DAPs: Deep Action Proposals for Action Understanding

Victor Escorcia[1], Fabian Caba Heilbron[1],
Juan Carlos Niebles[2,3], Bernard Ghanem[1]

[1] King Abdullah University of Science and Technology (KAUST), Saudi Arabia.
[2] Stanford University. [3] Universidad del Norte, Colombia.
{victor.escorcia, fabian.caba, bernard.ghanem}@kaust.edu.sa,
jniebles@cs.stanford.edu,

Abstract. Object proposals have contributed significantly to recent advances in object understanding in images. Inspired by the success of this approach, we introduce *Deep Action Proposals* (DAPs), an effective and efficient algorithm for generating temporal action proposals from long videos. We show how to take advantage of the vast capacity of deep learning models and memory cells to retrieve from untrimmed videos temporal segments, which are likely to contain actions. A comprehensive evaluation indicates that our approach outperforms previous work on a large scale action benchmark, runs at 134 FPS making it practical for large-scale scenarios, and exhibits an appealing ability to generalize, i.e. to retrieve good quality temporal proposals of actions unseen in training.

Keywords: action proposals, action detection, long-short term memory.

1 Introduction

Nowadays, the ubiquity of digital cameras and social networks has increased the amount of visual media content (especially videos) generated and shared by people. In the face of this data deluge, it becomes crucial to develop efficient and scalable algorithms that can intelligently parse/browse visual data to discover semantic information. In this paper, we focus on the task of quickly localizing temporal chunks in untrimmed videos that are likely to contain human activities of interest. This is the well-known task of temporal action proposal generation. The detected temporal proposals can facilitate and speedup activity detection, indexing, and retrieval in long videos. For example, a "good" action proposal method can retrieve video snippets of a home-run being scored within a large corpus of baseball games or extract important moments during the construction of a new skyscraper. Motivated by the large-scale nature of the problem, we develop a temporal proposal algorithm that retrieves high fidelity proposals with a much smaller computational cost than previous methods (refer to Figure 1).

The idea of extracting regions with semantic content is not new in the computer vision community. Object proposals have proven to be one of the key elements in the current success of object detection at large scales, both in terms

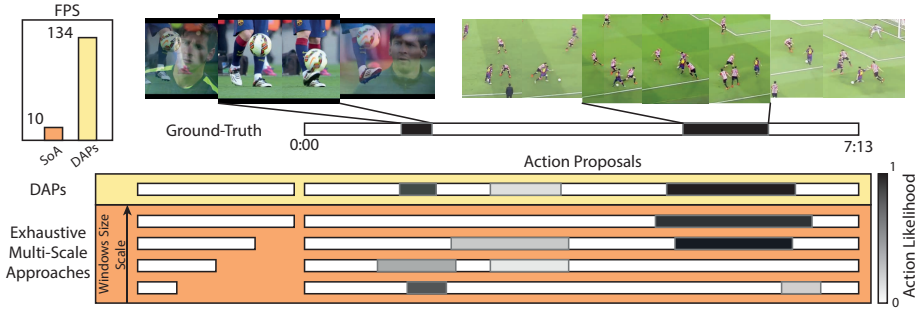

Fig. 1. An effective and efficient action proposal algorithm can localize segments of varied duration around actions occurring along a video without exhaustively exploring multiple temporal scales. This work shows how to produce high-quality temporal proposals likely to contain actions and to be 10x faster that the state of the art approach.

of efficiency and high detection rates [29, 28]. Efficient object proposal modules have also enabled a boost in performance of other high-level visual tasks, such as simultaneous detection and segmentation, object tracking, and image captioning [13, 14, 16, 20]. In order to push forward on high-level analysis of untrimmed videos, we argue that the development of action proposal methods should be put in the forefront of human activity understanding research.

Jain et al. [17] introduced the concept of action proposals by taking inspiration from object proposal methods in the image domain. Most previous action proposal approaches focused on producing spatio-temporal object proposals *i.e.* retrieving cuboids or tubelets containing actions [1, 6, 11, 24, 10, 39]. It is tempting to think that keeping the temporal part of these tubelets would result in good temporal segments confining actions. However, it was recently shown that the temporal footprint of some methods can be as accurate as sampling temporal proposals uniformly in the video [4]. Moreover, these methods evaluate their performance on simple or repetitive actions in short video clips, which makes it difficult to gauge their scalability to large collections of video sequences containing more challenging activities [18, 3]. Given the current state-of-the-art of spatio-temporal action proposals, it is worth exploring how only temporal action proposals can contribute to the semantic analysis of videos.

