CNN features are replaced with a 3D CNN to model the short temporal dynamics. These works were followed by Pan et al.; Shetty and Laaksonenl (2016b; 2016), which jointly embedded the 2D CNN features and spatiotemporal features extracted from a 3D CNN (Tran et al. 2015). More recently, there has been a desire to leverage auxiliary information to improve the performance of encoder-decoder models. In Pasunuru and Bansal (2017), auxiliary encoders and decoders are introduced to utilize extra video and sentence data. The entire model is learned by both predicting sentences conditioned on videos and self-reconstruction for the videos and captions. Similiar work includes Ramanishka et al.; Chen et al. (2016; 2017) where extra audio and topics of videos are leveraged. However, all of these previous models utilize features extracted from the top layer of the CNN.

There is also work that combines mult-level features from a CNN. In Sermanet, Kavukcuoglu, and Chintala (2013), a combination of intermediate layers is employed with the top layer for pedestrian detection, while a hypercolumn representation is utilized for object segmentation and localization in Hariharan et al. (2015) . Our proposed model is mostly related to Ballas et al. (2016), but distinct in important ways. The intermediate convolutional feature maps are leveraged, like Ballas et al. (2016), but an attention model is developed instead of the "stack" RNN in Ballas et al. (2016). In addition, a decoder enhanced with two attention mechanisms is constructed for generating captions, while a simple RNN decoder is employed in (Ballas et al. 2016). Finally, we use features extracted from C3D instead of a 2D CNN.

Method

Consider N training videos, the nth of which is denoted $\mathbf{X}^{(n)}$, with associated caption $\mathbf{Y}^{(n)}$. The length-T_n caption is represented $\mathbf{Y}^{(n)} = (\boldsymbol{y}_1^{(n)}, \ldots, \boldsymbol{y}_{T_n}^{(n)})$, with $\boldsymbol{y}_t^{(n)}$ a 1-of-V encoding vector, with V the size of the vocabulary.

For each video, the C3D feature extractor (Tran et al. 2015) produces a set of features $\mathbf{A}^{(n)} = \{\boldsymbol{a}_1^{(n)}, \ldots, \boldsymbol{a}_L^{(n)}, \boldsymbol{a}_{L+1}^{(n)}\}$, where $\{\boldsymbol{a}_1^{(n)}, \ldots, \boldsymbol{a}_L^{(n)}\}$ are feature maps extracted from L convolutional layers, and $\boldsymbol{a}_{L+1}^{(n)}$ is a vector obtained from the top fully-connected layer. For notational simplicity, we omit all the superscript (n) over $\mathbf{X}, \mathbf{Y}, \boldsymbol{y}, \mathbf{A}, \boldsymbol{a}$ and subscript n under T throughout the paper. Details are provided in Appendix.

Caption Model

The t-th word in a caption, $\boldsymbol{y}_t$, is mapped to an M-dimensional vector $\boldsymbol{w}_t = \mathbf{W}_e \boldsymbol{y}_t$, where $\mathbf{W}_e \in \mathbb{R}^{M \times V}$ is a learned word-embedding matrix, *i.e.*, $\boldsymbol{w}_t$ is a column of $\mathbf{W}_e$ chosen by the one-hot $\boldsymbol{y}_t$. The probability of caption $\mathbf{Y} = \{\boldsymbol{y}_t\}_{t=1,T}$ is defined as

$$p(\mathbf{Y}|\mathbf{A}) = p(\boldsymbol{y}_1|\mathbf{A})\prod_{t=2}^{T} p(\boldsymbol{y}_t|\boldsymbol{y}_{<t}, \mathbf{A}) . \qquad (1)$$

