Spatio-Temporal Dynamics and Semantic Attribute Enriched Visual Encoding for Video Captioning

Nayyer Aafaq Naveed Akhtar Wei Liu Syed Zulqarnain Gilani Ajmal Mian
Computer Science and Software Engineering,
The University of Western Australia.
nayyer.aafaq@research.uwa.edu.au,{naveed.akhtar, wei.liu, syed.gilani, ajmal.mian}@uwa.edu.au

Abstract

Automatic generation of video captions is a fundamental challenge in computer vision. Recent techniques typically employ a combination of Convolutional Neural Networks (CNNs) and Recursive Neural Networks (RNNs) for video captioning. These methods mainly focus on tailoring sequence learning through RNNs for better caption generation, whereas off-the-shelf visual features are borrowed from CNNs. We argue that careful designing of visual features for this task is equally important, and present a visual feature encoding technique to generate semantically rich captions using Gated Recurrent Units (GRUs). Our method embeds rich temporal dynamics in visual features by hierarchically applying Short Fourier Transform to CNN features of the whole video. It additionally derives high level semantics from an object detector to enrich the representation with spatial dynamics of the detected objects. The final representation is projected to a compact space and fed to a language model. By learning a relatively simple language model comprising two GRU layers, we establish new state-of-the-art on MSVD and MSR-VTT datasets for METEOR and ROUGE$_L$ metrics.

1. Introduction

Describing videos in natural language is trivial for humans, however it is a very complex task for machines. To generate meaningful video captions, machines are required to understand objects, their interaction, spatio-temporal order of events and other such minutiae in videos; yet, have the ability to articulate these details in grammatically correct and meaningful natural language sentences [1]. The bicephalic nature of this problem has recently led researchers from Computer Vision and Natural Language Processing (NLP) to combine efforts in addressing its challenges [3, 4, 5, 30]. Incidentally, wide applications of video captioning in emerging technologies, e.g. procedure generation from instructional videos [2], video indexing and retrieval [45, 55]; have recently caused it to receive attention as a fundamental task in Computer Vision.

Early methods in video captioning and description, e.g. [26, 9] primarily aimed at generating the correct Subject, Verb and Object (a.k.a. SVO-Triplet) in the captions. More recent methods [50, 39] rely on Deep Learning [28] to build frameworks resembling a typical neural machine translation system that can generate a single sentence [57, 33] or multiple sentences [38, 43, 60] to describe videos. The two-pronged problem of video captioning provides a default division for the deep learning methods to encode visual contents of videos using Convolutional Neural Networks (CNNs) [44, 48] and decode those into captions using language models. Recurrent Neural Networks (RNNs) [16, 14, 22] are the natural choice for the latter component of the problem.

Since semantically correct sentence generation has a longer history in the field of NLP, deep learning based captioning techniques mainly focus on language modelling [51, 34]. For visual encoding, these methods forward pass video frames through a pre-trained 2D CNN; or a video clip through a 3D CNN, and extract features from an inner layer of the network - referred as 'extraction layer'. Features of frames/clips are commonly combined with mean pooling to compute the final representation of the whole video. This, and similar other visual encoding techniques [33, 51, 18, 34] - due to the nascency of video captioning research - grossly under-exploit the prowess of visual representation for the captioning task. To the best of our knowledge, this paper presents the first work that concentrates on improving the visual encoding mechanism for the captioning task.

We propose a visual encoding technique to compute representations enriched with spatio-temporal dynamics of the scene, while also accounting for the high-level semantic attributes of the videos. Our visual code ('v' in Fig. 1) fuses information from multiple sources. We process activations of 2D and 3D CNN extraction layers by hierarchically applying Short Fourier Transform [31] to them,

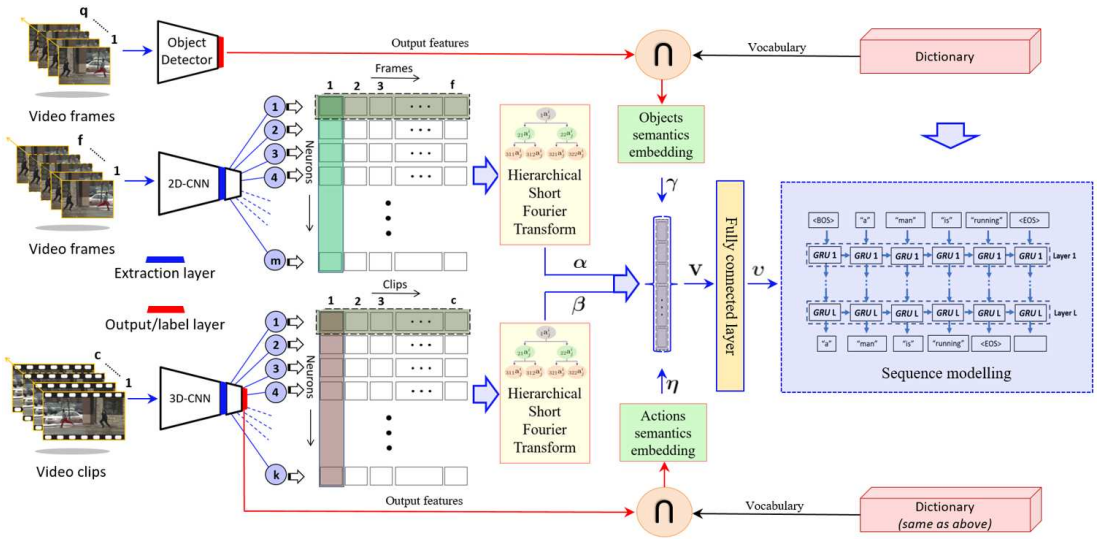

Figure 1. The 'c' clips and 'f' frames of a video are processed with 3D and 2D CNNs respectively. Neuron-wise Short Fourier Transform is applied hierarchically to the extraction layer activations of these networks (using the whole video). This results in spatio-temporal dynamics enriched encodings $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Relevant high-level object semantics $\boldsymbol{\gamma}$ and action semantics $\boldsymbol{\eta}$ are derived using the intersection of vocabulary from the language model dictionary with the labels of 3D CNN and an Object Detector. The output features of the Object Detector are also used to embed spatial dynamics of the scene and plurality of the objects therein. The resulting codes are compressed with a fully-connected layer and used to learn a multi-layer GRU as a language model.

