Dense-Captioning Events in Videos

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, Juan Carlos Niebles
Stanford University
{ranjaykrishna, kenjihata, fren, feifeili, jniebles}@cs.stanford.edu

Abstract

Most natural videos contain numerous events. For example, in a video of a "man playing a piano", the video might also contain "another man dancing" or "a crowd clapping". We introduce the task of dense-captioning events, which involves both detecting and describing events in a video. We propose a new model that is able to identify all events in a single pass of the video while simultaneously describing the detected events with natural language. Our model introduces a variant of an existing proposal module that is designed to capture both short as well as long events that span minutes. To capture the dependencies between the events in a video, our model introduces a new captioning module that uses contextual information from past and future events to jointly describe all events. We also introduce ActivityNet Captions, a large-scale benchmark for dense-captioning events. ActivityNet Captions contains 20k videos amounting to 849 video hours with 100k total descriptions, each with its unique start and end time. Finally, we report performances of our model for dense-captioning events, video retrieval and localization.

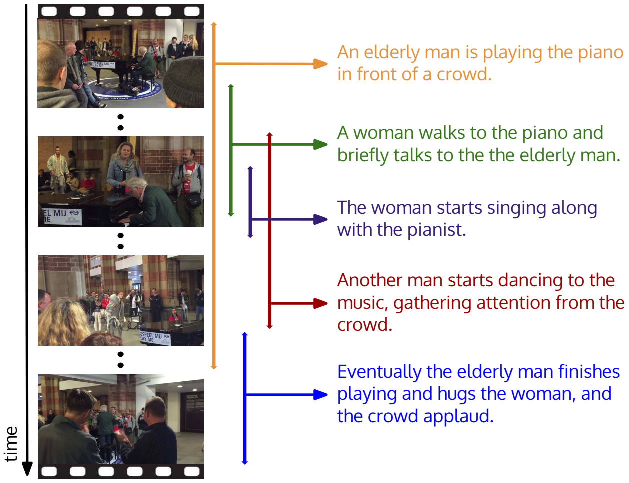

An elderly man is playing the piano in front of a crowd.
A woman walks to the piano and briefly talks to the the elderly man.
The woman starts singing along with the pianist.
Another man starts dancing to the music, gathering attention from the crowd.
Eventually the elderly man finishes playing and hugs the woman, and the crowd applaud.
time

Figure 1: Dense-captioning events in a video involves detecting multiple events that occur in a video and describing each event using natural language. These events are temporally localized in the video with independent start and end times, resulting in some events that might also occur concurrently and overlap in time.

1. Introduction

With the introduction of large scale activity datasets [22, 19, 12, 4], it has become possible to categorize videos into a discrete set of action categories [27, 10, 9, 40, 35]. For example, in Figure 1, such models would output labels like *playing piano* or *dancing*. While the success of these methods is encouraging, they all share one key limitation: detail. To elevate the lack of detail from existing action detection models, subsequent work has explored explaining video semantics using sentence descriptions [29, 32, 28, 38, 37]. For example, in Figure 1, such models would likely concentrate on *an elderly man playing the piano in front of a crowd*. While this caption provides us more details about who is playing the piano and mentions an audience, it fails to recognize and articulate all the other events in the video. For example, at some point in the video, *a woman starts singing along with the pianist* and then later *another man starts dancing to the music*. In order to identify all the events in a video and describe them in natural language, we introduce the task of *dense-captioning events*, which requires a model to generate a set of descriptions for multiple events occurring in the video and localize them in time.

Dense-captioning events is analogous to dense-image-captioning [16]; it describes videos and localize events in time whereas dense-image-captioning describes and localizes regions in space. However, we observe that dense-captioning events comes with its own set of challenges distinct from the image case. One observation is that events in videos can range across multiple time scales and can even overlap. While *piano recitals* might last for the entire duration of a long video, *the applause* takes place in a couple of seconds. To capture all such events, we need to design ways of encoding short as well as long sequences of video frames to propose events. Past captioning works have circumvented this problem by encoding the entire video sequence by mean-pooling [38] or by using a recurrent neural network (RNN) [37]. While this works well for short clips, encoding long video sequences that span minutes leads to vanishing gradients, preventing successful training. To overcome this limitation, we extend recent work on generating action proposals [8] to **multi-scale detection of events**. Also, our proposal module processes each video in a forward pass, allowing us to detect events as they occur.

Another key observation is that the events in a given video are usually related to one another. In Figure 1, *the crowd applauds* because a *a man was playing the piano*. Therefore, our model must be able to use context from surrounding events to caption each event. A recent paper has attempted to describe videos with multiple sentences [51]. However, their model generates sentences for instructional "cooking" videos where the events occur sequentially and highly correlated to the objects in the video [31]. We show that their model does not generalize to "open" domain videos where events are action oriented and can even overlap. We introduce a **captioning module that utilizes the context** from all the events from our proposal module to generate each sentence. In addition, we show a variant of our captioning module that can operate on streaming videos by attending over only the past events. Our full model attends over both past as well as future events and demonstrates the importance of using context.