Specifically, the first word $\boldsymbol{y}_1$ is drawn from $p(\boldsymbol{y}_1|\mathbf{A}) = \text{softmax}(\mathbf{V}\boldsymbol{h}_1)$, where $\boldsymbol{h}_1 = \tanh(\mathbf{C}\boldsymbol{a}_{L+1})$; throughout the paper bias terms are omitted, for simplicity. $\mathbf{C}$ is a weight matrix mapping video features $\boldsymbol{a}_{L+1}$ to the RNN hidden state space. $\mathbf{V}$ is a matrix connecting the RNN hidden state to a softmax, for computing a distribution over words. All other words in the caption are then sequentially generated using an RNN, until the end-sentence symbol is generated. Conditional distribution $p(\boldsymbol{y}_t|\boldsymbol{y}_{<t}, \mathbf{A})$ is specified as $\text{softmax}(\mathbf{V}\boldsymbol{h}_t)$, where $\boldsymbol{h}_t$ is recursively updated as $\boldsymbol{h}_t = \mathcal{H}(\boldsymbol{w}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{z}_t)$. $\boldsymbol{z}_t = \phi(\boldsymbol{h}_{t-1}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$ is the context vector used in the attention mechanism, capturing the relevant visual features associated with the spatiotemporal attention (also weighting level of feature abstraction), as detailed below. The transition function $\mathcal{H}(\cdot)$ is implemented with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997).
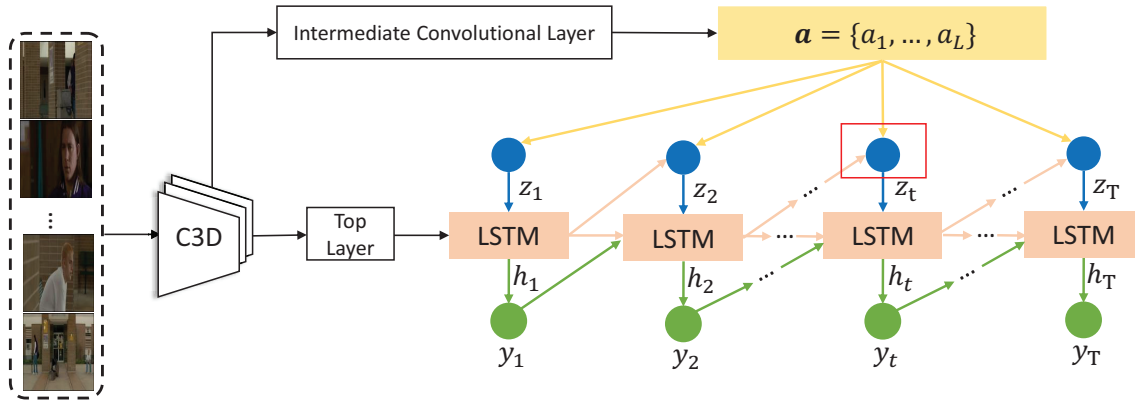
Figure 1: Illustration of our proposed caption-generation model. The model leverages a fully-connected map from the top layer as well as convolutional maps from different mid-level layers of a pretrained 3D convolutional neural network (C3D). The context vector z_t is generated from the previous hidden unit h_{t-1} and the convolutional maps $\{a_1, \ldots, a_L\}$ (the red frame), which is detailed in Figure 2.

Given the video $\mathbf{X}$ (with features $\mathbf{A}$) and associated caption $\mathbf{Y}$, the objective function is the sum of the log-likelihood of the caption conditioned on the video representation:

$$\log p(\mathbf{Y}|\mathbf{A}) = \log p(\boldsymbol{y}_1|\mathbf{A}) + \sum_{t=2}^{T} \log p(\boldsymbol{y}_t|\boldsymbol{y}_{<t}, \mathbf{A}). \quad (2)$$

Equation (2) is a function of all model parameters to be learned; they are not explicitly depicted in (2) for notational simplicity. Further, (2) corresponds to a single video-caption pair, and when training we sum over all such training pairs. The proposed model is illustrated in Figure 1.

Attention Mechanism

We first define notation needed to describe attention mechanism $\phi(\boldsymbol{h}_{t-1}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_L)$. Each feature map, $\boldsymbol{a}_l$, is a 4D tensor, with elements corresponding to two spatial coordinates (*i.e.*, vertical and horizontal dimensions in a given frame), the third tensor index is the frame-index dimension, and the fourth dimension is associated with the filter index (for the convolutional filters). To be explicit, at CNN layer l, the number of dimensions of this tensor are denoted $n_x^l \times n_y^l \times n_z^l \times n_F^l$, with respective dimensions corresponding to vertical, horizontal, frame, and filter (*e.g.*, n_F^l convolutional filters at layer l). Note that dimensions n_x^l, n_y^l and n_z^l vary with layer level l (getting smaller with increasing l, due to pooling).

We define $\boldsymbol{a}_{i,l}$ as a n_F^l-dimensional *vector*, corresponding to a fixed coordinate in three of the tensor dimensions, *i.e.*, $i \in [1, \ldots, n_x^l] \times [1, \ldots, n_y^l] \times [1, \ldots, n_z^l]$, while sweeping across all n_F^l feature/filter dimensions. Further, define $\boldsymbol{a}_l(k)$ as a 3D tensor, associated with 4D tensor $\boldsymbol{a}_l$. Specifically, $\boldsymbol{a}_l(k)$ corresponds to the 3D tensor manifested from $\boldsymbol{a}_l$, with $k \in \{1, \ldots, n_F^l\}$ a fixed coordinate in the dimension of the filter index. Hence, $\boldsymbol{a}_l(k)$ corresponds to the 3D feature map at layer l, due to the kth filter at that layer.