where InceptionResNetv2 [46] and C3D [48] are used as the 2D and 3D CNNs respectively. The proposed neuron-wise activation transformation using *whole videos* results in encoding fine temporal dynamics of the scenes. We encode spatial dynamics by processing objects' locations and their multiplicity information extracted from an Object Detector (YOLO [37]). The semantics attached to the output layers of the Object Detector and the 3D CNN are also exploited to embed high-level semantic attributes in our visual codes. We compress the visual codes and learn a language model using the resulting representation. With highly rich visual codes, a relatively simple Gated Recurrent Unit (GRU) network is proposed for language modeling, comprising two layers, that already results in on-par or better performance compared to the existing sophisticated models [52, 54, 34, 18] on multiple evaluation metrics. The main contributions of this paper are as follows. We propose a visual encoding technique that effectively encapsulates spatio-temporal dynamics of the videos and embeds relevant high-level semantic attributes in the visual codes for video captioning. The proposed visual features contain the detected object attributes, their frequency of occurrences as well as the evolution of their locations over time. We establish the effectiveness of the proposed encoding by learning a GRU-based language model and perform thorough experimentation on MSVD [11] and MSR-VTT [57] datasets. Our method achieves up to 2.64% and 2.44% gain in the state-of-the-art on METEOR and ROUGE$_L$ metrics for these datasets.

2. Related Work

Classical methods in video captioning commonly use template based techniques in which Subject (S), Verb (V), and Object (O) are detected separately and then joined together in a sentence. However, the advancement of deep learning research has also transcended to modern video captioning methods. The latest approaches in this direction generally exploit deep learning for visual feature encoding as well as its decoding into meaningful captions.

In template based approaches, the first successful video captioning method was proposed by Kojima et al. [26] that focuses on describing videos of one person performing one action only. Their heavy reliance on the correctness of manually created activity concept hierarchy and state transition model prevented its extension to more complex videos. Hanckmann et al. [21] proposed a method to automatically describe events involving multiple actions (seven on average), performed by one or more individuals. Whilst most of the prior work was restricted to constrained domains [25, 9], Krishnamoorthy et al. [27] led the early works of describing open domain videos. [20] proposed semantic hierarchies to establish relationships between actor, action and objects. [40] used CRF to model the relationship between visual entities and treated video description as a machine translation problem. However, the aforementioned approaches depend on predefined sentence templates and fill in the template by detecting entities from classical methods. These approaches are not sufficient for the syntactically rich sentence generation to describe open domain videos.

In contrast to the methods mentioned above, deep models directly generate sentences given a visual input. For example LSTM-YT [51] feed in visual contents of video obtained by average pooling all the frames into LSTM and produce the sentences. LSTM-E [33] explores the relevance between the visual context and sentence semantics. The initial visual features in this framework were obtained using 2D-CNN and 3D-CNN whereas the final video representation was achieved by average pooling the features from frames / clips neglecting the temporal dynamics of the video. TA [59] explored the temporal domain of video by introducing an attention mechanism to assign weights to the features of each frame and later fused them based on attention weights. S2VT [50] incorporated optical flow to cater for the temporal information of the video. SCN-LSTM [18] proposed semantic compositional network that can detect the semantic concepts from mean pooled visual content of the video and fed that information into a language model to generate captions with more relevant words. LSTM-TSA[34] proposed a transfer unit that extracts semantic attributes from both images as well as mean pooled visual content of videos and added it as a complementary information to the video representation to further improve the quality of caption generation. M^3-VC [54] proposed a multi-model memory network to cater for long term visual-textual dependency and to guide the visual attention.

Even though the above methods have employed deep learning, they have used mean pooled visual features or attention based high level features from CNNs. These features have been used directly in their framework in the language model or by introducing additional unit in the standard framework. We argue that this technique under-utilizes the state of the art CNN features in video captioning framework. We propose features that are rich in visual content and empirically show that this enrichment of visual features alone when combined with a standard and simple language model can outperform existing state of the art methods. Visual features are part of every video captioning framework. Hence, instead of using high level or mean pooled features, building on top of our visual features can further enhance the video captioning frameworks' performances.

3. Proposed Approach

Let $\mathcal{V}$ denote a video that has 'f' frames or 'c' clips. The fundamental task in automatic video captioning is to generate a textual sentence $\mathcal{S} = \{\mathcal{W}_1, \mathcal{W}_2, ..., \mathcal{W}_w\}$ comprising 'w' words that matches closely to human generated captions for the same video. Deep learning based video captioning methods typically define an energy loss function of the following form for this task:

$$\Xi(\mathbf{v}, \mathcal{S}) = -\sum_{t=1}^{w} \log \Pr\left(\mathcal{W}_t | \mathbf{v}, \mathcal{W}_1, ...\mathcal{W}_{t-1}\right), \quad (1)$$

where Pr(.) denotes the probability, and $\mathbf{v} \in \mathbb{R}^d$ is a visual representation of $\mathcal{V}$. By minimizing the cost defined as the Expected value of the energy $\Xi(.)$ over a large corpus of videos, it is hoped that the inferred model $\mathcal{M}$ can automatically generate meaningful captions for unseen videos.

In this formulation, '$\mathbf{v}$' is considered a training input, that makes remainder of the problem a sequence learning task. Consequently, the existing methods in video captioning mainly focus on tailoring RNNs [16] or LSTMs [22] to generate better captions, *assuming* effective visual encoding of $\mathcal{V}$ to be available in the form of '$\mathbf{v}$'. The representation prowess of CNNs has made them the default choice for visual encoding in the existing literature. However, due to the nascency of video captioning research, only primitive methods of using CNN features for '$\mathbf{v}$' can be found in the literature. These methods directly use 2D/3D CNN features or their concatenations for visual encoding, where the temporal dimension of the video is resolved by mean pooling [33, 34, 18].

We acknowledge the role of apt sequence modeling for video description, however, we also argue that designing specialized visual encoding techniques for captioning is equally important. Hence, we mainly focus on the operator $\mathcal{Q}(.)$ in the mapping $\mathcal{M}(\mathcal{Q}(\mathcal{V}))) \rightarrow \mathcal{S}$, where $\mathcal{Q}(\mathcal{V}) \rightarrow \mathbf{v}$. We propose a visual encoding technique that, along harnessing the power of CNN features, explicitly encodes spatio-temporal dynamics of the scene in the visual representation, and embeds semantic attributes in it to further help the sequence modelling phase of video description to generate semantically rich textual sentences.