To evaluate our model and benchmark progress in dense-captioning events, we introduce the ActivityNet Captions dataset[1]. ActivityNet Captions contains 20k videos taken from ActivityNet [4], where each video is annotated with a series of temporally localized descriptions (Figure 1). To showcase long term event detection, our dataset contains videos as long as 10 minutes, with each video annotated with on average 3.65 sentences. The descriptions refer to events that might be simultaneously occurring, causing the video segments to overlap. We ensure that each description in a given video is unique and refers to only one segment. While our videos are centered around human activities, the descriptions may also refer to non-human events such as: *two hours later, the mixture becomes a delicious cake to eat*. We collect our descriptions using crowdsourcing and find that there is high agreement in the temporal event segments, which is in line with research suggesting that brain activity is naturally structured into semantically meaningful events [2].

With ActivityNet Captions, we are able to provide the first results for the task of dense-captioning events. Together with our online proposal module and our online captioning module, we show that we can detect and describe events in long or even streaming videos. We demonstrate that we are able to detect events found in short clips as well as in long video sequences. Furthermore, we show that utilizing context from other events in the video improves dense-captioning events. Finally, we demonstrate how ActivityNet Captions can be used to study video retrieval as well as event localization.

2. Related work

Dense-captioning events bridges two separate bodies of work: temporal action proposals and video captioning. First, we review related work on action recognition, action detection and temporal proposals. Next, we survey how video captioning started from video retrieval and video summarization, leading to single-sentence captioning work. Finally, we contrast our work with recent work in captioning images and videos with multiple sentences.

Early work in **activity recognition** involved using hidden Markov models to learn latent action states [45], followed by discriminative SVM models that used key poses and action grammars [26, 36, 30]. Similar works have used hand-crafted features [33] or object-centric features [25] to recognize actions in fixed camera settings. More recent works have used dense trajectories [39] or deep learning features [17] to study actions. While our work is similar to these methods, we focus on describing such events with natural language instead of a fixed label set.

To enable action localization, **temporal action proposal** methods started from traditional sliding window approaches [7] and later started building models to propose a handful of possible action segments [8, 5]. These proposal methods have used dictionary learning [5] or RNN architectures [8] to find possible segments of interest. However, such methods required each video frame to be processed once for every sliding window. DAPs introduced a framework to allow proposing overlapping segments using a sliding window. We modify this framework by removing the sliding windows and outputting proposals at every time step in a single pass of the video. We further extend this model and enable it to detect long events by implementing a multi-scale version of DAPs, where we sample frames at longer strides.

Orthogonal to work studying proposals, early approaches that connected video with language studied the task of **video retrieval with natural language**. They worked on generating a common embedding space between language and videos [28, 44]. Similar to these, we evaluate how well existing models perform on our dataset. Additionally, we introduce the task of localizing a given sentence given a video frame, allowing us to now also evaluate whether our models are able to locate specified events.

In an effort to start describing videos, methods in **video summarization** aimed to congregate segments of videos that include important or interesting visual information [49,

[1] The dataset is available at `http://cs.stanford.edu/people/ranjaykrishna/densevid/`. For a detailed analysis of our dataset, please see our supplementary material.

46, 13, 3]. These methods attempted to use low level features such as color and motion or attempted to model objects [52] and their relationships [41, 11] to select key segments. Meanwhile, others have utilized text inputs from user studies to guide the selection process [34, 23]. While these summaries provide a means of finding important segments, these methods are limited by small vocabularies and do not evaluate how well we can explain visual events [50].

After these summarization works, early attempts at **video captioning** [38] simply mean-pooled video frame features and used a pipeline inspired by the success of image captioning [18]. However, this approach only works for short video clips with only one major event. To avoid this issue, others have proposed either a recurrent encoder [6, 37, 42] or an attention mechanism [48]. To capture more detail in videos, a new paper has recommended describing videos with paragraphs (a list of sentences) using a hierarchical RNN [24] where the top level network generates a series of hidden vectors that are used to initialize low level RNNs that generate each individual sentence [51]. While our paper is most similar to this work, we address two important missing factors. First, the sentences that their model generates refer to different events in the video but are not localized in time. Second, they use the TACoS-MultiLevel [31], which contains less than 200 videos and is constrained to "cooking" videos and only contain non-overlapping sequential events. We address these issues by introducing the ActivityNet Captions dataset which contains overlapping events and by introducing our captioning module that uses temporal context to capture the interdependency between all the events in a video.

Finally, we build upon the recent work on **dense-image-captioning** [16], which generates a set of localized descriptions for an image. Further work for this task has used spatial context to improve captioning [47, 43]. Inspired by this work, and by recent literature on using spatial attention to improve human tracking [1], we design our captioning module to incorporate temporal context (analogous to spatial context except in time) by attending over the other events in the video.

3. Dense-captioning events model

Overview. Our goal is to design an architecture that jointly localizes temporal proposals of interest and then describes each with natural language. The two main challenges we face are to develop a method that can (1) detect multiple events in short as well as long video sequences and (2) utilize the context from past, concurrent and future events to generate descriptions of each one. Our proposed architecture (Figure 2) draws on architectural elements present in recent work on action proposal [8] and social human tracking [1] to tackle both these challenges.

Formally, the input to our system is a sequence of video frames $\mathbf{U} = \{\mathbf{u}_t\}$ where $t \in \{0, ..., T-1\}$ indexes the frames in temporal order. Our output is a set of sentences $s_i = (t^{\text{start}}, t^{\text{end}}, \{v_j\})$ consists of the start and end times for each sentence and is defined by a set of words $v_j \in V$ with differing lengths for each sentence where V is our vocabulary set.

