Attend and Interact: Higher-Order Object Interactions for Video Understanding

Chih-Yao Ma*[1], Asim Kadav[2], Iain Melvin[2], Zsolt Kira[3], Ghassan AlRegib[1], and Hans Peter Graf[2]

[1]Georgia Institute of Technology, [2]NEC Laboratories America, [3]Georgia Tech Research Institute

Abstract

Human actions often involve complex interactions across several inter-related objects in the scene. However, existing approaches to fine-grained video understanding or visual relationship detection often rely on single object representation or pairwise object relationships. Furthermore, learning interactions across multiple objects in hundreds of frames for video is computationally infeasible and performance may suffer since a large combinatorial space has to be modeled. In this paper, we propose to efficiently learn higher-order interactions between arbitrary subgroups of objects for fine-grained video understanding. We demonstrate that modeling object interactions significantly improves accuracy for both action recognition and video captioning, while saving more than 3-times the computation over traditional pairwise relationships. The proposed method is validated on two large-scale datasets: Kinetics and ActivityNet Captions. Our SINet and SINet-Caption achieve state-of-the-art performances on both datasets even though the videos are sampled at a maximum of 1 FPS. To the best of our knowledge, this is the first work modeling object interactions on open domain large-scale video datasets, and we additionally model higher-order object interactions which improves the performance with low computational costs.

1. Introduction

Video understanding tasks such as activity recognition and caption generation are crucial for various applications in surveillance, video retrieval, human behavior understanding, etc. Recently, datasets for video understanding such as Charades [42], Kinetics [21], and ActivityNet Captions [22] contain diverse real-world examples and represent complex human and object interactions that can be difficult to model with state-of-the-art video understanding methods [42]. Consider the example in Figure 1. To accurately predict *cooking on campfire* and *cooking egg* among other similar action classes requires understanding of fine-grained object relationships and interactions. For example, a hand breaks an egg, eggs are in a bowl, the bowl is on top of the campfire, campfire is a fire built with wood at a camp, etc. Although recent state-of-the-art approaches for action recognition have demonstrated significant improvements over datasets such as UCF101 [45], HMDB51 [23], Sports-1M [20], THUMOS [18], ActivityNet [5], and YouTube-8M [1], they often focus on representing the overall visual scene (coarse-grained) as sequence of inputs that are combined with temporal pooling, e.g. CRF, LSTM, 1D Convolution, attention, and NetVLAD [4, 29, 30, 41], or use 3D Convolution for the whole video sequence [6, 37, 46]. These approaches ignore the fine-grained details of the scene and do not infer interactions between various objects in the video. On the other hand, in video captioning tasks, although prior approaches use spatial or temporal attention to selectively attend to fine-grained visual content in both space and time, they too do not model object interactions.

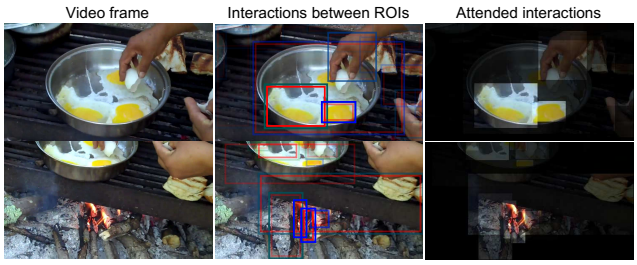

Figure 1. *Higher-order object interactions* are progressively detected based on selected inter-relationships. ROIs with the same color (weighted r, g, b) indicating there exist inter-object relationships, e.g. eggs in the same bowl, hand breaks egg, and bowl on top of campfire (interaction within the same color). Groups of inter-relationships then jointly model higher-order object interaction of the scene (interaction between different colors). *Right*: ROIs are highlighted with their attention weights for higher-order interactions. The model further reasons about the interactions through time and predicts *cooking on campfire* and *cooking egg*. Images are generated from SINet (best viewed in color).

*Work performed as a NEC Labs intern

Prior work in understanding visual relationships in the image domain has recently emerged as a prominent research problem, e.g. scene graph generation [27, 53] and visual relationship detection [7, 8, 14, 17, 58, 59]. However, it is unclear how these techniques can be adapted to open-domain video tasks, given that the video is intrinsically more complicated in terms of temporal reasoning and computational demands. More importantly, a video may consist of a large number of objects over time. Prior approaches on visual relationship detection typically model the full pairwise (or triplet) relationships. While this may be realized for images, videos often contain hundreds or thousands of frames. Learning relationships across multiple objects alongside the temporal information is computationally infeasible on modern GPUs, and performance may suffer due to the fact that a finite-capacity neural network is used to model a large combinatorial space. Furthermore, prior work in both image and video domains [31, 32] often focus on pairwise relationships or interactions, where interactions over groups of interrelated objects—*higher-order interactions*—are not explored, as shown in Figure 2.

Toward this end, we present a generic recurrent module for fine-grained video understanding, which dynamically discovers higher-order object interactions via an efficient dot-product attention mechanism combined with temporal reasoning. Our work is applicable to various open domain video understanding problems. In this paper, we validate our method on two video understanding tasks with new challenging datasets: action recognition on Kinetics [21] and video captioning on ActivityNet Captions [22] (with ground truth temporal proposals). By combining both coarse- and fine-grained information, our **SINet** (Spatiotemporal Interaction Network) for action recognition and **SINet-Caption** for video captioning achieve state-of-the-art performance on both tasks while using RGB video frames sampled at only maximum 1 FPS. To the best of our knowledge, this is the first work of modeling object interactions on open domain large-scale video datasets, and we also show that modeling higher-order object interactions can further improve the performance at low computational costs.