3.1. Visual Encoding

For clarity, we describe the visual representation of a video $\mathcal{V}$ as $\mathbf{v} = [\boldsymbol{\alpha};\ \boldsymbol{\beta};\ \boldsymbol{\gamma};\ \boldsymbol{\eta}]$, where $\boldsymbol{\alpha}$ to $\boldsymbol{\gamma}$ are themselves column-vectors computed by the proposed technique. We explain these computations in the following.

3.1.1 Encoding Temporal Dynamics

In the context of video description, features extracted from pre-trained 2D-CNNs, e.g. VGG [44] and 3D-CNNs, e.g. C3D [48] have been shown useful for visual encoding of videos. The standard practice is to forward pass individual video frames through a 2D CNN and store activation values of a pre-selected *extraction layer* of the network. Then, perform mean pooling over those activations for all the frames to compute the visual representation. A similar procedure is adopted with 3D CNN with a difference that video clips are used in forward passes instead of frames.

A simple mean pooling operation over activation values is bound to fail in encoding fine-grained temporal dynamics of the video. This is true for both 2D and 3D CNNs, despite the fact that the latter models video clips. We address this shortcoming by defining transformations $\mathcal{T}_f(\mathcal{F}) \rightarrow \boldsymbol{\alpha}$

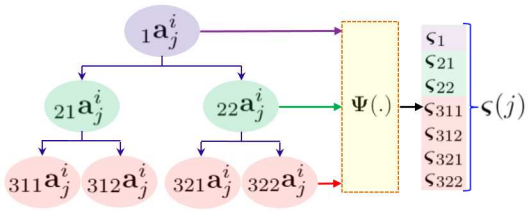

Figure 2. Illustration of hierarchical application of Short Fourier Transform $\mathbf{\Phi}(.)$ to the activations $\mathbf{a}_j^i$ of the j^{th} neuron of the extraction layer for the i^{th} video.

and $\mathcal{T}_c(\mathcal{C}) \rightarrow \boldsymbol{\beta}$, such that $\mathcal{F} = \{\overset{\text{2D}}{\mathbf{a}}_1, \overset{\text{2D}}{\mathbf{a}}_2, ..., \overset{\text{2D}}{\mathbf{a}}_f\}$ and $\mathcal{C} = \{\overset{\text{3D}}{\mathbf{a}}_1, \overset{\text{3D}}{\mathbf{a}}_2, ..., \overset{\text{3D}}{\mathbf{a}}_c\}$. Here, $\overset{\text{2D}}{\mathbf{a}}_t$ and $\overset{\text{3D}}{\mathbf{a}}_t$ denote the activation vectors of the extraction layers of 2D and 3D CNNs for the t^{th} video frame and video clip respectively. The aim of these transformations is to compute $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ that encode temporal dynamics of the *complete* video with high fidelity.

We use the last $avg\ pool$ layer of InceptionResnetV2 [46] to compute $\overset{\text{2D}}{\mathbf{a}}_i$, and the $fc6$ layer of C3D [48] to get $\overset{\text{3D}}{\mathbf{a}}_i$. The transformations $\mathcal{T}_{f/c}(.)$ are defined over the activations of those extraction layers. Below, we explain $\mathcal{T}_f(.)$ in detail. The transformation $\mathcal{T}_c(.)$ is similar, except that it uses activations of clips instead of frames.

Let $a_{j,t}^i$ denote the activation value of the j^{th} neuron of the network's extraction layer for the t^{th} frame of the i^{th} training video. We leave out the superscript 2D for better readability. To perform the transform, we first define $_1\mathbf{a}_j^i = [a_{j,1}^i, a_{j,2}^i, ..., a_{j,f}^i] \in \mathbb{R}^f$ and compute $\Psi(_1\mathbf{a}_j^i) \rightarrow \boldsymbol{\varsigma}_1 \in \mathbb{R}^p$, where the operator $\mathbf{\Psi}(.)$ computes the Short Fourier Transform [31] of the vector in its argument and stores the first 'p' coefficients of the transform. Then, we divide $_1\mathbf{a}_j^i$ into two smaller vectors $_{21}\mathbf{a}_j^i \in \mathbb{R}^h$ and $_{22}\mathbf{a}_j^i \in \mathbb{R}^{h-f}$, where $h = \lfloor \frac{f}{2} \rfloor$. We again apply the operator $\mathbf{\Psi}(.)$ to these vectors to compute $\boldsymbol{\varsigma}_{21}$ and $\boldsymbol{\varsigma}_{22}$ in p-dimensional space. We recursively perform the same operations on $\boldsymbol{\varsigma}_{21}$ and $\boldsymbol{\varsigma}_{22}$ to get the p-dimensional vectors $\boldsymbol{\varsigma}_{311}$, $\boldsymbol{\varsigma}_{312}$, $\boldsymbol{\varsigma}_{321}$, and $\boldsymbol{\varsigma}_{322}$. We combine all these vectors as $\boldsymbol{\varsigma}(j) = [\boldsymbol{\varsigma}_1, \boldsymbol{\varsigma}_{21}, \boldsymbol{\varsigma}_{22}, ..., \boldsymbol{\varsigma}_{322}] \in \mathbb{R}^{(p\times 7)\times 1}$. We also illustrate this operation in Fig. 2. The same operation is performed individually for each neuron of our extraction layer. We then concatenate $\boldsymbol{\varsigma}(j) : j \in \{1, 2, ..., m\}$ to form $\boldsymbol{\alpha} \in \mathbb{R}^{(p\times 7\times m)\times 1}$, where m denotes the number of neurons in the extraction layer. As a result of performing $\mathcal{T}_f(\mathcal{F}) \rightarrow \boldsymbol{\alpha}$, we have computed a representation the video while accounting for fine temporal dynamics in the whole sequence of video frames. Consequently, $\mathcal{T}_f(.)$ results in a much more informative representation than that obtained with mean pooling of the neuron activations.