Our model first sends the video frames through a proposal module that generates a set of proposals:

$$P = \{(t_i^{\text{start}}, t_i^{\text{end}}, \text{score}_i, \mathbf{h}_i)\} \tag{1}$$

All the proposals with a $score_i$ higher than a threshold are forwarded to our language model that uses context from the other proposals while captioning each event. The hidden representation $\mathbf{h}_i$ of the event proposal module is used as inputs to the captioning module, which then outputs descriptions for each event, while utilizing the context from the other events.

3.1. Event proposal module

Prior event detection work usually pools video features globally into a fixed sized vector [6, 37, 42], which is sufficient for representing short video clips but is unable to detect multiple events in long videos. Previous work [14] showed that actions can be modeled with steady feature derivatives using the intuition that visual features of events change at a fixed rate. We design an event proposal module to be a variant of DAPs [8] that can detect longer events by sampling at different rates, allowing the model to encode events that occur over a wider range of feature changes, allowing us to capture short events with faster changes as well as longer events with slower changes.

Input. Our proposal module receives a series of features capturing semantic information from the video frames. Concretely, the input to our proposal module is a sequence of features: $\{\mathbf{f}_t = F(\mathbf{u}_t : \mathbf{u}_{t+\delta})\}$ where δ is the time resolution of each feature $\mathbf{f}_t$. In our paper, F extracts C3D features [15] where $\delta = 16$ frames. The output of F is a tensor of size $N \times D$ where $D = 500$ dimensional features and $N = T/\delta$ discretizes the video frames.

DAPs. Next, we feed these features into a variant of DAPs [8] where we sample the videos features at different strides (1, 2, 4 and 8 for our experiments) and feed them into a proposal long short-term memory (LSTM) unit. The longer strides are able to capture longer events. The LSTM accumulates evidence across time as the video features progress. We do not modify the training of DAPs and only change the model at inference time by outputting K proposals at every time step, each proposing an event with temporal start and end times. So, the LSTM is capable of generating proposals at different overlapping time intervals and we only need to iterate over the video once, since all the strides can be computed in parallel. Whenever the proposal LSTM detects an event, we use the hidden state of the

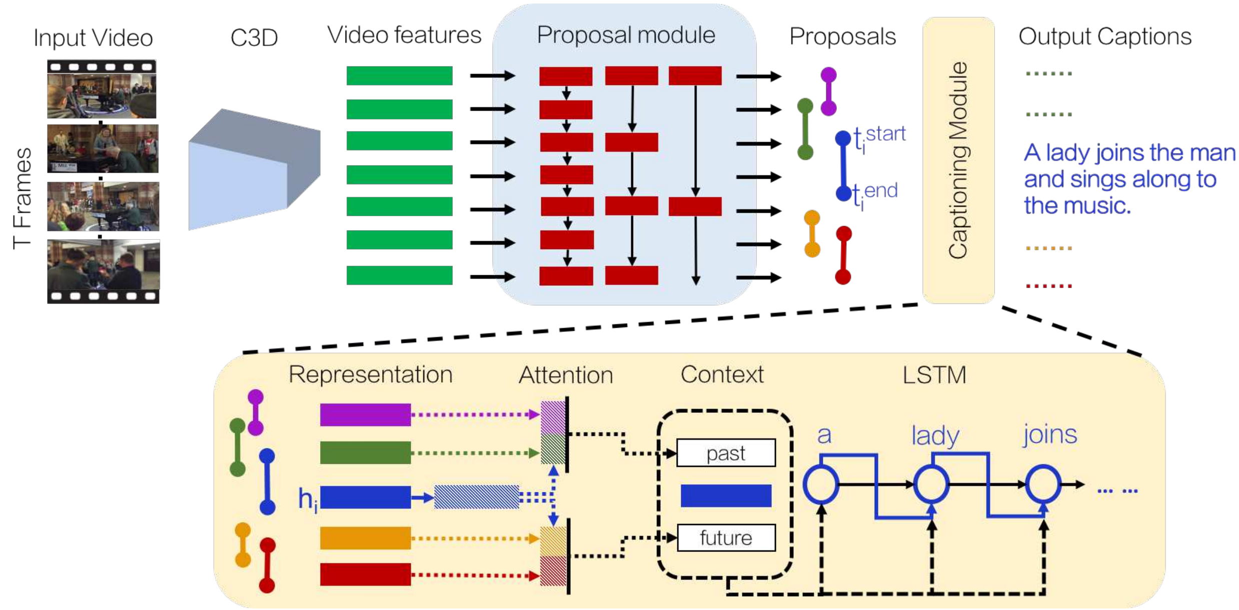


Figure 2: Complete pipeline for dense-captioning events in videos with descriptions. We first extract C3D features from the input video. These features are fed into our proposal module at varying stride to predict both short as well as long events. Each proposal, which consists of a unique start and end time and a hidden representation, is then used as input into the captioning module. Finally, this captioning model leverages context from neighboring events to generate each event description.

LSTM at that time step as a feature representation of the visual event. Note that the proposal model can output proposals for events that can be overlapping. While traditional DAPs uses non-maximum suppression to eliminate overlapping outputs, we keep them separately and treat them as individual events.

3.2. Captioning module with context

Once we have the event proposals, the next stage of our pipeline is responsible for describing each event. A naive captioning approach could treat each description individually and use a captioning LSTM network to describe each one. However, most events in a video are correlated and can even cause one another. For example, we saw in Figure 1 that the *man playing the piano* caused the *other person to start dancing*. We also saw that after the man finished playing the piano, the *audience applauded*. To capture such correlations, we design our captioning module to incorporate the "context" from its neighboring events. Inspired by recent work [1] on human tracking that utilizes spatial context between neighboring tracks, we develop an analogous model that captures temporal context in videos by grouping together events in time instead of tracks in space.