In fact, very recent work has explored the generation of temporal action proposals directly from videos [4, 22, 30]. Most of these approaches focus on exploring a large number of regions in the video at multiple scales (i.e. temporal lengths) and selecting among them proposals through an efficient feature extraction and classification pipeline. Unlike these methods and as illustrated in Figure 1, we propose an effective and efficient approach that leverages the capacity of deep learning models with memory blocks to extract action proposals at different temporal scales in only one pass through the video. This is done by encoding a

video sequence as a discriminative sequence of states, from which action likely segments can be localized with varied duration inside a video sequence.
Contributions: **(i)** We propose a new approach for temporal action proposal generation, specifically targeting long videos. This is done by training a well-suited memory network to reliably output the temporal location and scale of a fixed number of proposals. **(ii)** Our model is able to generate proposals of multiple temporal scales with a single pass through the video and to generalize well to new unseen actions. **(iii)** Extensive experiments on large-scale benchmarks show that our method achieves a better recall than other proposal methods. **(iv)** Our approach is computationally efficient and runs at 134 FPS.

1.1 Related Work

We summarize the most recent work on topics related to the task of action proposal generation and our proposed methodology.
Object Proposals: Exhaustively running computationally intensive object classifiers with a sliding window approach is not as common as it was eight years ago. Instead, the use of generic or class-specific object proposals is now a cornerstone in the object detection pipeline. These proposal algorithms retrieve high-quality candidate regions that are likely to contain an object (high recall), before classification is performed [8, 29, 15]. This approach has proven to be an effective and scalable way to find possible locations of an object in an image.

The latest trend in this area is designing algorithms with high ranking quality *i.e.* achieving high object recall with less number of bounding boxes, preferably with a small computational overhead and the potential to scale to hundreds of object categories [35, 40, 37]. Here, discriminative methods based on deep learning models have helped improve the ranking quality of proposal approaches [7, 37, 28, 32]. Inspired by this work, we extend the use of deep and recurrent networks to temporal action proposal generation by introducing a new architecture.
Action Detection: In contrast to object detection methods, the dominant approach for action detection is still to use a sliding window approach [26, 18, 12] combined with action classifiers trained on multiple features [2, 9, 33]. Previous approaches have reduced the computational overhead of sliding window search by using branch-and-bound techniques [5, 27] and exploiting some characteristics of the visual descriptors. In contrast, our model efficiently reduces the number of evaluated windows by encoding a sequence of visual descriptors.
Spatio-Temporal Action Proposals: Recently, ideas from the area of object proposals have been extrapolated to action recognition in the video domain [6, 17, 11, 21, 24, 10, 39]. Most of these methods produce spatio-temporal object segments to perform spatio-temporal detection of simple or cyclic actions on short video sequences, hence their scalability to real-world scenarios is uncertain. These methods rely on straddling of voxels [6, 17], reasoning over dense trajectories [24, 39], or non real-time object proposals [11], which increase their computational cost and reduce their competitiveness at large scales.
Temporal Action Proposals: Very recently, work emerged that focused on temporal segments which are likely to contain human actions [4, 22, 30]. Similar

to grouping techniques for retrieving object proposals, Mettes *et al.* create a hierarchy of fragments by hierarchical clustering based on semantic visual similarity of contiguous frames [22]. The main disadvantages of this approach are its strong dependence on an unsupervised grouping method that diminishes its repeatability [15] and the absence of an *actioness* score for each fragment in the hierarchy. In comparison, we use a supervised method that learns to generate segments on a video and predict their action likelihood. Most closely related with our approach are methods that use category-independent classifiers to explore many segments in video and exhaustively evaluate segments of multiple temporal scales [4, 30]. Our method improves over previous ones by using a powerful deep learning model that allows for less windows to be scanned and multiple temporal scales to be considered simultaneously in a single pass through the video. We leverage long-short term memory cells to learn an appropriate encoding of the video sequence as a set of discriminative states. We experimentally show that this representation is able to regress the temporal location and duration of relevant segments on the original sequence, while running at 134 FPS.