We define $\mathcal{T}_c(.)$ in a similar manner for the set $\mathcal{C}$ of video clip activations. This transformation results in $\boldsymbol{\beta} \in \mathbb{R}^{(p\times 7\times k)\times 1}$, where k denotes the number of neurons in the extraction layer of the 3D CNN. It is worth mentioning that a 3D CNN is already trained on short video *clips*. Hence, its features account for the temporal dimension of $\mathcal{V}$ to some extent. Nevertheless, accounting for the fine temporal details in the whole video adds to our encoding significantly (see Section 4.3). It is noteworthy that exploiting Fourier Transform in a hierarchical fashion to encode temporal dynamics has also been considered in human action recognition [53, 36]. However, this work is the first to apply Short Fourier Transform hierarchically for video captioning.

3.1.2 Encoding Semantics and Spatial Evolution

It is well-established that the latter layers of CNNs are able to learn features at higher levels of abstraction due to hierarchical application of convolution operations in the earlier layers [28]. The common use of activations of e.g. fully-connected layers as visual features for captioning is also motivated by the fact that these representations are *discriminative transformations of high-level* video features. We take this concept further and argue that the output layers of CNNs can themselves serve as discriminative encodings of the highest abstraction level for video captioning. We describe the technique to effectively exploit these features in the paragraphs to follow. Here, we briefly emphasize that the output layer of a network contains additional information for video captioning beyond what is provided by the commonly used extraction layers of networks, because:

1. The output labels are yet another transformation of the extraction layer features, resulting from network weights that are unaccounted for by extraction layer.

2. The semantics attached to the output layer are at the same level of abstraction that is encountered in video captions - a unique property of the output layers.

We use the output layers of an Object Detector (i.e. YOLO [37]) and a 3D CNN (i.e. C3D [48]) to extract semantics pertaining to the objects and actions recorded in videos. The core idea is to quantitatively embed object labels, their frequencies of occurrence, and evolution of their spatial locations in videos in the visual encoding vector. Moreover, we also aim to enrich our visual encoding with the semantics of actions performed in the video. The details of materializing this concept are presented below.

Objects Information: Different from classifiers that only predict labels of input images/frames, object detectors can localize multiple objects in individual frames, thereby providing cues for ascertaining plurality of the same type of objects in individual frames and evolution of objects' locations in multiple frames. Effective embedding of such high-level information in vector '$\mathbf{v}$' promises descriptions that can clearly differentiate between e.g. 'people running' and 'person walking' in a video.

The sequence modeling component of a video captioning system generates a textual sentence by selecting words from

a large dictionary $\mathcal{D}$. An object detector provides a set $\widetilde{\mathcal{L}}$ of object labels at its output. We first compute $\mathcal{L} = \mathcal{D} \bigcap \widetilde{\mathcal{L}}$, and define $\boldsymbol{\gamma} = [\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, ..., \boldsymbol{\zeta}_{|\mathcal{L}|}]$, where $|.|$ denotes the cardinality of a set. The vectors $\boldsymbol{\zeta}_i, \forall i$ in $\boldsymbol{\gamma}$ are further defined with the help 'q' frames sampled from the original video. We perform this sampling using a fixed time interval between the sampled frames of a given video. The samples are passed through the object detector and its output is utilized in computing $\boldsymbol{\zeta}_i, \forall i$. A vector $\boldsymbol{\zeta}_i$ is defined as $\boldsymbol{\zeta}_i = [\text{Pr}(\ell_i), \text{Fr}(\ell_i), \boldsymbol{\nu}_i^1, \boldsymbol{\nu}_i^2, ..., \boldsymbol{\nu}_i^{(q-1)}]$, where ℓ_i indicates the i^{th} element of $\mathcal{L}$ (i.e. an object name), Pr(.) and Fr(.) respectively compute the probability and frequency of occurrence of the object corresponding to ℓ_i, and $\boldsymbol{\nu_i}^z$ represent the velocity of the object between the frames z and $z+1$ (in the sampled q frames).

We define $\boldsymbol{\gamma}$ over 'q' frames, whereas the used object detector processes individual frames that results in a probability and frequency value for each frame. We resolve this and related mismatches by using the following definitions of the components of $\boldsymbol{\zeta}_i$:

- $\text{Pr}(.) = \max_z \text{Pr}_z(.) : z \in \{1, ..., q\}$.
- $\text{Fr}(.) = \frac{\max_z \text{Fr}_z(.)}{N} : z \in \{1, ..., q\}$, where '$N$' is the allowed maximum number of the same class of objects detected in a frame. We let $N = 10$ in experiments.
- $\nu_i^z = [\delta_x^z, \delta_y^z] : \delta_x^z = \tilde{x}^{z+1} - \tilde{x}^z$ and $\delta_y^z = \tilde{y}^{z+1} - \tilde{y}^z$. Here, $\tilde{x}, \tilde{y}$ denote the Expected values of the x and y coordinates of the same type of objects in a given frame, such that the coordinates are also normalized by the respective frame dimensions.

We let $q = 5$ in our experiments, resulting in $\boldsymbol{\zeta}_i \in \mathbb{R}^{10}, \forall i$ that compose $\boldsymbol{\gamma} \in \mathbb{R}^{(10 \times |\mathcal{L}|) \times 1}$. The indices of coefficients in $\boldsymbol{\gamma}$ identify the object labels in videos (i.e. probable nouns to appear in the description). Unless an object is detected in the video, the coefficients of $\boldsymbol{\gamma}$ corresponding to it are kept zero. The proposed embedding of high level semantics in $\boldsymbol{\gamma}$ contain highly relevant information about objects in explicit form for a sequence learning module of video description system.

Actions Information: Videos generally record objects and their interaction. The latter is best described by the actions performed in the videos. We already use a 3D CNN that learns action descriptors for the videos. We tap into the output layer of that network to further embed high level action information in our visual encoding. To that end, we compute $\mathcal{A} = \widetilde{\mathcal{A}} \bigcap \mathcal{D}$, where $\mathcal{A}$ is the set of labels at the output of the 3D CNN. Then, we define $\boldsymbol{\eta} = \left[[\vartheta_1, \text{Pr}(\ell_1)], [\vartheta_2, \text{Pr}(\ell_2)], ..., [\vartheta_{|\mathcal{A}|}, \text{Pr}(\ell_{|\mathcal{A}|})]\right] \in \mathbb{R}^{(2 \times |\mathcal{A}|) \times 1}$, where ℓ_i is the i^{th} element of $\mathcal{A}$ (an action label) and ϑ is a binary variable that is 1 only if the action is predicted by the network.