2. Related work

We discuss existing work on video understanding based on action recognition and video captioning as well as related work on detecting visual relationships in images and videos.

Action recognition: Recent work on action recognition using deep learning involves learning compact (coarse) representations over time and use pooling or other aggregation methods to combine information from each video frame, or even across different modalities [10, 13, 30, 41, 43]. The representations are commonly obtained directly from forward passing a single video frame or a short video snippet to a 2D ConvNet or 3D ConvNet [6, 37, 46]. Another branch of work uses Region Proposal Networks (RPNs) to jointly train action detection models [15, 25, 36]. These methods use an RPN to extract object features (ROIs), but they do not model or learn interactions between objects in the scene. Distinct from these models, we explore human action recognition task using coarse-grained context information and fine-grained higher-order object interactions. Note that we focus on modeling object interactions for understanding video in a fine-grained manner and we consider other modalities, e.g. optical flow and audio information, to be complementary to our method.

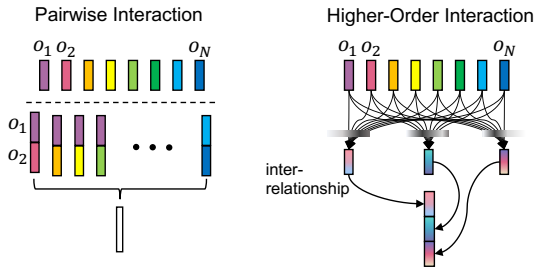

Figure 2. Typically, object interaction methods focus on pairwise interactions (left). We efficiently model the *higher-order interactions* between arbitrary subgroups of objects for video understanding, in which the inter-object relationships in one group are detected and objects with significant relationships (i.e. those that serve to improve action recognition or captioning in the end) are attentively selected (right). The higher-order interaction between groups of selected object relationships are then modeled after concatenation.

Video captioning: Similar to other video tasks using deep learning, initial work on video captioning learn compact representations combined over time. This single representation is then used as input to a decoder, e.g. LSTM, at the beginning or at each word generation to generate a caption for the target video [33, 49, 50]. Other work additionally uses spatial and temporal attention mechanisms to selectively focus on visual content in different space and time during caption generation [38, 44, 54, 55, 57]. Similar to using spatial attention during caption generation, another line of work has additionally incorporated semantic attributes [11, 34, 40, 56]. However, these semantic or attribute detection methods, with or without attention mechanisms, do not consider object relationships and interactions, i.e. they treat the detected attributes as a bag of words. Our work, SINet-Caption uses higher-order object relationships and their interactions as visual cues for caption generation.

Interactions/Relationships in images: Recent advances in detecting visual relationships in images use separate branches in a ConvNet to explicitly model objects, humans, and their interactions [7, 14]. Visual relationships can also be realized by constructing a scene graph which uses a structured representation for describing object relationships and their attributes [19, 26, 27, 53]. Other work on detecting

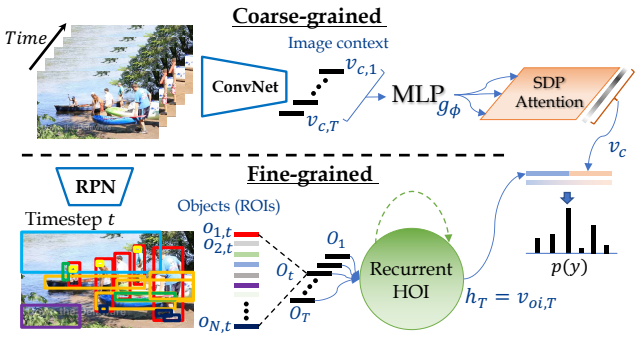

Figure 3. Overview of the SINet for action recognition. **Coarse-grained:** each video frame is encoded into a feature vector $v_{c,t}$. The sequence of vectors are then pooled via temporal SDP-Attention into single vector representation v_c. **Fine-grained:** Each object (ROI) obtained from RPN is encoded in a feature vector $o_{n,t}$. We detect the higher-order object interaction using the proposed generic recurrent Higher-Order Interaction (HOI) module. Finally, coarse-grained (image context) and fine-grained (higher-order object interactions) information are combined to perform action prediction.

visual relationships explore relationships by pairing different objects in the scene [8, 17, 39, 58]. While these models can successfully detect visual relationships for images, a scene with many objects may have only a few individual interacting objects. It would be inefficient to detect all relationships across all individual object pairs [59], making these methods intractable for the video domain.

Interactions/Relationships in videos: Compared to the image domain, there is limited work in exploring relationships for video understanding. Ni et al. [31] use a probabilistic graphical model to track interactions, but their model is insufficient to model interactions involving multiple objects. To overcome this issue, Ni et al. [32] propose using a set of LSTM nodes to incrementally refine the object detections. In contrast, Lea et al. [24] propose to decompose the input image into several spatial units in a feature map, which then captures the object locations, states, and their relationships using shared ConvNets. However, due to lack of appropriate datasets, existing work focuses on indoor or cooking settings where the human subject along with the objects being manipulated are at the center of the image. Also, these methods only handle pairwise relationships between objects. However, human actions can be complex and often involve higher-order object interactions. Therefore, we propose to attentively model object inter-relationships and discover the higher-order interactions on large-scale and open domain videos for fine-grained understanding.