Incorporating context. To capture the context from all other neighboring events, we categorize all events into two buckets relative to a reference event. These two context buckets capture events that have already occurred (past), and events that take place after this event has finished (future). Concurrent events are split into one of the two buckets: past if it ends early and future otherwise. For a given video event from the proposal module, with hidden representation h_i and start and end times of $[t_i^{\text{start}}, t_i^{\text{end}}]$, we calculate the past and future context representations as follows:

$$\mathbf{h}_i^{\text{past}} = \frac{1}{Z^{\text{past}}} \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} < t_i^{\text{end}}] a_{ij} \mathbf{h}_j \tag{2}$$

$$\mathbf{h}_i^{\text{future}} = \frac{1}{Z^{\text{future}}} \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} >= t_i^{\text{end}}] a_{ij} \mathbf{h}_j \tag{3}$$

where $\mathbf{h}_j$ is the hidden representation of the other proposed events in the video. a_{ij} is the attention used to determine how relevant event j is to event i. Z is the normalization that is calculated as $Z^{\text{past}} = \sum_{j \neq i} \mathbb{1}[t_j^{\text{end}} < t_i^{\text{end}}]$. We calculate a_{ij} as follows:

$$\mathbf{w}_i = \mathbf{w}_a \mathbf{h}_i + \mathbf{b}_a \tag{4}$$

$$a_{ij} = \mathbf{w}_i \mathbf{h}_j \tag{5}$$

where $\mathbf{w}_i$ is the annotation vector calculated from the learnt weights $\mathbf{w}_a$ and bias $\mathbf{b}_a$. We use the dot product of $\mathbf{w}_i$ and $\mathbf{h}_j$ to calculate a_{ij}. The concatenation of $(\mathbf{h}_i^{\text{past}}, \mathbf{h}_i, \mathbf{h}_i^{\text{future}})$ is then fed as the input to the captioning LSTM that describes the event. With the help of the context, each LSTM also has knowledge about events that have happened or will happen and can tune its captions accordingly.

Language modeling. Each language LSTM is initialized to have 2 layers with 512 dimensional hidden representation.

We randomly initialize all the word vector embeddings from a Gaussian with standard deviation of 0.01. We sample predictions from the model using beam search of size 5.

3.3. Implementation details.

Loss function. We use two separate losses to train both our proposal model ($\mathcal{L}_{\text{prop}}$) and our captioning model ($\mathcal{L}_{\text{cap}}$). Our proposal models predicts confidences ranging between 0 and 1 for varying proposal lengths. We use a weighted cross-entropy term to evaluate each proposal confidence.

We only pass to the language model proposals that have a high IoU with ground truth proposals. Similar to previous work on language modeling [20, 18], we use a cross-entropy loss across all words in every sentence. We normalize the loss by the batch-size and sequence length in the language model. We weight the contribution of the captioning loss with $\lambda_1 = 1.0$ and the proposal loss with $\lambda_2 = 0.1$:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{cap}} + \lambda_2 \mathcal{L}_{\text{prop}} \tag{6}$$

Training and optimization. We train our full dense-captioning model by alternating between training the language model and the proposal module every 500 iterations. We first train the captioning module without any context features for 10 epochs before adding in the context features. We initialize all weights using a Gaussian with standard deviation of 0.01. We use stochastic gradient descent with momentum 0.9 to train. We use an initial learning rate of 1×10^{-2} for the language model and 1×10^{-3} for the proposal module. For efficiency, we do not finetune the C3D feature extraction.

Our training batch-size is set to 1. We cap all sentences to be a maximum sentence length of 30 words and implement all our code in PyTorch 0.1.10. One mini-batch runs in approximately 15.84 ms on a Titan X GPU and it takes 2 days for the model to converge.

4. ActivityNet Captions dataset

The ActivityNet Captions dataset connects videos to a series of temporally annotated sentences. Each sentence covers an unique segment of the video, describing an event that occurs. These events may occur over very long or short periods of time and are not limited in any capacity, allowing them to co-occur. We will now present an overview of the dataset and also provide a detailed analysis and comparison with other datasets in our supplementary material.

4.1. Dataset statistics

On average, each of the 20k videos in ActivityNet Captions contains 3.65 temporally localized sentences, resulting in a total of 100k sentences. We find that the number of sentences per video follows a normal distribution. Furthermore, as the video duration increases, the number of sentences also increases. Each sentence has an average length of 13.48 words, which is also normally distributed.

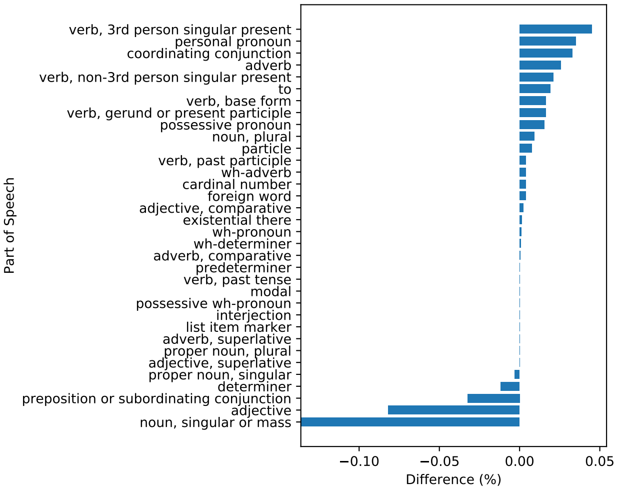


Figure 3: The parts of speech distribution of ActivityNet Captions compared with Visual Genome, a dataset with multiple sentence annotations per image. There are many more verbs and pronouns represented in ActivityNet Captions, as the descriptions often focus on actions.