We concatenate the above described vectors $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$ and $\boldsymbol{\eta}$ to form our visual encoding vector $\mathbf{v} \in \mathbb{R}^d$, where $d = 2 \times (p \times 7 \times m) + (10 \times |\mathcal{L}|) + (2 \times |\mathcal{A}|)$. Before passing this vector to a sequence modelling component of our method, we perform its compression using a fully connected layer, as shown in Fig. 1. Using $tanh$ activation function and fixed weights, this layer projects '$\mathbf{v}$' to a 2K-dimensional space. The resulting projection '$\boldsymbol{\upsilon}$' is used by our language model.

3.2. Sequence Modelling

We follow the common pipeline of video description techniques that feeds visual representation of a video to a sequence modelling component, see Fig. 1. Instead of resorting to a sophisticated language model, we develop a relatively simpler model employing multiple layers of Gated Recurrent Units (GRUs) [14]. GRUs are known to be more robust to vanishing gradient problem - an issue encountered in long captions - due to their ability of remembering the relevant information and forgetting the rest over time. A GRU has two gates: reset Γ_r and update Γ_u, where the update gate decides how much the unit updates its previous memory and the reset gate determines how to combine the new input with the previous memory. Concretely, our language model computes the hidden state $h^{<t>}$ of a GRU as:

$$\Gamma_u = \sigma(W_u[h^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[h^{<t-1>}, x^{<t>}] + b_r)$$

$$\tilde{h}^{<t>} = \tanh\left(W_h[\Gamma_r \odot h^{<t-1>}, x^{<t>}] + b_h\right.$$

$$h^{<t>} = \Gamma_u \odot \tilde{h}^{<t>} + (1 - \Gamma_u) \odot h^{<t-1>}$$

where, $\odot$ denotes the hadamard product, $\sigma(.)$ is sigmoid activation , $W_q, \forall q$ are learnable weight matrices, and $b_{u/r/h}$ denote the respective biases. In our approach, $h^{<0>} = \boldsymbol{\upsilon}$ for a given video, whereas the signal x is the word embedding vector. In Section 4.3, we report results using two layers of GRUs, and demonstrate that our language model under the proposed straightforward sequence modelling already provides highly competitive performance due to the proposed visual encoding.

4. Experimental Evaluation

4.1. Datasets

We evaluate our technique using two popular benchmark datasets from the existing literature in video description, namely Microsoft Video Description (MSVD) dataset [11], and MSR-Video To Text (MSR-VTT) dataset [57]. We first give details of these datasets and their processing performed in this work, before discussing the experimental results.

MSVD Dataset [11]: This dataset is composed of 1,970 YouTube open domain videos that predominantly show only a single activity each. Generally, each clip is spanning over 10 to 25 seconds. The dataset provides multilingual human annotated sentences as captions for the videos. We experiment with the captions in English. On average, 41 ground truth captions can be associated with a single video. For benchmarking, we follow the common data split of 1,200 training samples, 100 samples for validation and 670 videos for testing [59, 54, 18].

MSR-VTT Dataset [57]: This recently introduced open domain videos dataset contains a wide variety of videos for the captioning task. It consists of 7,180 videos that are transformed into 10,000 clips. The clips are grouped into 20 different categories. Following the common settings [57], we divide the 10,000 clips into 6,513 samples for training, 497 samples for validation and the remaining 2,990 clips for testing. Each video is described by 20 single sentence annotations by Amazon Mechanical Turk (AMT) workers. This is one of the largest clips-sentence pair dataset available for the video captioning task, which is the main reason of choosing this dataset for benchmarking our technique.

4.2. Dataset Processing & Evaluation Metrics

We converted the captions in both datasets to lower case, and removed all punctuations. All the sentences were then tokenized. We set the vocabulary size for MSVD to 9,450 and for MSR-VTT to 23,500. We employed "*fasttext*" [10] word embedding vectors of dimension 300. Embedding vectors of 1,615 words for MSVD and 2,524 words for MSR-VTT were not present in the pretrained set. Instead of using randomly initialized vectors or ignoring the out of vocabulary words entirely in the training set, we generated embedding vectors for these words using character n-grams within the word, and summing the resulting vectors to produce the final vector. We performed dataset specific fine-tuning on the pretrained word embeddings.

In order to compare our technique with the existing methods, we report results on the four most popular metrics, including; Bilingual Evaluation Understudy (BLEU) [35], Metric for Evaluation of Translation with Explicit Ordering (METEOR) [7], Consensus based Image Description Evaluation (CIDEr_D) [49] and Recall Oriented Understudy of Gisting Evaluation (ROUGE_L) [29]. We refer to the original works for the concrete definitions of these metrics. The subscript 'D' in CIDEr indicates the metric variant that inhibits higher values for inappropriate captions in human judgment. Similarly, the subscript 'L' indicates the variant of ROUGE that is based on recall-precision scores of the longest common sequence between the prediction and the ground truth. We used the Microsoft COCO server [12] to compute our results.

4.3. Experiments

In our experiments reported below[1], we use Inception-ResnetV2 (IRV2) [46] as the 2D CNN, whereas C3D [48] is used as the 3D CNN. The last '$avg\ pool$' layer of the former, and the '$fc6$' layer of the latter are considered as the *extraction layers*. The 2D CNN is pre-trained on the popular ImageNet dataset [41], whereas Sports 1M dataset [24] is used for the pre-training of C3D. To process videos, we re-size the frames to match the input dimensions of these networks. For the 3D CNN, we use 16-frame clips as inputs with an 8-frame overlap. YOLO [37] is used as the object detector in all our experiments. To train our language model, we include a start and an end token to the captions to deal with the dynamic length of different sentences. We set the maximum sentence length to 30 words in the case of experiments with MSVD dataset, and to 50 for the MSR-VTT dataset. These length limits are based on the available captions in the datasets. We truncate a sentence if its length exceeds the set limit, and zero pad in the case of shorter length. We tune the hyper-parameters of our language model on the validation set. The results below use two layers of GRUs, that employ 0.5 as the dropout value. We use the RMSProp algorithm with a learning rate 2×10^{-4} to train the models. A batch size of 60 is used for training in our experiments. We performed training of our models for 50 epochs. We used the sparse cross entropy loss to train our model. The training is conducted using NVIDIA Titan XP 1080 GPU. We used TensorFlow framework for development.