3. Model

Despite the recent successes in video understanding, there has been limited progress in understanding relationships and interactions that occur in videos in a fine-grained manner. To do so, methods must not only understand the high-level video representations but also be able to explicitly model the relationships and interactions between objects in the scene. Toward this end, we propose to exploit both overall image context (coarse) and higher-order object interactions (fine) in the spatiotemporal domain for general video understanding tasks.

In the following section, we first describe the **SINet** on action recognition followed by extending it to **SINet-Caption** for the video captioning task.

3.1. Action Recognition Model

3.1.1 Coarse-grained image context

As recent studies have shown, using LSTM to aggregate a sequence of image representations often results in limited performance since image representations can be similar to each other and thus lack temporal variances [1, 21, 29]. As shown in Figure 3 (top), we thus begin by attending to key image-level representations to summarize the whole video sequence via the Scale Dot-Product Attention (SDP-Attention) [47]:

$$\alpha_c = softmax(\frac{{X_c}^\top X_c}{\sqrt{d_\phi}}), \quad X_c = g_\phi(V_c) \tag{1}$$

$$v_c = \overline{\alpha_c \, {X_c}^\top} \tag{2}$$

where V_c is a set of image features: $V_c = \{v_{c,1}, v_{c,2}, ..., v_{c,T}\}, v_{c,t} \in \mathbb{R}^m$ is the image feature representation encoded via a ConvNet at time t, and t ranges from $\{1, 2, ..., T\}$ for a given video length. g_ϕ is a Multi-Layer Perceptron (MLP) with parameter ϕ, d_ϕ is the dimension of last fully-connected (FC) layer of g_ϕ, $X_c \in \mathbb{R}^{d_\phi \times T}$ is the projected image feature matrix, $\sqrt{d_\phi}$ is a scaling factor, and $\alpha_c \in \mathbb{R}^{T \times T}$ is an attention weight applied to the (projected) sequence of image representations V_c. The weighted image representations are then mean-pooled to form video representation v_c.

3.1.2 Fine-grained higher-order object interactions

Traditional pairwise object interactions only consider how each object interacts with another object. We instead model inter-relationships between arbitrary subgroups of objects, the members of which are determined by a learned attention mechanism, as illustrated in Figure 2. Note that this covers pair-wise or triplet object relationships as a special case, in which the learned attention only focus on one single object.

Problem statement: We define *objects* to be a certain region in the scene that might be used to determine the visual relationships and interactions. Each object representation can be directly obtained from an RPN and further encoded into an object feature. Note that we do not encode object class information from the detector into the feature

representation since there exists a cross-domain problem, and we may miss some objects that are not detected by the pre-trained object detector. Also, we do not know the corresponding objects across time since linking objects through time can be computationally expensive for long videos. As a result, we have variable-lengths of object sets residing in a high-dimensional space that spans across time. Our objective is to efficiently detect higher-order interactions from these rich yet unordered object representation sets across time.

In the simplest setting, an interaction between objects in the scene can be represented via summation operation of individual object information. For example, one method is to add the learnable representations and project these representations into a high-dimensional space where the object interactions can be exploited by simply summing up the object representations. Another approach which has been widely used with images is by pairing all possible object candidates (or subject-object pairs) [7, 8, 17, 39, 58]. However, this is infeasible for video, since a video typically contains hundreds or thousands of frame and the set of object-object pairs is too large to fully represent. Detecting object relationships frame by frame is computationally expensive, and the temporal reasoning of object interactions is not used.

Recurrent Higher-Order Interaction (HOI): To overcome these issues, we propose a generic recurrent module for detecting higher-order object interactions for fine-grained video understanding problems, as shown in Figure 4. The proposed recurrent module dynamically selects object candidates which are important to discriminate the human actions. The combinations of these objects are then concatenated to model higher order interaction using group to group or triplet groups of objects.

First, we introduce learnable parameters for the incoming object features via MLP projection g_{θ_k}, since the object features are pre-trained from another domain and may not necessarily present interactions towards action recognition. The projected object features are then combined with overall image content and previous object interaction to generate K sets of weights to select K groups of objects [1]. Objects with inter-relationships are selected from an attention weight, which generates a probability distribution over all object candidates. The attention is computed using inputs from current (projected) object features, overall image visual representation, and previously discovered object interactions (see Figure 4), which provide the attention mechanism with maximum context.

$$\alpha_k = Attention(g_{\theta_k}(O_t), v_{c,t}, h_{t-1}) \quad (3)$$

where the input O_t is a set of *objects*: $O_t =$

[1]The number K depends on the complexity of the visual scene and the requirement of the task (in this case, action recognition). We leave dynamically selecting K to future work.