On average, each sentence describes 36 seconds and 31% of their respective videos. However, the entire paragraph for each video on average describes 94.6% of the entire video, demonstrating that each paragraph annotation still covers all major actions within the video. Furthermore, we found that 10% of the temporal descriptions overlap, showing that the events cover simultaneous events.

Finally, our analysis on the sentences themselves indicate that ActivityNet Captions focuses on verbs and actions. In Figure 3, we compare against Visual Genome [21], the image dataset with most number of image descriptions ($\sim$ 4.5 million). With the percentage of verbs comprising ActivityNet Captions being significantly more, we find that ActivityNet Captions shifts sentence descriptions from being object-centric in images to action-centric in videos. Furthermore, as there exists a greater percentage of pronouns in ActivityNet Captions, we find that the sentence labels will more often refer to entities found in prior sentences.

4.2. Temporal agreement amongst annotators

To verify that ActivityNet Captions ’s captions mark semantically meaningful events [2], we collected two distinct, temporally annotated paragraphs from different workers for each of the 4926 validation and 5044 test videos. Each pair of annotations was then tested to see how well they temporally corresponded to each other. We found that, on average, each sentence description had a temporal intersection over union (tIoU) of 70.2% with the maximal overlapping com-

	with GT proposals						with learnt proposals					
	B@1	B@2	B@3	B@4	M	C	B@1	B@2	B@3	B@4	M	C
LSTM-YT [37]	18.22	7.43	3.24	1.24	6.56	14.86	-	-	-	-	-	-
S2VT [38]	20.35	8.99	4.60	2.62	7.85	20.97	-	-	-	-	-	-
H-RNN [51]	19.46	8.78	4.34	2.53	8.02	20.18	-	-	-	-	-	-
no context (ours)	20.35	8.99	4.60	2.62	7.85	20.97	12.23	3.48	2.10	0.88	3.76	12.34
online−attn (ours)	21.92	9.88	5.21	3.06	8.50	22.19	15.20	5.43	2.52	1.34	4.18	14.20
online (ours)	22.10	10.02	5.66	3.10	8.88	22.94	17.10	7.34	3.23	1.89	4.38	15.30
full−attn (ours)	26.34	13.12	6.78	3.87	9.36	24.24	15.43	5.63	2.74	1.72	4.42	15.29
full (ours)	**26.45**	**13.48**	**7.12**	**3.98**	**9.46**	**24.56**	**17.95**	**7.69**	**3.86**	**2.20**	**4.82**	**17.29**

Table 1: We report Bleu (B), METEOR (M) and CIDEr (C) captioning scores for the task of dense-captioning events. On the left, we report performances of just our captioning module with ground truth proposals. On the right, we report the combined performances of our complete model, with top 1000 proposals predicted from our proposal module. Since prior work has focused only on describing entire videos and not also detecting a series of events, we only compare existing video captioning models using ground truth proposals.

bination of sentences from the other paragraph. Since these results agree with prior work [2], we found that workers generally agree with each other when annotating temporal boundaries of video events.

5. Experiments

We evaluate our model by detecting multiple events in videos and describing them. We refer to this task as dense-captioning events (Section 5.1). We test our model on ActivityNet Captions, which was built specifically for this task.

Next, we provide baseline results on two additional tasks that are possible with our model. The first of these tasks is localization (Section 5.2), which tests our proposal model's capability to adequately localize all the events for a given video. The second task is retrieval (Section 5.3), which tests a variant of our model's ability to recover the correct set of sentences given the video or vice versa. Both these tasks are designed to test the event proposal module (localization) and the captioning module (retrieval) individually.

5.1. Dense-captioning events

To dense-caption events, our model is given an input video and is tasked with detecting individual events and describing each one with natural language.

Evaluation metrics. Inspired by the dense-image-captioning [16] metric, we use a similar metric to measure the joint ability of our model to both localize and caption events. This metric computes the average precision across tIoU thresholds of 0.3, 0.5, 0.7 when captioning the top 1000 proposals. We measure precision of our captions using traditional evaluation metrics: Bleu, METEOR and CIDEr.