We introduce two attention mechanisms when predicting word $\boldsymbol{y}_t$: (*i*) spatiotemporal-localization attention, and (*ii*) abstraction-level attention; these, respectively, measure the

relative importance of a particular spatiotemporal location and a particular CNN layer (feature abstraction) for producing $\boldsymbol{y}_t$, based on the word-history information $\boldsymbol{y}_{<t}$.

To achieve this, we seek to map $\boldsymbol{a}_l \rightarrow \hat{\boldsymbol{a}}_l$, where 4D tensors $\hat{\boldsymbol{a}}_l$ all have the same dimensions, are embedded into similar semantic spaces, and are aligned spatialtem-porally. Specifically, $\hat{\boldsymbol{a}}_l$, $l = 1, \ldots, L - 1$ are aligned in the above ways with $\boldsymbol{a}_L$. To achieve this, we filter each $\boldsymbol{a}_l$, $l = 1, \ldots, L - 1$, and then apply max-pooling; the filters seek semantic alignment of the features (including feature dimension), and the pooling is used to spatiotemporally align the features with $\boldsymbol{a}_L$. Specifically, consider

$$\hat{\boldsymbol{a}}_l = f(\textstyle\sum_{k=1}^{n_F^l} \boldsymbol{a}_l(k) * \mathbf{U}_{k,l}), \quad (3)$$

for $l = 1, \ldots, L - 1$, and with $\hat{\boldsymbol{a}}_L = \boldsymbol{a}_L$. As discussed above, $\boldsymbol{a}_l(k)$ is the 3D feature map (tensor) for dictionary $k \in \{1, \ldots, n_F^l\}$ at layer l. $\mathbf{U}_{k,l}$ is the 3D convolutional filters to achieve alignment which is a 4D learnable tensor. The convolution $*$ in (3) operates in the three shift dimensions, and $\boldsymbol{a}_l(k) * \mathbf{U}_{k,l}$ manifests a 4D tensor. Specifically, each of the n_F^L 3D "slices" of $\mathbf{U}_{k,l}$ are spatiotemporally convolved (3D convolution) with $\boldsymbol{a}_{k,l}$, and after summing over the n_F^l convolutional filters, followed by $f(\cdot)$, this manifests each of the n_F^L 3D slices of $\hat{\boldsymbol{a}}_l$. Function $f(\cdot)$ is an element-wise nonlinear activation function, followed by max pooling, with the pooling dimensions meant to realize final dimensions consistent with $\boldsymbol{a}_L$, *i.e.*, dimension $n_x^L \times n_y^L \times n_z^L \times n_F^L$. Consequently, $\hat{\boldsymbol{a}}_{i,l} \in \mathbb{R}^{n_F^L}$ is a feature vector with $i \in [1, \ldots, n_x^L] \times [1, \ldots, n_y^L] \times [1, \ldots, n_z^L]$. Note that it may only require one 3D tensor $\mathbf{U}_l$ applied on each 3D slices $\boldsymbol{a}_l(k)$ for each layer to achieve spatiotemporal alignment of the layer-dependent features. However, the features from two distinct layers will not be in the same "semantic" space, making it difficult to assess the value of the layer-dependent features. The multiple tensors in set $\{\mathbf{U}_{k,l}\}$ provide the desired semantic alignment between layers, allowing analysis of the value of features from different layers via a single MLP, in the following (5) and (7).
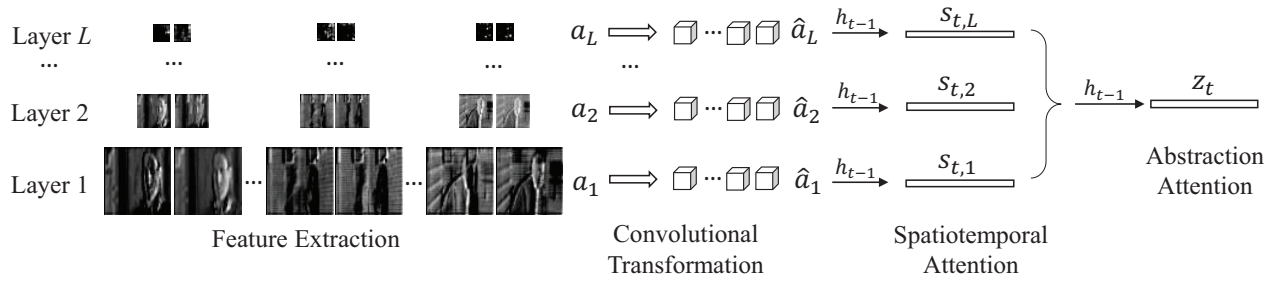
Figure 2: Illustration of our attention mechanism. The video features are extracted by C3D, followed by a 3D convolutional transformation. At each time step, the spatiotemporal attention takes these features and previous hidden units to generate L feature vectors which are fed to the abstraction attention, manifesting the context vector.