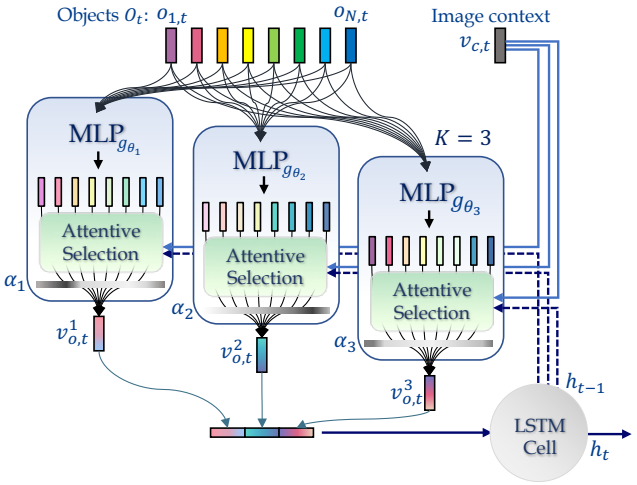

Figure 4. **Recurrent Higher-Order Interaction** module dynamically selects K groups of arbitrary objects with detected inter-object relationships via learnable attention mechanism. This attentive selection module uses the overall image context representation $v_{c,t}$, current set of (projected) objects O_t, and previous object interactions h_{t-1} to generate k^{th} weights α_k for k^{th} selections. The higher-order interaction between groups of selected objects is then modeled via concatenation and the following LSTM cell.

$\{o_{1,t}, o_{2,t}, ..., o_{N,t}\}, o_{n,t} \in \mathbb{R}^m$ is the n^{th} object feature representation at time t. The g_{θ_k} is a MLP with parameter θ_k, the parameters are learnable synaptic weights shared across all objects $o_{n,t}$ and through time t. $v_{c,t}$ denotes as encoded image feature at current time t, and h_{t-1} is the previous output of LSTM cell which represents the previous discovered object interaction. Formally, given an input sequence, a LSTM network computes the hidden vector sequences $\mathbf{h} = (h_1, h_2, ..., h_T)$. Lastly, α_k is an attention weight computed from the proposed attention module.

Attentive selection module: Here we discuss two possible choices for the attention module, as shown in Figure 5. Dot-product attention considers inter-relationships when selecting the objects, and α-attention does not.

- Dot-product attention: In order to model higher-order interactions, which models inter-object relationships in each group of selected objects, we use dot-product attention since the attention weights computed for each object is the combination of all objects.

Formally, the current image representation $v_{c,t}$ and the last object interaction representation h_{t-1} are first projected to introduce learnable weights. The projected $v_{c,t}$ and h_{t-1} are then repeated and expanded N times (the number of objects in O_t). We directly combine this information with projected objects via matrix addition and use it as input to dot-product attention. We added a scale factor as in [47]. The input to the first matrix multiplication and the attention weights over all objects can be defined as:

$$X_k = repeat(W_{h_k} h_{t-1} + W_{c_k} v_{c,t}) + g_{\theta_k}(O_t) \quad (4)$$

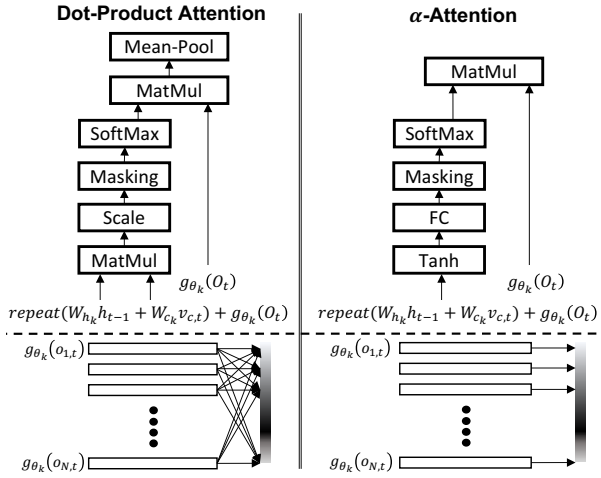

Figure 5. Attention modules: dot-product attention and α-attention. Both attention mechanisms take input from overall image representation $v_{c,t}$, current set of objects O_t, and previous object interactions h_{t-1} computed from LSTM cell at time $t-1$.

$$\alpha_k = softmax(\frac{{X_k}^\top X_k}{\sqrt{d_\theta}}) \tag{5}$$

where $W_{h_k} \in \mathbb{R}^{d_\theta \times d_h}$ and $W_{c_k} \in \mathbb{R}^{d_\theta \times d_{v_{c,t}}}$ are learned weights for h_{t-1} and $v_{c,t}$, d_θ is the dimension of last fully-connected layer of g_{θ_k}, $X_k \in \mathbb{R}^{d_\theta \times N}$ is the input to k^{th} attention module, and $\sqrt{d_\theta}$ is a scaling factor, $\alpha_k \in \mathbb{R}^{N \times N}$ is the computed k^{th} attention. We omit the bias term for simplicity. The attended object feature at time t is then calculated as mean-pooling on weighted objects:

$$v_{o,t}^k = \overline{\alpha_k \left(g_{\theta_k}(O_t)\right)^\top} \tag{6}$$

where the output $v_{o,t}^k$ is a single feature vector representation which encodes the k^{th} object inter-relationships of a video frame at time t.

- α-attention: The α-attention uses the same input format as dot-product attention, but the attention is computed using a *tanh* function and a fully-connected layer:

$$\alpha_k = softmax({w_k}^\top tanh(X_k)) \tag{7}$$

where $w_k \in \mathbb{R}^{d_\theta}$ is a learned weight, and $\alpha_k \in \mathbb{R}^{1 \times N}$ is the computed k^{th} attention. The attended object feature at time t is then calculated as a convex combination:

$$v_{o,t}^k = \sum_n \alpha_{k_n} (g_{\theta_k}(o_{n,t})) \tag{8}$$

We use the α-attention as a baseline to show how considering the inter-relationships of objects (dot-product attention) can further improve the accuracy when ROIs are selected separately.