4.3.1 Results on MSVD dataset

We comprehensively benchmark our method against the current state-of-the-art in video captioning. We report the results of the existing methods and our approach in Table. 1. For the existing techniques, recent best performing methods are chosen and their results are directly taken from the existing literature (same evaluation protocol is ensured). The table columns present scores for the metrics BLEU-4 (B-4), METEOR (M), CIDEr_D (C) and ROUGE_L (R).

The last seven rows of the Table report results of different variants of our method to highlight the contribution of various components of the overall technique. GRU-MP indicates that we use our two-layer GRU model, while the common 'Mean Pooling (MP)' strategy is adopted to resolve the temporal dimension of videos. 'C3D' and 'IRV2' in the parentheses identify the networks used to compute the visual codes. We abbreviate the joint use of C3D and IRV2 as 'CI'. We use 'EVE' to denote our Enriched Visual Encoding that applies Hierarchical Fourier Transform - indicated by the subscript 'hft' - on the activations of the network extraction layers. The proposed final technique, that

[1]Due to through evaluation, supplementary material also contains further results. Only the best performing setting is discussed here.

Table 1. Benchmarking on MSVD dataset [11] in terms of BLEU-4 (B-4), METEOR (M), CIDEr$_D$ (C) and ROUGE$_L$ (R). See the text for the description of proposed method GRU-EVE's variants.

Model	B-4	M	C	R
FGM [47]	13.7	23.9	-	-
S2VT [50]	-	29.2	-	-
LSTM-YT [51]	33.3	29.1	-	-
Temporal-Attention (TA) [59]	41.9	29.6	51.67	-
h-RNN [60]	49.9	32.6	65.8	-
MM-VDN [56]	37.6	29.0	-	-
HRNE [32]	43.8	33.1	-	-
GRU-RCN [6]	47.9	31.1	67.8	-
LSTM-E [33]	45.3	31.0	-	-
SCN-LSTM [18]	51.1	33.5	77.7	-
DMRM [58]	51.1	33.6	74.8	-
LSTM-TSA [34]	**52.8**	33.5	74.0	-
TDDF [61]	45.8	33.3	73.0	69.7
BAE [8]	42.5	32.4	63.5	-
PickNet [13]	46.1	33.1	76.0	69.2
aLSTMs [19]	50.8	33.3	74.8	-
M^3-IC [54]	**52.8**	33.3	-	-
RecNet$_{local}$ [52]	52.3	34.1	**80.3**	69.8
GRU-MP - (C3D)	28.8	27.7	42.6	61.6
GRU-MP - (IRV2)	41.4	32.3	68.2	67.6
GRU-MP - (CI)	41.0	31.3	61.9	67.6
GRU-EVE$_{\text{hft}}$ - (C3D)	40.6	31.0	55.7	67.4
GRU-EVE$_{\text{hft}}$ - (IRV2)	45.6	33.7	74.2	**69.8**
GRU-EVE$_{\text{hft}}$ - (CI)	47.8	**34.7**	75.8	**71.1**
GRU-EVE$_{\text{hft+sem}}$ - (CI)	47.9	**35.0**	78.1	**71.5**

also incorporates the high-level semantic information - indicated by the subscript '+sem' - is mentioned in the last row of the Table. We also follow the same notational conventions for our method in the remaining Tables.

Our method achieves a strong 35 value of METEOR, which provides a $\frac{35.0-34.1}{34.1} \times 100 = 2.64\%$ gain over the closest competitor. Similarly, gain over the current state-of-the-art for ROUGE$_L$ is 2.44%. For the other metrics, our scores remain competitive to the best performing methods. It is emphasized, that our approach derives its main strength from the visual encoding part in contrast to sophisticated language model, which is generally the case for the existing methods. Naturally, complex language models entail difficult and computationally expensive training process, which is not a limitation of our approach.

We illustrate representative qualitative results of our method in Fig. 3. We abbreviate our final approach as 'GRU-EVE' in the figure for brevity. The semantic details and accuracy of e.g. plurality, nouns and verbs is clearly visible in the captions generated by the proposed method. The figure also reports the captions for GRU-MP-(CI) and GRU-EVE$_{\text{hft}}$-(CI) to show the difference resulting from hierarchical Fourier transform (hft) as compared to the Mean Pooling (MP) strategy. These captions justify the noticeable gain achieved by the proposed hft over the traditional MP

Table 2. Performance comparison with single 2D-CNN based methods on MSVD dataset [11].

Model	METEOR
FGM [47]	23.90
S2VT [50]	29.2
LSTM-YT [51]	29.07
TA [59]	29.0
p-RNN [60]	31.1
HRNE [32]	33.1
BGRCN [6]	31.70
MAA [17]	31.80
RMA [23]	31.90
LSTM-E [33]	29.5
M^3-inv3 [54]	32.18
mGRU [62]	33.39
GRU-EVE$_{\text{hft}}$-(IRV2)	**33.7**

Table 3. Performance comparison on MSVD dataset [11] with the methods using multiple features. The scores of existing methods are taken from [54]. V denotes VGG19, C is C3D, I$_v$ denotes Inception-V3, G is GoogleNet and I denotes InceptionResNet-V2

Model	METEOR
SA-G-3C [59]	29.6
S2VT-RGB-Flow [50]	29.8
LSTM-E-VC [33]	31.0
p-RNN-VC [60]	32.6
M^3-I$_v$C [54]	33.3
GRU-EVE$_{\text{hft+sem}}$ - (CI)	**35.0**

in Table 1. We also observe in the table that our method categorically outperforms the mean pool based methods, i.e. LSTM-YT [51], LSTM-E [33], SCN-LSTM [18], and LSTM-TSA[34] on METEOR, CIDEr and ROUGE$_L$. Under these observations, we safely recommend the proposed hierarchical Fourier transformation as the substitute for the 'mean pooling' in video captioning.

In Table 2, we compare the variant of our method based on a single CNN with the best performing single CNN based existing methods. The results are directly taken from [54] for the provided METEOR metric. As can be seen, our method outperforms all these methods. In Table 3, we also compare our method on METEOR with the state-of-the-art methods that necessarily use multiple visual features to obtain the best performance. A significant 5.1% gain is achieved by our method to the closest competitor in this regard.