With $\{\hat{\boldsymbol{a}}_l\}_{l=1,L}$ semantically and spatiotemporally aligned, we now seek to jointly quantify the value of a particular spatiotemporal region and a particular feature layer ("abstraction") for prediction of the next word. For each $\hat{\boldsymbol{a}}_{i,l}$, the attention mechanism generates two positive weights, α_{ti} and β_{tl}, which measure the relative importance of location i and layer l for producing $\boldsymbol{y}_t$ based on $\boldsymbol{y}_{<t}$. Attention weights α_{ti} and β_{tl} and context vector $\boldsymbol{z}_t$ are computed as

$$e_{ti} = \boldsymbol{w}_\alpha^T \tanh(\mathbf{W}_{a\alpha}\hat{\boldsymbol{a}}_i + \mathbf{W}_{h\alpha}\boldsymbol{h}_{t-1}), \tag{4}$$

$$\alpha_{ti} = \mathrm{softmax}(\{e_{ti}\}), \quad \boldsymbol{s}_t = \psi(\{\hat{\boldsymbol{a}}_i\}, \{\alpha_{ti}\}), \tag{5}$$

$$b_{tl} = \boldsymbol{w}_\beta^T \tanh(\mathbf{W}_{s\beta}\boldsymbol{s}_{tl} + \mathbf{W}_{h\beta}\boldsymbol{h}_{t-1}), \tag{6}$$

$$\beta_{tl} = \mathrm{softmax}(\{b_{tl}\}), \quad \boldsymbol{z}_t = \psi(\{\boldsymbol{s}_{tl}\}, \{\beta_{tl}\}), \tag{7}$$

where $\hat{\boldsymbol{a}}_i$ is a vector composed by stacking $\{\hat{\boldsymbol{a}}_{i,l}\}_{l=1,L}$ (all features at position i). e_{ti} and b_{tl} are scalars reflecting the importance of spatiotemporal region i and layer l to predicting $\boldsymbol{y}_t$, while α_{ti} and β_{tl} are *relative* weights of this importance, reflected by the softmax output. $\psi(\cdot)$ is a function developed further below, that returns a single feature vector when given a set of feature vectors, and their corresponding weights across all i or l. Vector $\boldsymbol{s}_{tl}$ reflects the sub-portion of $\boldsymbol{s}_t$ associated with layer l.

In (5) we provide attention in the spatiotemporal dimensions, with that spatiotemporal attention shared across all L (now aligned) CNN layers. In (7) the attention is further refined, focusing attention in the layer dimension. To make the following discussion concrete, we describe the attention function within the context of $\boldsymbol{z}_t = \psi(\{\boldsymbol{s}_{tl}\}, \{\beta_{tl}\})$. This function setup is applied in the same way to $\boldsymbol{s}_t = \psi(\{\hat{\boldsymbol{a}}_i\}, \{\alpha_{ti}\})$. The attention model is illustrated in Figure 2.

Soft attention Following Bahdanau, Cho, and Bengio (2015), we formulate the soft attention model by computing a weighted sum of the input features

$$\boldsymbol{z}_t = \psi(\{\boldsymbol{s}_{tl}\}, \{\beta_{tl}\}) = \sum_{l=1}^{L} \beta_{tl}\boldsymbol{s}_{tl}. \tag{8}$$

The model is differentiable for all parameters and can be learned end-to-end using standard back-propagation.

Hard attention Let $\boldsymbol{m}_t \in \{0, 1\}^L$ be a vector of all zeros, and a single one, and the location of the non-zero element of $\boldsymbol{m}_t$ identifies the location to extract features for generating the next word. We impose

$$\boldsymbol{m}_t \sim \mathsf{Mult}(1, \{\beta_{tl}\}), \qquad \boldsymbol{z}_t = \sum_{l=1}^{L} m_{tl}\boldsymbol{s}_{tl}. \tag{9}$$

In this case, optimizing the objective function in (2) is intractable. However, the marginal log-likelihood can be lower-bounded as

$$\log p(\mathbf{Y}|\mathbf{A}) = \mathbb{E}_{p(\boldsymbol{m}|\mathbf{A})} \log p(\mathbf{Y}|\boldsymbol{m}, \mathbf{A}), \tag{10}$$

where $\boldsymbol{m} = \{\boldsymbol{m}_t\}_{t=1,\dots,T}$. We utilize Monte Carlo integration to approximate the expectation, $\mathbb{E}_{p(\boldsymbol{m}|\mathbf{A})}$, and stochastic gradient descent (SGD) for parameter optimization. The gradients are approximated by importance sampling (Mnih and Rezende 2016; Burda, Grosse, and Salakhutdinov 2016) to achieve unbiased estimation and reduce the variance. Details are provided in Appendix.