Finally, for both attention mechanisms, the selected object candidates $v_{o,t}^k$ are then concatenated and used as the input to a LSTM cell. The output $v_{oi,t}$ is then defined as the higher-order object interaction representation at current time t.

$$v_{oi,t} = LSTMCell(v_{o,t}^1 \| v_{o,t}^2 \| ... \| v_{o,t}^K) \tag{9}$$

where $\|$ denotes concatenation between feature vectors. The last hidden state of the LSTM cell $h_T = v_{oi,T}$ is the representation of overall object interactions for the entire video sequence.

Note that by concatenating selected inter-object relationships into a single higher-order interaction representation, the selective attention module tends to select different groups of inter-relationships, since concatenating duplicate inter-relationships does not provide extra information and will be penalized. For an analysis of what inter-relationships are selected, please refer to the supplement.

3.1.3 Late fusion of coarse and fine

Finally, the attended context information v_c obtained from the image representation provides coarse-grained understanding of the video, and the object interactions discovered through the video sequences $v_{oi,T}$ provide fine-grained understanding of the video. We concatenate them as the input to the last fully-connected layer, and train the model jointly to make a final action prediction.

$$p(y) = softmax(W_p(v_c \| v_{oi,T}) + b_p) \tag{10}$$

where $W_p \in \mathbb{R}^{d_y \times (d_{v_c} + d_{v_{oi,T}})}$ and $b_p \in \mathbb{R}^{d_y}$ are learned weights and biases.

3.2. Video Captioning Model

We now describe how SINet can be extended from sequence-to-one to a sequence-to-sequence problem for video captioning. Our goal in providing fine-grained information for video captioning is that, for each prediction of the word, the model is aware of the past generated word, previous output, and the summary of the video content. At each word generation, it has the ability to selectively attend to various parts of the video content in both space and time, as well as to the detected object interactions.

Our **SINet-Caption** is inspired by prior work using hierarchical LSTM for captioning tasks [2, 44], and we extend and integrate it with SINet so that the model can leverage the detected higher-order object interactions. We use a two-layered LSTM integrated with the coarse- and fine-grained information, as shown in Figure 6. The two LSTM layers are: Attention LSTM and Language LSTM. The Attention LSTM identifies which part of the video in spatiotemporal feature space is needed for Language LSTM to generate the next word. Different from prior work, which applied attention directly over all image patches in the entire video [55],

i.e. attended to objects individually, our attentive selection module attends to object interactions while considering their temporal order.

Attention LSTM: The Attention LSTM fuses the previous hidden state output of Language LSTM $h^2_{t_w-1}$, overall representation of the video, and the input word at time t_w-1 to generate the hidden representation for the following attention module. Formally, the input to Attention LSTM can be defined as:

$$x^1_{t_w} = h^2_{t_w-1} \parallel \overline{g_\phi(V_c)} \parallel W_e \Pi_{t_w-1} \tag{11}$$

where $\overline{g_\phi(V_c)}$ is the projected and mean-pooled image features, g_ϕ is a MLP with parameters ϕ, $W_e \in \mathbb{R}^{E\times\Sigma}$ is a word embedding matrix for a vocabulary of size Σ, and Π_{t_w-1} is one-hot encoding of the input word at time t_w-1. Note that t is the video time, and t_w is the timestep for each word generation.

Temporal attention module: We adapt the same α-attention module as shown in Figure 5 to attend over projected image features $g_\phi(V_c)$. The two types of input for this temporal attention module are from outputs of the Attention LSTM and projected image features.

$$X_a = repeat(W_h h^1_{t_w}) + W_c g_\phi(V_c) \tag{12}$$

where $h^1_{t_w}$ is the output of Attention LSTM, $W_h \in \mathbb{R}^{d_\phi \times d_{h^1_{t_w}}}$ and $W_c \in \mathbb{R}^{d_\phi \times d_\phi}$ are learned weights for $h^1_{t_w}$ and $g_\phi(V_c)$. d_ϕ is the dimension of the last FC layer of g_ϕ.

Co-attention: We directly apply the temporal attention obtained from image features on object interaction representations $\mathbf{h} = (h_1, h_2, ..., h_T)$ (see Sec 3.1.2 for details).

Language LSTM: Finally, the Language LSTM takes in input which is the concatenation of output of the Attention LSTM $h^1_{t_w}$, attended video representation $\hat{v}_{c,t_w}$, and co-attended object interactions $\hat{h}_{t_w}$ at timestep t_w.

$$x^2_{t_w} = h^1_{t_w} \parallel \hat{v}_{c,t_w} \parallel \hat{h}_{t_w} \tag{13}$$

The output of Language LSTM is then used to generate each word, which is a conditional probability distribution defined as:

$$p(y_{t_w}|y_{1:t_w-1}) = softmax(W_p h^2_{t_w}) \tag{14}$$

where $y_{1:t_w-1}$ is a sequence of outputs $(y_1, ..., y_{t_w-1})$ and $W_p \in \mathbb{R}^{\Sigma \times d_{h^2_{t_w}}}$ is learned weights for $h^2_{t_w}$. All bias terms are omitted for simplicity.

4. Datasets and Implementations

4.1. Datasets:

Kinetics dataset: To evaluate SINet on a sequence-to-one problem for video, we use the Kinetics dataset for action recognition [21]. The Kinetics dataset contains 400 human action classes and has approximately 300k video clips (833 video hours). Most importantly, different from previous datasets which mostly cover sports actions [20, 23, 45], Kinetics includes human-object interactions and human-human interactions. We sampled videos at 1 FPS only, as opposed to sampling at 25 FPS reported for Kinetics [21].