4.3.2 Results on MSR-VTT dataset

MSR-VTT [57] is a recently released dataset. We compare performance of our approach on this dataset with the latest published models such as Alto [42], RUC-UVA [15], TDDF [61], PickNet [13], M^3-VC [54] and RecNet$_{local}$ [52]. The results are summarized in Table 4. Similar to the MSVD dataset, our method significantly improves the state-of-the-art on this dataset on METEOR and

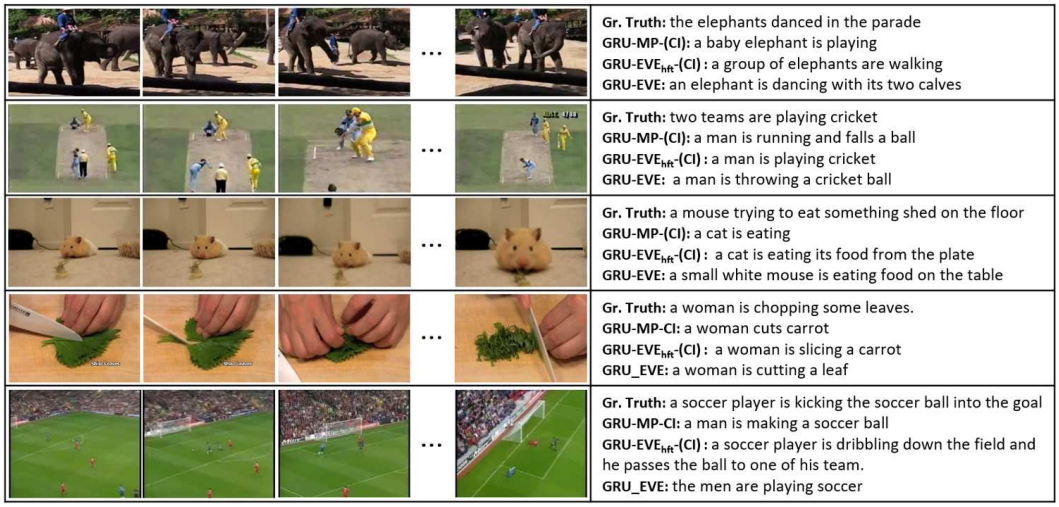

Figure 3. Illustration of caption generated for MSVD test set: The final approach is abbreviated as GRU-EVE for brevity. A sentence from ground truth captions is shown for reference.

Table 4. Benchmarking on MSR-VTT dataset [57] in terms of BLEU-4 (B-4), METEOR (M), CIDEr$_D$ (C) and ROUGE$_L$ (R).

Model	B-4	M	C	R
Alto [42]	39.8	26.9	45.7	59.8
RUC-UVA [15]	38.7	26.9	45.9	58.7
TDDF [61]	37.3	27.8	43.8	59.2
PickNet [13]	38.9	27.2	42.1	59.5
M^3-VC [54]	38.1	26.6	-	-
RecNet$_{local}$ [52]	39.1	26.6	42.7	59.3
GRU-EVE$_{hft}$ - (IRV2)	32.9	26.4	39.2	57.2
GRU-EVE$_{hft}$ - (CI)	36.1	27.7	45.2	59.9
GRU-EVE$_{hft+sem}$ - (CI)	38.3	**28.4**	**48.1**	**60.7**

ROUGE$_L$ metrics, while achieving strong results on the remaining metrics. These result ascertain the effectiveness of the proposed enriched visual encoding for visual captioning. We provide examples of qualitative results on this dataset in the supplementary material of the paper.

5. Discussion

We conducted a through empirical evaluation of the proposed method to explore its different aspects. Below we discuss and highlight few of these aspects in the text. Where necessary, we also provide results in the supplementary material of the paper to back the discussion.

For the settings discussed in the previous section, we generally observed semantically rich captions generated by the proposed approach. In particular, these captions well captured the plurality of objects and their motions/actions. Moreover, the captions generally described the whole videos instead of its partial clips. Instead of only two, we also tested different number of GRU layers, and observed that increasing the number of GRU layers deteriorated the BLEU-4 score. However, there were improvements in all the remaining metrics. We retained only two GRU layers in the final method mainly for computational gains. Moreover, we also tested different architectures of GRU, e.g. with state sizes 512, 1024, 2048 and 4096. We observed a trend of performance improvement until 2048 states. However, further states did not improve the performance. Hence, 2048 were finally used in the results reported in the previous section.

Whereas all the components of the proposed technique contributed to the overall final performance, the biggest revelation of our work is the use of hierarchical Fourier Transform to capture the temporal dynamics of videos. As compared to the 'nearly standard' mean pooling operation performed in the existing captioning pipeline, the proposed use of Fourier Transform promises a significant performance gain for any method. Hence, we safely recommend replacing the mean pooling operation with our transformation for the future techniques.

6. Conclusion

We presented a novel technique for visual encoding of videos to generate semantically rich captions. Besides capitalizing on the representation power of CNNs, our method explicitly accounts for the spatio-temporal dynamics of the scene, and high-level semantic concepts encountered in the video. We applying Short Fourier Transform to 2D and 3D CNN features of the videos in a hierarchical manner, and account for the high-level semantics by processing output layer features of an Object Detector and the 3D CNN. Our enriched visual representation is used to learn a relatively simple GRU based language model that performs on-par or better than the existing video description methods on popular MSVD and MSR-VTT datasets.

Acknowledgment This research was supported by ARC Discovery Grant DP160101458 and partially by DP190102443. The Titan XP GPU used in our experiments was donated by NVIDIA corporation.