2 Our Approach: Deep Action Proposals

We propose a new *Deep Action Proposals* (DAPs) network for the task of temporal action proposal generation. From a long input video sequence, we aim to retrieve temporal segments that likely contain actions of interest. Figure 2 summarizes our model architecture, which is described in detail in Section 2.1. Section 2.2 describes the training and inference procedures.

2.1 Architecture

Our DAPs network encodes a stream of visual observations of length T frames into discriminative states, from which we infer the temporal location and duration $\{s_i\}_{i=1}^{K}$ of K action proposals inside the stream. Each proposal s_i is associated with a confidence score c_i. Our network integrates the following modules:
Visual encoder: It encodes a small video volume into a meaningful low dimensional feature vector. In practice, we use activations from the top layer of a 3D convolutional network trained for action classification (C3D network [34]).
Sequence encoder: It encodes the sequence of visual codes as a discriminative sequence of hidden states. Here, we use a long-short term memory (LSTM) network. In contrast to traditional feed-forward layers, it directly models the sequential information in a principled and effective manner [31, 23].
Localization module: It predicts the location of K proposals inside the stream based on a linear combination of the last state in the *sequence encoder*. In this way, our model can output segments of different lengths in one pass instead of the traditional way of scanning over overlapping segments with multiple window sizes. Each proposal s_i is predicted by the localization module.
Prediction module: It predicts the confidence c_i that proposal s_i contains an action within its temporal extent. In practice, c_i is the output of a sigmoid function over a linear combination of the last state of the sequence encoder.

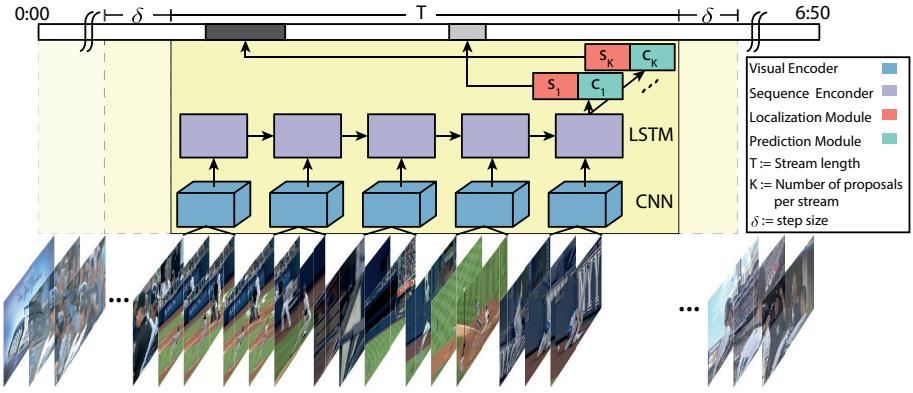

Fig. 2. Our Deep Action Proposals (DAPs) architecture effectively encodes a stream of visual observations (of length T frames) into discriminative states from which it is possible to localize K proposals $\{s_i\}_{i=1}^K$ with confidences $\{c_i\}_{i=1}^K$ inside the stream. We generate several segments where it is possible to find actions along a video sequence by sliding it with step size δ.

2.2 Inference and Learning

Inference: In order to produce several candidate segments where actions are likely within a long video sequence, we slide our DAPs network over it with step size δ. Every time our model scans a video stream of length T frames, it places K segments of varied duration inside it with their respective action likelihoods. In contrast with previous approaches that scan the same clip of video with multiple sized windows, we encode the information of the clip in order to improve efficiency at inference time. In that way, our algorithm scans the whole video sequence in *only* one pass with one stream (or window) size T, while still producing segments of different duration.

Learning: Another way to interpret our DAPs network is in the form of a function f that maps a video stream v (of length T frames) onto a set of K segments inside the stream with their respective action likelihood. Formally, we have $(S, C) = f_{K,T}(v; \theta)$ where $S = \{s_i\}_{i=1}^K$ and $C = \{c_i\}_{i=1}^K$ represent the set of all predicted segments and their action likelihoods, respectively. Here, θ represents the parameters of our model.