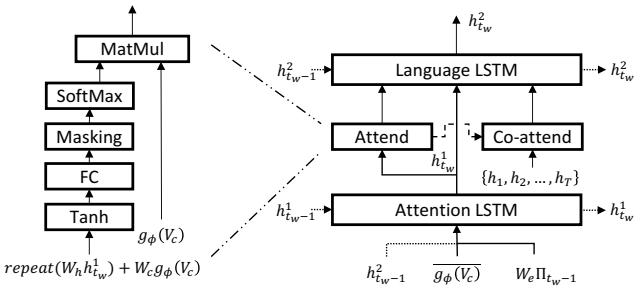

Figure 6. Overview of the proposed SINet-Caption for video captioning. The Attention LSTM with α-attention is used to selectively attend to temporal video frame features. The computed temporal attention is then used to attend to temporal object interactions $\{h_1, h_2, ..., h_T\}$ (see Figure 4). Concatenation of the outputs of Attention LSTM, attended video frame feature, and attended object interactions is then used as input for language decoder LSTM.

ActivityNet Captions dataset: To evaluate SINet-Caption on a sequence-to-sequence problem for video, we use ActivityNet Captions for video captioning. The ActivityNet Captions dataset contains 20k videos and has total of 849 video hours with 100K total descriptions. To demonstrate our proposed idea, we focus on providing fine-grained understanding of the video to describe video events with natural language, as opposed to identifying the temporal proposals. We thus use the ground truth temporal segments and treat each temporal segment independently. We use this dataset over others because ActivityNet Captions is action-centric, as opposed to object-centric [22]. This fits our goal of detecting higher-order object interactions for understanding human actions. All sentences are capped to be a maximum length of 30 words. We sample predictions using beam search of size 5 for captioning. While the previous work sample C3D features every 8 frames [22], we only sampled video at maximum 1 FPS. Video segments longer than 30 secs. are evenly sampled at maximum 30 samples.

4.2. Implementation Details:

We now discuss how to extract image and object features for both Kinetics and ActivityNet Captions.

Image feature: We fine-tune a pre-trained ResNeXt-101 [52] on Kinetics sampled at 1 FPS (approximately 2.5 million images). We use SGD with Nesterov momentum as the optimizer. The initial learning rate is $1e-4$ and drops by 10x when validation loss saturates for 5 epochs. The weight decay is $1e-4$ and the momentum is 0.9, and the batch size is 128. We use standard data augmentation by randomly cropping and horizontally flipping video frames dur-

Table 1. Prediction accuracy on the Kinetics validation set. All of our results use only RGB videos sampled at 1 FPS. Maximum number of objects per frame is set to be 30.

Method	Top-1	Top-5
I3D[2](25 FPS) [6] (test)	71.1	89.3
TSN (Inception-ResNet-v2) (2.5 FPS) [4, 51]	73.0	90.9
Ours (1 FPS)		
Img feature + LSTM (baseline)	70.6	89.1
Img feature + temporal SDP-Attention	71.1	89.6
Obj feature (mean-pooling)	72.2	90.2
Img + obj feature (mean-pooling)	73.1	91.1
SINet (α-attention)	73.9	91.5
SINet (dot-product attention)	**74.2**	**91.7**

ing training. When extracting image features, the smaller edge of the image is scaled to 256 pixels and we crop the center of the image as input to the fine-tuned ResNeXt-101. Each image feature is a 2048-d feature vector.

Object feature: We generate the object features by first obtaining the coordinates of ROIs from a Deformable R-FCN [9] (pre-trained on MS-COCO) with ResNet-101 [16] as backbone architecture. We set the IoU threshold for NMS to be 0.2. Empirically, we found that it is important to maintain a balance of image and object features, especially when image features were obtained from a network which was fine-tuned on the target dataset. Thus, for each of the ROIs, we extract features using coordinates and adaptive max-pooling from the same model (ResNeXt-101) that was fine-tuned on Kinetics. The resulting object feature for each ROI is a 2048-d feature vector. ROIs are ranked according to their ROI scores. We select top 30 objects for Kinetics and top 15 for ActivityNet Captions. Note that we have a varied number of ROIs for each video frame, and video length can also be different. We do not use the object class information since we may miss some of the objects that were not detected, due to the cross-domain problem. For the same reason, the bounding-box regression process is not performed here since we do not have the ground-truth bounding boxes.

Training: We train SINet and SINet-Caption with ADAM optimizer. The initial learning rate is set to $1e{-}5$ for Kinetics and $1e-3$ for ActivityNet Captions. Both learning rates automatically drop by 10x when validation loss is saturated. The batch sizes are 64 and 32 respectively for Kinetics and ActivityNet Captions.

5. Evaluation

5.1. Action recognition on Kinetics:

In this section, we conduct an ablation study of SINet on Kinetics.

Does temporal SDP-Attention help? Several studies have pointed out that using temporal mean-pooling or LSTMs may not be the best method to aggregate the sequence of image representations for videos [4, 29, 30]. To overcome this issue, we use temporal SDP-Attention instead of LSTM. As we can see from Table 1, using temporal SDP-Attention has proven to be superior to traditional LSTM and already performs comparably with 3D ConvNet that uses a much higher video sampling rate.