Experiments

Datasets

We present results on three benchmark datasets: Microsoft Research Video Description Corpus (YouTube2Text) (Chen and Dolan 2011), Montreal Video Annotation Dataset (M-VAD) (Torabi, C Pal, and Courville 2015), and Microsoft Research Video to Text (MSR-VTT) (Xu et al. 2016).

The Youtube2Text contains 1970 Youtube clips, and each video is annotated with around 40 sentences. For fair comparison, we used the same splits as provided in Venugopalan et al. (2015b), with 1200 videos for training, 100 videos for validation, and 670 videos for testing.

The M-VAD is a large-scale movie description dataset, which is composed of 46587 movie snippets annotated with 56752 sentences. We follow the setting in (Torabi, C Pal, and Courville 2015), taking 36920 videos for training, 4950 videos for validation, and 4717 videos for testing.

The MSR-VTT is a newly collected large-scale video dataset, consisting of 20 video categories. The dataset was split into 6513, 2990 and 497 clips in the training, testing and validation sets. Each video has about 20 sentence descriptions. The ground-truth captions in the testing set are not available now. Thus, we split the original training dataset

into a training set of 5513 clips and a testing set of 1000 clips.

We convert all captions to lower case and remove the punctuation, yielding vocabulary sizes $V = 12594$ for Youtube2Text, $V = 13276$ for M-VAD, and $V = 8071$ for MSR-VTT.

We consider the RGB frames of videos as input, and all videos are resized to 112×112 spatially, with 2 frames per second. The C3D (Tran et al. 2015) is pretrained on Sports-1M dataset (Karpathy et al. 2014), consisting of 1.1 million sports videos belonging to 487 categories. We extract the features from four convolutional layers and one fully connected layer, named as *pool2, pool3, pool4, pool5* and *fc-7* in the C3D (Tran et al. 2015), respectively. Detailed model architectures and training settings are provided in Appendix. We do not perform any dataset-specific tuning and regularization other than dropout (Srivastava et al. 2014) and early stopping on validation sets.

Evaluation

The widely used BLEU (Papineni et al. 2002), ME-TEOR (Banerjee and Lavie 2005) and CIDEr (Vedantam, Lawrence, and Parikh 2015) metrics are employed to quantitatively evaluate the performance of our video caption generation model, and other models in the literature. For each dataset, we show three types of results, using part of or all of our model to illustrate the role of each component:

1. *C3D fc7 + LSTM* : The LSTM caption decoder is employed, only using features extracted from the top fully-connected layer. No context vector z_t is generated from intermediate convolutional layer features.

2. *Spatiotemporal attention + LSTM*: The context vector z_t is included, but only features extracted from a certain convolutional layer are employed, *i.e.*, z_t is equal to s_t in (5). The spatiotemporal attention is implemented with the soft attention in (8).

3. *dualAFR*: This is our proposed model. We present results based on both soft attention and hard attention.

To compare our method with other multi-level feature methods, we show several baseline results on Youtube2Text:

1. *MLP* : Each a_l is fed to a different MLP to achieve the same dimension of a_L. The context vector z_t is obtained by abstraction-level and spatiotemporal attention.

2. *Max/Average-pooling*: A max or average pooling operation is utilized to achieve saptiotemporal alignment, and an MLP is then employed to embed the feature vectors into the similar semantic space. Details are provided in Appendix.

3. *Hypercolumn*: This method is similar to max/average-pooling but replace the pooling operation with hypercolumn representation (Hariharan et al. 2015).

In these methods, the attention weights are produced by the soft attention in (8).

To verify the effectiveness of our video caption generation model and C3D features, we also implement a strong baseline method based on the LSTM encoder-decoder network (Cho et al. 2014), where ResNet (He et al. 2016) is employed as the feature extractor on each frame. We denote results using this method as *ResNet + LSTM* .

We also present human evaluation results based on relevance and coherence [1]. Compared with single layer baseline models, the inference of dualAFR is about $1.2 \sim 1.5$ times slower and requires about $2 \sim 4$ times extra memory.

Quantitative Results

Results are summarized in Tables 1 and 2. The proposed models achieve state-of-the-art results on most metrics on all three datasets. The M-to-M method (Pasunuru and Bansal 2017) is the only model showing better BLEU and CIDEr on Youtube2Text. Note that the M-to-M model is trained with two additional datasets: UFC-101 which contains 13,320 video clips and Stanford Natural Language Inference corpus which contains 190,113 sentence pairs. In contrast, we achieve competitive or better results by using only the data inside the training set and analogous pretrained C3D.