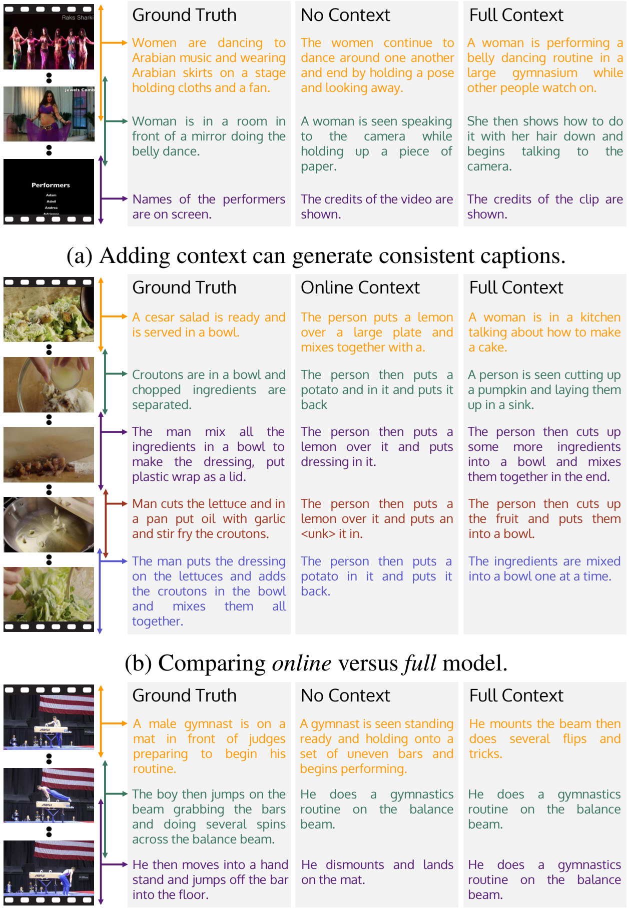
To isolate the performance of language in the predicted captions without localization, we also use ground truth locations across each test image and evaluate predicted captions.

	B@1	B@2	B@3	B@4	M	C
no context						
1^{st} sen.	23.60	12.19	7.11	4.51	9.34	**31.56**
2^{nd} sen.	19.74	8.17	3.76	1.87	7.79	19.37
3^{rd} sen.	18.89	7.51	3.43	1.87	7.31	19.36
online						
1^{st} sen.	24.93	12.38	7.45	4.77	8.10	30.92
2^{nd} sen.	19.96	8.66	4.01	1.93	7.88	19.17
3^{rd} sen.	19.22	7.72	3.56	**1.89**	7.41	19.36
full						
1^{st} sen.	**26.33**	**13.98**	**8.45**	**5.52**	**10.03**	29.92
2^{nd} sen.	**21.46**	**9.06**	**4.40**	**2.33**	**8.28**	**20.17**
3^{rd} sen.	**19.82**	**7.93**	**3.63**	1.83	**7.81**	**20.01**

Table 2: We report the effects of context on captioning the 1^{st}, 2^{nd} and 3^{rd} events in a video. We see that performance increases with the addition of past context in the online model and with future context in full model.

Baseline models. Since all the previous models proposed so far have focused on the task of describing entire videos and not detecting a series of events, we only compare existing video captioning models using ground truth proposals. Specifically, we compare our work with *LSTM-YT* [37], *S2VT* [38] and *H-RNN* [51]. *LSTM-YT* pools together video features to describe videos while *S2VT* [38] encodes a video using an RNN. *H-RNN* [51] generates paragraphs by using one RNN to caption individual sentences while the second RNN is used to sequentially initialize the hidden state for the next sentence generation. Our model can be though of as a generalization of the *H-RNN* model as it uses context, not just from the previous sentence but from surrounding events in the video. Additionally, our method treats context, not as features from object detectors but encodes it from unique parts of the proposal module.

Variants of our model. Additionally, we compare different



Ground Truth	No Context	Full Context
Women are dancing to Arabian music and wearing Arabian skirts on a stage holding cloths and a fan.	The women continue to dance around one another and end by holding a pose and looking away.	A woman is performing a belly dancing routine in a large gymnasium while other people watch on.
Woman is in a room in front of a mirror doing the belly dance.	A woman is seen speaking to the camera while holding up a piece of paper.	She then shows how to do it with her hair down and begins talking to the camera.
Names of the performers are on screen.	The credits of the video are shown.	The credits of the clip are shown.

(a) Adding context can generate consistent captions.

Ground Truth	Online Context	Full Context
A cesar salad is ready and is served in a bowl.	The person puts a lemon over a large plate and mixes together with a.	A woman is in a kitchen talking about how to make a cake.
Croutons are in a bowl and chopped ingredients are separated.	The person then puts a potato and in it and puts it back	A person is seen cutting up a pumpkin and laying them up in a sink.
The man mix all the ingredients in a bowl to make the dressing, put plastic wrap as a lid.	The person then puts a lemon over it and puts dressing in it.	The person then cuts up some more ingredients into a bowl and mixes them together in the end.
Man cuts the lettuce and in a pan put oil with garlic and stir fry the croutons.	The person then puts a lemon over it and puts an <unk> it in.	The person then cuts up the fruit and puts them into a bowl.
The man puts the dressing on the lettuces and adds the croutons in the bowl and mixes them all together.	The person then puts a potato in it and puts it back.	The ingredients are mixed into a bowl one at a time.

(b) Comparing *online* versus *full* model.

Ground Truth	No Context	Full Context
A male gymnast is on a mat in front of judges preparing to begin his routine.	A gymnast is seen standing ready and holding onto a set of uneven bars and begins performing.	He mounts the beam then does several flips and tricks.
The boy then jumps on the beam grabbing the bars and doing several spins across the balance beam.	He does a gymnastics routine on the balance beam.	He does a gymnastics routine on the balance beam.
He then moves into a hand stand and jumps off the bar into the floor.	He dismounts and lands on the mat.	He does a gymnastics routine on the balance beam.

(c) Context might add more noise to rare events.