[2]Results obtained from https://github.com/deepmind/kinetics-i3d

Table 2. Comparison of pairwise (or triplet) object interaction with the proposed higher-order object interaction with dot-product attentive selection method on Kinetics. The maximum number of objects is set to be 15. FLOP is calculated per video. For details on calculating FLOP, please refer to the supplementary material.

Method	Top-1	Top-5	FLOP (e^9)
Obj (mean-pooling)	73.1	90.8	1.9
Obj pairs (mean-pooling)	73.4	90.8	18.3
Obj triplet (mean-pooling)	72.9	90.7	77.0
SINet ($K=1$)	73.9	91.3	2.7
SINet ($K=2$)	74.2	91.5	5.3
SINet ($K=3$)	**74.2**	**91.7**	8.0

Does object interaction help? We first evaluate how much higher-order object interactions can help in identifying human actions. Considering mean-pooling over the object features to be the simplest form of object interaction, we show that mean-pooling over the object features per frame and using LSTM for temporal reasoning has already outperformed single compact image representations, which is currently the trend for video classification methods. Directly combining image features with temporal SDP-Attention and object features over LSTM further reaches 73.1% top-1 accuracy. This already outperforms the state-of-the-art TSN [51] method using a deeper ConvNet with a higher video sampling rate. Beyond using mean-pooling as the simplest form of object interaction, our proposed method to dynamically discover and model higher-order object interactions further achieved 74.2% top-1 and 91.7% top-5 accuracy. The selection module with dot-product attention, in which we exploit the inter-relationships between objects within the same group, outperforms α-attention where the inter-relationships are ignored.

Does attentive selection help? Prior work on visual relationships and VQA concatenate pairwise object features for detecting object relationships. In this experiment, we compare the traditional way of creating object pairs or triplets with our proposed attentive selection method. We use temporal SDP-Attention for image features, and dot-project attention for selecting object interactions. As shown in Table 2, concatenating pairwise features marginally improves over the simplest form of object interactions while increasing the computational cost drastically. By further concatenating three object features, the space for meaningful object interactions becomes so sparse that it instead reduced the prediction accuracy, and the number of operations (FLOP) further increases drastically. On the other hand, our

Table 3. METEOR, ROUGE-L, CIDEr-D, and BLEU@N scores on the ActivityNet Captions test and validation set. All methods use ground truth proposal except LSTM-A_3 [12]. Our results with ResNeXt spatial features use videos sampled at maximum 1 FPS only.

Method	B@1	B@2	B@3	B@4	ROUGE-L	METEOR	CIDEr-D
Test set							
LSTM-YT [50] (C3D)	18.22	7.43	3.24	1.24	-	6.56	14.86
S2VT [49] (C3D)	20.35	8.99	4.60	2.62	-	7.85	20.97
H-RNN [55] (C3D)	19.46	8.78	4.34	2.53	-	8.02	20.18
S2VT + full context [22] (C3D)	26.45	13.48	7.21	3.98	-	9.46	24.56
LSTM-A_3 + policy gradient + retrieval [12] (ResNet + P3D ResNet [37])	-	-	-	-	-	12.84	-
Validation set (Avg. 1st and 2nd)							
LSTM-A_3 (ResNet + P3D ResNet) [12]	17.5	9.62	5.54	**3.38**	13.27	7.71	16.08
LSTM-A_3 + policy gradient + retrieval [12] (ResNet + P3D ResNet [37])	17.27	9.70	5.39	3.13	14.29	8.73	14.75
SINet-Caption — img (C3D)	17.18	7.99	3.53	1.47	18.78	8.44	38.22
SINet-Caption — img (ResNeXt)	18.81	9.31	4.27	1.84	20.46	9.56	43.12
SINet-Caption — obj (ResNeXt)	19.07	9.48	4.38	1.92	20.67	9.56	44.02
SINet-Caption — img + obj — no co-attention (ResNeXt)	**19.93**	9.82	**4.52**	2.03	21.08	9.79	44.81
SINet-Caption — img + obj — co-attention (ResNeXt)	19.78	**9.89**	**4.52**	1.98	**21.25**	**9.84**	**44.84**

attentive selection method can improve upon these methods while saving significant computation time. Empirically, we also found that reducing the number of objects per frame from 30 to 15 yields no substantial difference on prediction accuracy. This indicates that the top 15 objects with highest ROI score are sufficient to represent fine-grained details of the video. For detailed qualitative analysis of how objects are selected at each timestep and how SINet reasons over a sequence of object interactions, please see the supplement.

We are aware of that integrating optical flow or audio information with RGB video can further improve the action recognition accuracy [4, 6]. We instead focus on modeling object interactions for understanding video in a fine-grained manner, and we consider other modalities to be complementary to our higher-order object interactions.

5.2. Video captioning on ActivityNet Captions:

We focus on understanding human actions for video captioning rather than on temporal proposals. Hence, we use ground truth temporal proposals for segmenting the videos and treat each video segment independently. All methods in Table 3 use ground truth temporal proposal, except LSTM-A_3 [12]. Our performances are reported with four language metrics, including BLEU [35], ROUGH-L [28], METEOR [3], and CIDEr-D [48].