Note that our model consistently yields significant improvements over models with only spatiotemporal attention, which further achieve better performance than only taking the C3D top fully-connected layer features; this demonstrates the importance of leveraging intermediate convolutional layer features. In addition, our model outperforms all the baseline results of multi-level feature based methods, demonstrating the effectiveness of our 3D convolutional transformation operation. This is partly a consequence of the sparse connectivity of the convolution operation, which indicates fewer parameters are required.

Human Evaluation

Besides the automatic metrics, we present human evaluation on the Youtube2Text dataset. In each survey, we compare our results from single model (soft attention or hard attention) with the strongest baseline "C3D fc7 + pool4" by taking a random sample of 100 generated captions, and ask the human evaluator to select the result with better relevance and coherence. We obtain 25 repsonses (2448 samples in total) and the results are shown in Table 3. The proposed dualAFR outperforms the strongest baseline on both relevance and coherence which is consistent with the automatic metrics.

Qualitative Results

Following (Xu et al. 2015), we visualize the attention components learned by our model on Youtube2Text. As observed from Figure 3, the spatiotemporal attention aligns the objects in the video well with the corresponding words. In addition, the abstraction-level attention tends to focusing on low level features when the model generates a noun and high level features when an article or a preposition is being generated. More results are provided in Appendix. Examples of generated captions from unseen videos on Youtube2Text are shown in Figure 4. We find the results with abstraction-layer

---

[1]Our human evaluation follows the algorithm in the COCO captioning challenge (http://cocodataset.org/dataset.htm#captions-challenge2015).

Table 1: Results on BLEU-4, METEOR and CIDEr metrics compared to other models and baselines on Youtube2Text and M-VAD datasets.

Methods	Youtube2Text			M-VAD		
	BLEU-4	METEOR	CIDEr	BLEU-4	METEOR	CIDEr
LSTM-E (VGG + C3D) (Pan et al. 2016b)	45.3	31.0	-	-	6.7	-
GRU-RCN (Ballas et al. 2016)	47.90	31.14	67.82	-	-	-
h-RNN (C3D+VGG) (Yu et al. 2016)	49.9	32.6	65.8	-	-	-
TGM (Chen et al. 2017)	48.76	34.36	80.45	-	-	-
M-to-M (Pasunuru and Bansal 2017)	**54.5**	36.0	**92.4**	-	7.4	-
Baselines						
ResNet + LSTM	44.08	30.99	66.88	0.81	6.22	5.54
C3D fc7 + LSTM	45.34	31.21	66.12	0.83	6.31	5.96
C3D fc7 + pool2	45.46	31.53	67.38	0.98	6.42	6.01
C3D fc7 + pool3	48.07	33.52	69.97	1.12	6.71	6.97
C3D fc7 + pool4	48.18	34.47	70.97	1.24	6.89	7.12
C3D fc7 + pool5	47.75	33.35	69.71	1.02	6.49	6.48
Baselines of multi-level features with attention						
MLP + soft attention	35.99	23.56	44.21	0.41	5.99	5.43
Max-pooling + soft attention	45.35	31.50	62.99	0.82	6.21	5.55
Average-pooling + soft attention	48.53	33.59	65.07	0.91	6.45	6.12
Hypercolumn + soft attention	44.92	30.18	63.18	0.88	6.33	6.24
dualAFR						
Soft Attention (Single model)	51.74	36.39	72.18	1.94	7.72	7.98
Hard Attention (Single model)	51.77	36.41	72.21	1.82	7.12	8.12
Soft Attention (Ensemble of 10)	53.94	37.91	78.43	**2.14**	**8.22**	9.03
Hard Attention (Ensemble of 10)	54.27	**38.03**	78.31	2.08	7.12	**9.14**

Table 2: Results on BLEU-4, METEOR and CIDEr metrics compared to other models and baselines on MSR-VTT. [§](Ramanishka et al. 2016); [†](Shetty and Laaksonenl 2016); [§](Chen et al. 2017); [‡](Pasunuru and Bansal 2017).

Method	BLEU-4	METEOR	CIDEr
MMVD[§]	40.7	28.6	46.5
M-to-M[‡]	40.8	28.8	47.1
Aalto[†]	41.1	27.7	46.4
TGM[§]	44.33	29.37	49.26
Baselines			
ResNet + LSTM	39.54	26.59	45.22
C3D fc7 + LSTM	40.17	26.86	45.95
C3D fc7 + pool2	40.43	26.93	47.15
C3D fc7 + pool3	42.04	27.18	48.93
C3D fc7 + pool4	41.98	27.42	48.21
C3D fc7 + pool5	40.83	27.01	47.86
dualAFR			
Soft Attention (Single)	43.72	29.67	50.21
Hard Attention (Single)	43.89	28.71	50.29
Soft Attention (Ensemble)	44.99	**30.16**	51.13
Hard Attention (Ensemble)	**45.01**	29.98	**51.41**

Table 3: Human evaluation results on Youtube2Text.