Figure 4: Qualitative dense-captioning captions generated using our model. We show captions with the highest overlap with ground truth captions.

variants of our model. Our *no context* model is our implementation of *S2VT*. The *full* model is our complete model described in Section 3. The *online* model is a version of our full model that uses context only from past events and not from future events. This version of our model can be used to caption long streams of video in a single pass. The *full−attn* and *online−attn* models use mean pooling instead of attention to concatenate features, i.e. it sets $a_{ij} = 1$ in Equation 5.

Captioning results. Since all the previous work has focused on captioning complete videos, We find that *LSTM-YT* performs much worse than other models as it tries to encode long sequences of video by mean pooling their features (Table 1). *H-RNN* performs slightly better but attends over just the previous event to generate sentence, which causes it to only slightly outperform *LSTM-YT*. *S2VT* and our *no context* model performs better than the previous baselines with a CIDEr score of 20.97 as it uses an RNN to encode the video features. We see an improvement in performance to 22.19 and 22.94 when we incorporate context from past events into our *online−attn* and *online* models. Finally, we also considering events that will happen in the future, we see further improvements to 24.24 and 24.56 for the *full−attn* and *full* models. Note that while the improvements from using attention is not too large, we see greater improvements amongst videos with more events, suggesting that attention is useful for longer videos.

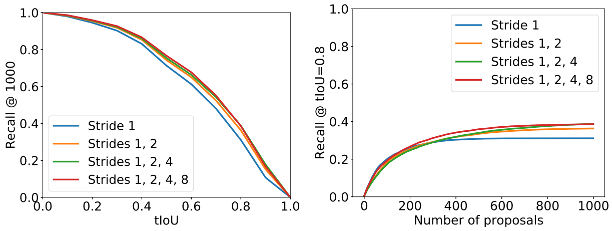


Figure 5: Evaluating our proposal module, we find that sampling videos at varying strides does in fact improve the module's ability to localize events, specially longer events.

Sentence order. To further benchmark the improvements calculated from using context, we report results using ground truth proposals for the first three sentences in each video (Table 2). While there are videos with more sentences, we report results only for the first three because almost all the videos in the dataset contains at least three sentences. We notice that the *online* and *full* models see most of their improvements from subsequent sentences, i.e. not the first sentence. In fact, we notice that after adding context, the CIDEr score for the *online* and *full* models tend to decrease for the 1^{st} sentence since these models know that they are generating a description for the first event in the video.

Results for dense-captioning events. When using proposals instead of ground truth events (Table 1), we see a similar trend where adding more context improves captioning. However, we also see that the improvements from attention are more pronounced since there are many events that the model has to caption. Attention allows the model to adequately focus on other events that are relevant to the current event. We show examples of qualitative results from the variants of our models in Figure 4. In (a), we see that the last caption in the *no context* model drifts off topic while the *full* model utilizes context to generate more reasonable context. In (b), we see that our *full* context model is able to use the knowledge that the vegetables are later *mixed in the bowl* to also mention *the bowl* in the third and fourth sentences, propagating context back through to past events. However, context is not always successful at generating better captions. In (c), when the proposed segments have a high overlap, our model fails to distinguish between the two

	Video retrieval				**Paragraph retrieval**			
	R@1	R@5	R@50	Med. rank	R@1	R@5	R@50	Med. rank
LSTM-YT [37]	0.00	0.04	0.24	102	0.00	0.07	0.38	98
no context [38]	0.05	0.14	0.32	78	0.07	0.18	0.45	56
online (ours)	0.10	**0.32**	0.60	36	0.17	0.34	0.70	33
full (ours)	**0.14**	**0.32**	**0.65**	**34**	**0.18**	**0.36**	**0.74**	**32**

Table 3: Results for video and paragraph retrieval. We see that the utilization of context to encode video events help us improve retrieval. R@k measures the recall at varying thresholds k and med. rank measures the median rank the retrieval.

events, causing it to repeat captions.

5.2. Event localization

One of the main goals of this paper is to develop models that can locate any given event within a video. Therefore, we test how well our model can predict the temporal location of events within the corresponding video, in isolation of the captioning module. Recall that our variant of the proposal module uses proposes videos at different strides. Specifically, we test with strides of 1, 2, 4 and 8. Each stride can be computed in parallel, allowing the proposal to run in a single pass.

Setup. We evaluate our proposal module using recall (like previous work [8]) against (1) the number of proposals and (2) the IoU with ground truth events. Specifically, we are testing whether, the use of different strides does in fact improve event localization.

Results. Figure 5 shows the recall of predicted localizations that overlap with ground truth over a range of IoU's from 0.0 to 1.0 and number of proposals ranging till 1000. We find that using more strides improves recall across all values of IoU's with diminishing returns . We also observe that when proposing only a few proposals, the model with stride 1 performs better than any of the multi-stride versions. This occurs because there are more training examples for smaller strides as these models have more video frames to iterate over, allowing them to be more accurate. So, when predicting only a few proposals, the model with stride 1 localizes the most correct events. However, as we increase the number of proposals, we find that the proposal network with only a stride of 1 plateaus around a recall of 0.3, while our multi-scale models perform better.

5.3. Video and paragraph retrieval

While we introduce dense-captioning events, a new task to study video understanding, we also evaluate our intuition to use context on a more traditional task: video retrieval.