For fair comparison with prior methods using C3D features, we report results with both C3D and ResNeXt spatial features. Since there is no prior result reported on the validation set, we compare against LSTM-A_3 [12] which reports results on the validation and test sets. This allows us to indirectly compare with methods reported on the test set. As shown in Table 3, while LSTM-A_3 clearly outperforms other methods on the test set with a large margin, our method shows better results on the validation sets across nearly all language metrics. We do not claim our method to be superior to LSTM-A_3 because of two fundamental differences. First, they do not rely on ground truth temporal proposals. Second, they use features extracted from an ResNet fine-tuned on Kinetics and another P3D ResNet [37] fine-tuned on Sports-1M, whereas we use a ResNeXt-101 fine-tuned on Kinetics sampled at maximum 1 FPS. Utilizing more powerful feature representations has been proved to improve the prediction accuracy by a large margin on video tasks. This also corresponds to our experiments with C3D and ResNeXt features, where the proposed method with ResNeXt features perform significantly better than C3D features.

Does object interaction help? SINet-Caption without any object interaction has already outperformed prior methods reported on this dataset. Additionally, by introducing an efficient selection module for detecting object interactions, SINet-Caption further improves across nearly all evaluation metrics, with or without co-attention. We also observed that introducing the co-attention from image features constantly shows improvement on the first validation set but having separate temporal attention for object interaction features show better results on second validation set (please see the supplement for results on each validation set).

6. Conclusion

We introduce a computationally efficient fine-grained video understanding approach for discovering higher-order object interactions. Our work on large-scale action recognition and video captioning datasets demonstrates that learning higher-order object relationships provides high accuracy over existing methods at low computation costs. We achieve state-of-the-art performances on both tasks with only RGB videos sampled at maximum 1 FPS.

Acknowledgments

Zsolt Kira was partially supported by the National Science Foundation and National Robotics Initiative (grant # IIS-1426998).

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 1, 3

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017. 5

[3] 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*, volume 29, pages 65–72, 2005. 8

[4] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017. 1, 7, 8

[5] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015. 1

[6] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 7, 8

[7] Y.-W. Chao, Y. Liu, X. Liu, H. Zeng, and J. Deng. Learning to detect human-object interactions. *arXiv preprint arXiv:1702.05448*, 2017. 2, 4

[8] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 4

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2017. 7

[10] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016. 2

[11] Z. Gan, C. Gan, X. He, Y. Pu, K. Tran, J. Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[12] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Khrisna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. *arXiv preprint arXiv:1710.08011*, 2017. 8

[13] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[14] G. Gkioxari, R. Girshick, P. Dollár, and K. He. Detecting and recognizing human-object interactions. *arXiv preprint arXiv:1704.07333*, 2017. 2

[15] G. Gkioxari, R. Girshick, and J. Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1080–1088, 2015. 2

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[17] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3, 4

[18] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos in the wild. *Computer Vision and Image Understanding*, 155:1–23, 2017. 1

[19] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3668–3678, 2015. 2

[20] 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. 1, 6

[21] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 3, 6

[22] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 6, 8

[23] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 1, 6

[24] C. Lea, A. Reiter, R. Vidal, and G. D. Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52. Springer, 2016. 3

[25] Y. Li, C. Huang, C. C. Loy, and X. Tang. Human attribute recognition by deep hierarchical contexts. In *European Conference on Computer Vision*, pages 684–700. Springer, 2016. 2

[26] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1261–1270, 2017. 2

[27] X. Liang, L. Lee, and E. P. Xing. Deep variation-structured reinforcement learning for visual relationship and attribute detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[28] 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. 8

[29] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *arXiv preprint arXiv:1703.10667*, 2017. 1, 3, 7

[30] A. Miech, I. Laptev, and J. Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 1, 2, 7

[31] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 756–763, 2014. 2, 3

[32] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1020–1028, 2016. 2, 3

[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. 2

[34] Y. Pan, T. Yao, H. Li, and T. Mei. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[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 association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 8

[36] X. Peng and C. Schmid. Multi-region two-stream r-cnn for action detection. In *European Conference on Computer Vision*, pages 744–759. Springer, 2016. 2

[37] Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 8

[38] V. Ramanishka, A. Das, J. Zhang, and K. Saenko. Top-down visual saliency guided by captions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[39] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, 2017. 3, 4

[40] Z. Shen, J. Li, Z. Su, M. Li, Y. Chen, Y.-G. Jiang, and X. Xue. Weakly supervised dense video captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[41] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2

[42] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016. 1

[43] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 2

[44] J. Song, Z. Guo, L. Gao, W. Liu, D. Zhang, and H. T. Shen. Hierarchical lstm with adjusted temporal attention for video captioning. *International Joint Conference on Artificial Intelligence*, 2017. 2, 5

[45] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 1, 6

[46] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3d: generic features for video analysis. *CoRR, abs/1412.0767*, 2:7, 2014. 1, 2

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 3, 4

[48] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 8

[49] 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. 2, 8

[50] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1495–1504, 2014. 2, 8

[51] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 7

[52] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 6

[53] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[54] 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. 2

[55] H. Yu, J. Wang, Z. Huang, Y. Yang, and W. Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4584–4593, 2016. 2, 5, 8

[56] Y. Yu, H. Ko, J. Choi, and G. Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, volume 3, page 7, 2017. 2

[57] M. Zanfir, E. Marinoiu, and C. Sminchisescu. Spatio-temporal attention models for grounded video captioning. In *Asian Conference on Computer Vision*, pages 104–119. Springer, 2016. 2

[58] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3, 4

[59] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, and A. Elgammal. Relationship proposal networks. In *CVPR*, volume 1, page 2, 2017. 2, 3