Method	Relevance	Coherence
Baseline wins	4.78%	1.63%
DualAFR wins	27.53%	6.49%
Not distinguishable	67.69%	91.88%

dicated as "Pool2" *etc.*). This demonstrates the effectiveness of our abstraction layer attention. More results are provided in Appendix.

Conclusion and Future Work

A novel video captioning model has been proposed, that adaptively selects/weights the feature abstraction (CNN layer), as well as the location within a layer-dependent feature map. We have implemented the attention using both "hard" and "soft" mechanisms, with the latter typically delivering better performance. Our model achieves excellent video caption generation performance, and has the capacity to provide interpretable alignments *seemingly* analogous to human perception.

We have focused on analysis of videos and associated captions. Similar ideas may be applied in the future to image captioning. Additionally, the CNN parameters were learned separately as a first step, prior to analysis of the captions. It is also of interest to consider CNN-parameter refinement conditioned on observed training captions.

attention (indicated as "soft attention" or " hard attention") is generally equal to or better than the best results, compared to those only taking a certain convolutional-layer feature (in-
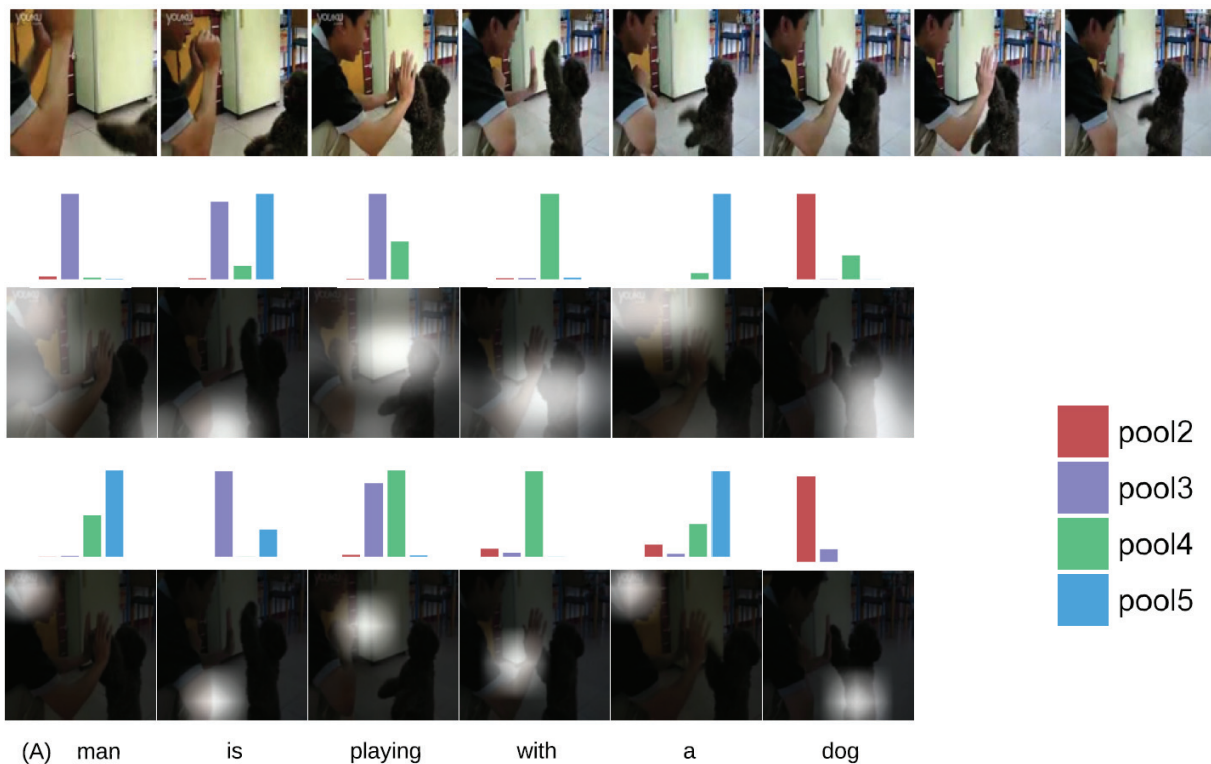
Figure 3: Visualization of attention weights aligned with input video and generated caption on Youtube2Text: (top) Sampled frames of input video, (middle) soft attention, (bottom) hard attention. We show the frame with the strongest attention weights. The bar plot above each frame corresponds layer attention weights β_{tl} when the corresponding word (under the frame) is generated.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Ballas, N.; Yao, L.; Pal, C.; and Courville, A. 2016. Delving deeper into convolutional networks for learning video representations. In *ICLR*.

Banerjee, S., and Lavie, A. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop*.