Setup. In video retrieval, we are given a paragraph and are asked to retrieve the correct video from the test set by matching each sentence in the paragraph to the ground truth proposals in the videos. Each sentence, along with its context, is encoded using our captioning module while each proposal is encoded with our proposal model. We train our model using a max-margin loss that attempts to align the correct sentence encoding to its corresponding video proposal encoding. We also report how this model performs if the task is reversed, where we are given a video as input and are asked to retrieve the correct paragraph from the complete set of paragraphs in the test set.

Results. We report our results in Table 3. We evaluate retrieval using recall at various thresholds and the median rank. We use the same baseline models as our previous tasks. We find that models that use RNNs (*no context*) to encode the video proposals perform better than max pooling video features (LSTM-YT). We also see a direct increase in performance when context is used. Unlike dense-captioning, we do not see a marked increase in performance when we include context from future events as well. We find that our online models performs almost at par with our full model.

6. Conclusion

We introduced the task of dense-captioning events and identified two challenges: (1) events can occur within a second or last up to minutes, and (2) events in a video are related to one another. To tackle both these challenges, we proposed a model that combines a new variant of an existing proposal module with a new captioning module. The proposal module samples video frames at different strides and gathers evidence to propose events at different time scales in one pass of the video. The captioning module attends over the neighboring events, utilizing their context to improve the generation of captions. We compare variants of our model and demonstrate that context does indeed improve captioning. We further show how the captioning model uses context to improve video retrieval and how our proposal model uses the different strides to improve event localization. Finally, this paper also releases a new dataset for dense-captioning events: ActivityNet Captions.

Acknowledgements. This research was sponsored in part by grants from the Office of Naval Research (N00014-15-1-2813) and Panasonic, Inc. We thank Shyamal Buch, Jun Young Gwak, Timnit Gebru, Alvaro Soto, and Alexandre Alahi for their helpful comments and discussion.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–971, 2016.

[2] C. Baldassano, J. Chen, A. Zadbood, J. W. Pillow, U. Hasson, and K. A. Norman. Discovering event structure in continuous narrative perception and memory. *bioRxiv*, page 081018, 2016.

[3] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International journal of computer vision*, 74(1):17–31, 2007.

[4] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. C. 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.

[5] F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1914–1923, 2016.

[6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[7] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1491–1498. IEEE, 2009.

[8] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision*, pages 768–784. Springer, 2016.

[9] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2782–2795, 2013.

[10] G. Gkioxari and J. Malik. Finding action tubes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 759–768, 2015.

[11] D. B. Goldman, B. Curless, D. Salesin, and S. M. Seitz. Schematic storyboarding for video visualization and editing. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 862–871. ACM, 2006.

[12] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`, 2015.

[13] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3090–3098, 2015.

[14] D. Jayaraman and K. Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3852–3861, 2016.

[15] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[16] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[17] S. Karaman, L. Seidenari, and A. Del Bimbo. Fast saliency based pooling of fisher encoded dense trajectories. In *ECCV THUMOS Workshop*, volume 1, 2014.

[18] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[19] 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.

[20] J. Krause, J. Johnson, R. Krishna, and L. Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Patterm Recognition (CVPR)*, 2017.

[21] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *International Journal on Computer Vision (IJCV)*, 2017.

[22] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[23] W. Liu, T. Mei, Y. Zhang, C. Che, and J. Luo. Multi-task deep visual-semantic embedding for video thumbnail selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3715, 2015.

[24] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[25] 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.

[26] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *European conference on computer vision*, pages 392–405. Springer, 2010.

[27] D. Oneata, J. Verbeek, and C. Schmid. Efficient action localization with approximately normalized fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2552, 2014.

[28] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Learning joint representations of videos and sentences with web image search. In *European Conference on Computer Vision*, pages 651–667. Springer, 2016.

[29] 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.

[30] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 612–619, 2014.

[31] 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*, pages 184–195. Springer, 2014.

[32] A. Rohrbach, M. Rohrbach, and B. Schiele. The long-short story of movie description. In *German Conference on Pattern Recognition*, pages 209–221. Springer, 2015.

[33] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.

[34] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.

[35] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2642–2649, 2013.

[36] A. Vahdat, B. Gao, M. Ranjbar, and G. Mori. A discriminative key pose sequence model for recognizing human interactions. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1729–1736. IEEE, 2011.

[37] 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.

[38] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

[39] L. Wang, Y. Qiao, and X. Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1:2, 2014.

[40] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *European Conference on Computer Vision*, pages 565–580. Springer, 2014.

[41] W. Wolf. Key frame selection by motion analysis. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1228–1231. IEEE, 1996.

[42] H. Xu, S. Venugopalan, V. Ramanishka, M. Rohrbach, and K. Saenko. A multi-scale multiple instance video description network. *arXiv preprint arXiv:1505.05914*, 2015.

[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume 14, pages 77–81, 2015.

[44] R. Xu, C. Xiong, W. Chen, and J. J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, volume 5, page 6, 2015.

[45] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden markov model. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 379–385. IEEE, 1992.

[46] H. Yang, B. Wang, S. Lin, D. Wipf, M. Guo, and B. Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4633–4641, 2015.

[47] L. Yang, K. Tang, J. Yang, and L.-J. Li. Dense captioning with joint inference and visual context. *arXiv preprint arXiv:1611.06949*, 2016.

[48] 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.

[49] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 982–990, 2016.

[50] S. Yeung, A. Fathi, and L. Fei-Fei. Videoset: Video summary evaluation through text. *arXiv preprint arXiv:1406.5824*, 2014.

[51] 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.

[52] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658, 1997.