References

[1] N. Aafaq, A. Mian, W. Liu, S. Z. Gilani, and M. Shah. Video description: A survey of methods, datasets and evaluation metrics. *arXiv preprint arXiv:1806.00186*, 2018. 1

[2] J.-B. Alayrac, P. Bojanowski, N. Agrawal, J. Sivic, I. Laptev, and S. Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *IEEE CVPR*, 2016. 1

[3] B. Andrei, E. Georgios, H. Daniel, M. Krystian, N. Siddharth, X. Caiming, and Z. Yibiao. A Workshop on Language and Vision at CVPR 2015. 1

[4] B. Andrei, M. Tao, N. Siddharth, Z. Quanshi, S. Nishant, L. Jiebo, and S. Rahul. A Workshop on Language and Vision at CVPR 2018. http://languageandvision.com/. 1

[5] R. Anna, T. Atousa, R. Marcus, P. Christopher, L. Hugo, C. Aaron, and S. Bernt. The Joint Video and Language Understanding Workshop at ICCV 2015. 1

[6] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016. 7

[7] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 6

[8] L. Baraldi, C. Grana, and R. Cucchiara. Hierarchical boundary-aware neural encoder for video captioning. In *IEEE CVPR*, 2017. 7

[9] A. Barbu, A. Bridge, Z. Burchill, D. Coroian, S. Dickinson, S. Fidler, A. Michaux, S. Mussman, S. Narayanaswamy, D. Salvi, et al. Video in sentences out. In *UAI*, 2012. 1, 2

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *In TACL*, pages 135–146, 2017. 6

[11] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL: Human Language Technologies-Volume 1*, pages 190–200. ACL, 2011. 2, 5, 6, 7

[12] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 6

[13] Y. Chen, S. Wang, W. Zhang, and Q. Huang. Less is more: Picking informative frames for video captioning. In *ECCV*, 2018. 7, 8

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 1, 5

[15] J. Dong, X. Li, W. Lan, Y. Huo, and C. G. Snoek. Early embedding and late reranking for video captioning. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1082–1086. ACM, 2016. 7, 8

[16] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 1, 3

[17] R. Fakoor, A.-r. Mohamed, M. Mitchell, S. B. Kang, and P. Kohli. Memory-augmented attention modelling for videos. *arXiv preprint arXiv:1611.02261*, 2016. 7

[18] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic Compositional Networks for visual captioning. In *IEEE CVPR*, 2017. 1, 2, 3, 6, 7

[19] L. Gao, Z. Guo, H. Zhang, X. Xu, and H. T. Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017. 7

[20] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2712–2719, 2013. 2

[21] P. Hanckmann, K. Schutte, and G. J. Burghouts. Automated textual descriptions for a wide range of video events with 48 human actions. In *ECCV*, pages 372–380, 2012. 2

[22] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 3

[23] A. K. Jain, A. Agarwalla, K. K. Agrawal, and P. Mitra. Recurrent memory addressing for describing videos. In *CVPR Workshops*, 2017. 7

[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 6

[25] M. U. G. Khan, L. Zhang, and Y. Gotoh. Human focused video description. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011. 2

[26] A. Kojima, T. Tamura, and K. Fukunaga. Natural language description of human activities from video images based on concept hierarchy of actions. *IJCV*, 50(2):171–184, 2002. 1, 2

[27] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI*, volume 1, page 2, 2013. 2

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015. 1, 4

[29] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004. 6

[30] M. Margaret, M. Ishan, H. Ting-Hao, and F. Frank. Story Telling Workshop and Visual Story Telling Challenge at NAACL 2018. 1

[31] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999. 1, 4

[32] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *IEEE CVPR*, pages 1029–1038, 2016. 7

[33] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In

Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4594–4602, 2016. 1, 3, 7

[34] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *IEEE CVPR*, 2017. 1, 2, 3, 7

[35] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on ACL*, pages 311–318, 2002. 6

[36] H. Rahmani and A. Mian. 3d action recognition from novel viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1506–1515, 2016. 4

[37] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *IEEE CVPR*, 2017. 2, 4, 6

[38] A. Rohrbach, M. Rohrbach, W. Qiu, A. Friedrich, M. Pinkal, and B. Schiele. Coherent multi-sentence video description with variable level of detail. In *German Conference on Pattern Recognition*, 2014. 1

[39] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *IJCV*, 123(1):94–120, 2017. 1

[40] M. Rohrbach, W. Qiu, I. Titov, S. Thater, M. Pinkal, and B. Schiele. Translating video content to natural language descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 433–440, 2013. 2

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 6

[42] R. Shetty and J. Laaksonen. Frame-and segment-level features and candidate pool evaluation for video caption generation. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1073–1076. ACM, 2016. 7, 8

[43] A. Shin, K. Ohnishi, and T. Harada. Beyond caption to narrative: Video captioning with multiple sentences. In *IEEE International Conference on Image Processing (ICIP)*, 2016. 1

[44] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 3

[45] J. Song, L. Gao, L. Liu, X. Zhu, and N. Sebe. Quantization-based hashing: a general framework for scalable image and video retrieval. *Pattern Recognition*, 75:175–187, 2018. 1

[46] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017. 2, 4, 6

[47] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *Coling*, volume 2, page 9, 2014. 7

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1, 2, 3, 4, 6

[49] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *IEEE CVPR*, 2015. 6

[50] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE international conference on computer vision*, pages 4534–4542, 2015. 1, 3, 7

[51] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, pages 1494–1504, 2015. 1, 3, 7

[52] B. Wang, L. Ma, W. Zhang, and W. Liu. Reconstruction network for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7622–7631, 2018. 2, 7, 8

[53] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):914–927, 2014. 4

[54] J. Wang, W. Wang, Y. Huang, L. Wang, and T. Tan. M3: Multimodal memory modelling for video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7512–7520, 2018. 2, 3, 6, 7, 8

[55] J. Wang, T. Zhang, N. Sebe, H. T. Shen, et al. A survey on learning to hash. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):769–790, 2018. 1

[56] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko. A multi-scale multiple instance video description network, A Workshop on Closing the Loop Between Vision and Language at ICCV 2015. 7

[57] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE CVPR*, 2016. 1, 2, 5, 6, 7, 8

[58] Z. Yang, Y. Han, and Z. Wang. Catching the temporal regions-of-interest for video captioning. In *25th ACM Multimedia*, pages 146–153, 2017. 7

[59] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proceedings of the IEEE international conference on computer vision*, pages 4507–4515, 2015. 3, 6, 7

[60] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *IEEE CVPR*, 2016. 1, 7

[61] X. Zhang, K. Gao, Y. Zhang, D. Zhang, J. Li, and Q. Tian. Task-driven dynamic fusion: Reducing ambiguity in video description. In *IEEE CVPR*, 2017. 7, 8

[62] L. Zhu, Z. Xu, and Y. Yang. Bidirectional multirate reconstruction for temporal modeling in videos. In *IEEE CVPR*, pages 2653–2662, 2017. 7