Burda, Y.; Grosse, R.; and Salakhutdinov, R. 2016. Importance weighted autoencoders. In *ICLR*.

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

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

Cho, K.; Merrienboer, B. V.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Gan, C.; Gan, Z.; He, X.; Gao, J.; and Deng, L. 2017a. Stylenet: Generating attractive visual captions with styles. In *CVPR*.

Gan, Z.; Gan, C.; He, X.; Pu, Y.; Tran, K.; Gao, J.; Carin, L.; and Deng, L. 2017b. Semantic compositional networks for visual captioning. In *CVPR*.

Hariharan, B.; Arbeláez, .; Girshick, R.; and Malik, J. 2015. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*.

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

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation*.

Karpathy, A., and Li, F. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

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

Kiros, R.; Salakhutdinov, R.; and Zemel, R. 2014. Multimodal neural language models. In *ICML*.

Mahendran, A., and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *CVPR*.

Mao, J.; Xu, W.; Yang, Y.; Wang, J.; and Yuille, A. 2015. Deep captioning with multimodal recurrent neural networks (m-RNN). In *ICLR*.

Mnih, A., and Rezende, D. J. 2016. Variational inference for monte carlo objectives. In *ICML*.

Pool2:	a baby is walking
Pool3:	a bear is walking
Pool4:	a bear is walking
Pool5:	a badger is walking
Soft attention:	a turtle is walking
Hard attention:	a turtle is walking
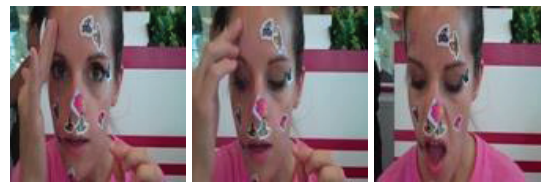
Pool2:	a girl is smiling
Pool3:	a girl is talking
Pool4:	a woman is licking a face
Pool5:	a girl is putting stickers on her face
Soft attention:	a woman is putting stickers on her face
Hard attention:	a woman is putting stickers on her face
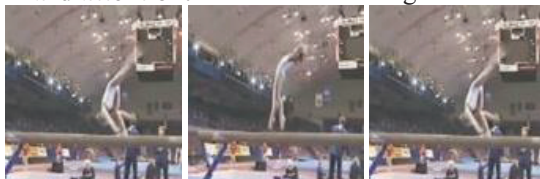
Pool2:	a group of men are playing
Pool3:	a group of men are playing
Pool4:	a girl is doing gymnastics
Pool5:	a gymnast falls down
Soft attention:	a woman is doing gymnastics
Hard attention:	a girl is doing gymnastics
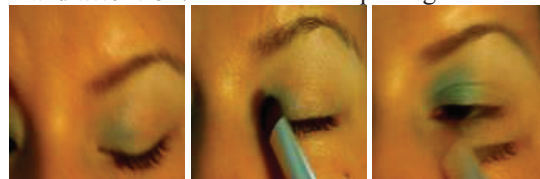
Pool2:	a woman is peeling a toad
Pool3:	a woman is applying eye makeup
Pool4:	a woman is licking a brush
Pool5:	a woman is plucking her face
Soft attention:	a woman is applying eye shadow
Hard attention:	a woman is applying eye makeup

Figure 4: Examples of generated captions with sampled frames of input video on YouTube2Text.

Mnih, V.; Heess, N.; Graves, A.; and Kavukcuoglu, K. 2014. Recurrent models of visual attention. In *NIPS*.

Pan, P.; Xu, Z.; Yang, Y.; Wu, F.; and Zhuang, Y. 2016a. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*.

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

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W. 2002. Bleu: a method for automatic evaluation of machine translation. *Transactions of the Association for Computational Linguistics*.

Pasunuru, R., and Bansal, M. 2017. Multi-task video captioning with video and entailment generation. In *ACL*.

Pu, Y.; Gan, Z.; Henao, R.; Yuan, X.; Li, C.; Stevens, A.; and Carin, L. 2016. Variational autoencoder for deep learning of images, labels and captions. In *NIPS*.

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

Sermanet, P.; Kavukcuoglu, K.; and Chintala, S.and LeCun, Y. 2013. Pedestrian detection with unsupervised multi-stage feature learning. In *CVPR*.

Shetty, R., and Laaksonenl, J. 2016. Frame- and segment-level features and candidate pool evaluation for video caption generation. In *ACM MM*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *JMLR*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Torabi, A.; C Pal, H. L.; and Courville, A. 2015. Using descriptive video services to create a large data source for video annotation research. In *arXiv preprint arXiv:1503.01070*.

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

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

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

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

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Xu, K.; Ba, J. L.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R. S.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

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

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

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

Zeiler, M., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.