We are interested in learning an appropriate function f such that: (i) segments produced by our model match the locations of actions $A = \{a_i\}_{i=1}^M$ in the sequence (the number of these actions in stream v is assumed less than K); and (ii) confidence values associated with segments that match an action are higher than other segments. This is done by formulating an assignment problem, which solves for an optimal matching between predictions from our DAPs function and ground truth action annotations in the video stream. Without loss of generality,

for each training segment, we solve the following problem:

$$(\mathbf{x}^*, \theta^*) = \underset{\mathbf{x},\theta}{\operatorname{argmin}} \quad \alpha \mathcal{L}_{\text{match}}(\mathbf{x}, S(\theta), A) + \mathcal{L}_{\text{conf}}(\mathbf{x}, C(\theta)) \\ \text{s.t.} \quad x_{ij} \in \{0,1\}, \quad \sum_i x_{ij} = 1 \tag{1}$$

where $x_{ij} = 1$ means that the i-th prediction s_i is assigned to the j-th ground truth annotation a_j. Here, we define $\mathcal{L}_{\text{match}}(\mathbf{x}, S(\theta), A)$ to be a function that penalizes (in the form of a Euclidean distance) matched segments that are distant from action annotations. Also, we take $\mathcal{L}_{\text{conf}}(\mathbf{x}, C(\theta))$ to enforce (in the form of binary cross-entropy) that the likelihood of matched segments be as high as possible, while simultaneously penalizing non-matched segments that occur with high likelihood. Finally, α is a tradeoff constant that combines both terms. This problem can be solved by alternating between solving the assignment problem for a given θ^k and back-propagating errors given an optimal assignment $\mathbf{x}^k$.

For simplicity we rely on a heuristic similar to [7] to relax the assignment problem by introducing K anchor segments $L = \{l_i\}_{i=1}^K$. In this way, we guide the localization module of the network towards K anchor segments summarizing the statistics of the annotations. This approach speeds up the optimization by: (i) guiding the learning towards statistically relevant locations; and (ii) solving the assignment problem up-front *i.e.* for every instance v we compare the predictions of our function (S, C) with (L, Y), where $Y = \{y_i\}_{i=1}^K, y_i \in \{0, 1\}$ defining that the *i-th* anchor segment matches a ground-truth annotation of the instance.

In practice, we obtain the location and duration of each anchor proposal by clustering the ground-truth annotations with k-means which gives rise to a diverse set of anchors throughout the stream. More details about the optimization problem are provided in the supplementary material.

Implementation Details: for our visual encoder, we use the publicly available pre-trained C3D model[34] which has a temporal resolution of 16 frames. To shorten the training time of our implementation, we reduce the dimensionality of the activations from the second fully-connected layer (*fc7*) of our visual encoder from 4096 to 500 dimensions using PCA. By cross-validation, we find that one layer and 256 output units achieves a good trade-off between accuracy and runtime. We use back-propagation through time with ADAGRAD update rule to find the parameters θ of our sequence encoder and output modules. By hyperparameter search, a learning rate of 10^{-4} and $\alpha = 1.0$ provide good results. In practice, we predict locations (s) as duration of the action and the frame index of its center (normalized by T).

The DAPs network is trained on video streams of length T frames from long untrimmed videos. From a labeled dataset like *THUMOS-14* with 11 hours of video and more than 3000 annotations, we are able to generate a large corpus of video streams (over 500 thousands) that might contain multiple actions. In practice, we densely extract video streams and cluster them according to their *tIoU* with annotations of the video. We sample streams from each cluster, so they are equally represented.

3 Experiments and Discussion

3.1 Experimental Setup

We validate the quality of our approach on labeled untrimmed videos from the challenging *THUMOS-14* benchmark, which contains over 24 hours of video from about 20 sport action categories. This part comprises 413 videos divided into 200 validation videos and 213 test videos. We train our DAPs model using 180 out of 200 videos from the validation set and hold out 20 videos for validation. We report results on the 213 test videos with temporal annotations. To study the generalization capability of our model across datasets, we also test on the validation set of the *ActivityNet* benchmark (release 1.2)[3], which comprises 76 hours of video and 100 action classes. No fine tuning is done on this benchmark.
Metrics. We assess the quality of our temporal proposals with the metrics from [15]. Specifically, we use *Average Recall* (AR) to measure the temporal proposal quality for a limited number of proposals. We compute AR for a *tIOU* between 0.5 to 1, as a function of the number of proposals. We expect the best proposal approach to achieve the best recall by generating tight temporal proposals at a fixed number of proposals. We also measure the recall at a fixed number of proposals, as a function of *tIoU*. This metric measures the localization quality of temporal proposals. We consider 1000 proposals for this.

In Section 3.3, we investigate the impact of applying action proposals in the context of action detection. Following the standard evaluation protocol, we measure the *mean Average Precision* (mAP) at 50% *tIoU*. We use the official toolkit provided by *THUMOS-14* [18].

3.2 Recall Analysis

In this section, we analyze recall performance of our method. Specifically, we study (i) the performance of variants of our approach, (ii) the performance competing temporal proposal methods, and (iii) the ability of our approach to generalize to actions that are unseen during training.
Variants of our approach. We evaluate the effect of hyper-parameters on our DAPs model on 20 videos from the *THUMOS-14* validation set. Figure 3 plots AR (first and third columns) and Recall at 1000 proposals (second and fourth columns) of our algorithm for different numbers of proposals per stream (K) and four different stream lengths (T).

As Figure 3 shows (two leftmost columns), our model is not very sensitive to the number of anchor proposals K for a stream length $T = 512$ frames. Our experiments show that larger K does not necessarily translate into better performance. We hypothesize that this behavior is a result of using k-means to select the anchors. This result suggests that the difference between selecting multiple anchors per segment might not be predictable, so we resort to choose this hyper-parameter by cross-validation. We choose $K = 64$ for the rest of our experiments, as a reasonable tradeoff between capacity and AR. In fact, our DAPs model with $K = 64$ achieves the highest average recall rate for more than

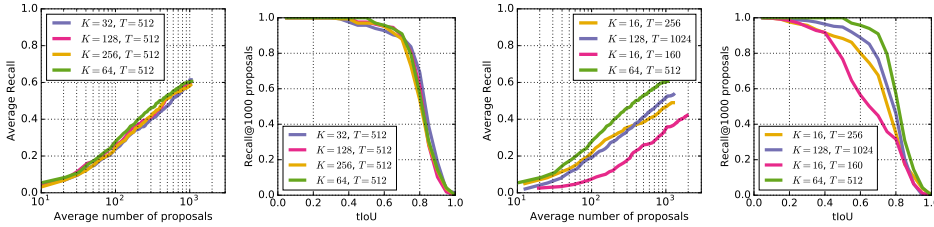

Fig. 3. We evaluate the effect of the hyper-parameters of our approach on a held-out portion of the validation set of *THUMOS-14*. We find that the performance of our model is stable with respect to the number of proposals per stream K (leftmost columns). On the other hand, we find that the choice of the stream length T is more critical (rightmost columns).

100 proposals and about 100% recall at a 50% *tIoU* with 1000 proposals, as shown in Figure 3 (second columns).

Next, we assess the impact of the stream length T on the performance of our architecture. We evaluate with $T \in \{160, 256, 512, 1024\}$ frames which covers $\{75, 92, 98, 99\}$% of the annotations in the validation set respectively. The results suggest that T is a crucial hyper-parameter for achieving high recall. From Figure 3 (rightmost column), we find that for *tIoU* of 50% at 1000 proposals the recall correlates with statistics of annotations. Therefore, we conclude that our model learns correctly to retrieve actions inside the range of T. Based on this analysis, we choose a value of $T = 512$ frames for other experiments.

In the experiments to follow, we report the results of our DAPs algorithm with $K = 64$, $T = 512$ which offers a good trade-off between accuracy, scalability, and run-time performance.

Comparison with other approaches. We compare the performance of our algorithm against recent approaches designed to retrieve temporal proposals, namely *Sparse-prop* [4], *BoFrag* [22], and *SCNN-prop* [30]. For completeness, we also compare to a representative spatio-temporal proposal method, *APT* [10]. For a fair comparison, we project *APT* spatio-temporal proposals to the temporal dimension only. We obtain *APT* results by running the public implementation provided by the authors. For all other methods, temporal proposals were kindly provided by the authors.

Figure 4 illustrates the AR and recall of 1000 proposals of all five methods on the *THUMOS-14* benchmark. Clearly, our DAPs significantly outperforms all other methods in both metrics. We hypothesize that it improves upon them by effectively encoding the sequence of visual codes as a discriminative set of states from where it is plausible to regress proposals with multiple durations. Notably, our DAPs algorithm boosts AR at 1000 proposals and recall of 1000 proposals at 50% *tIoU* to 58.1% and 95.7%, respectively. The later represents a relative improvement in recall of 27.2% over the *SCNN-prop* [30], which also trains a network to match temporal segmentation (based on *tIoU*) with ground truth annotations. Note that our approach achieves a better performance without

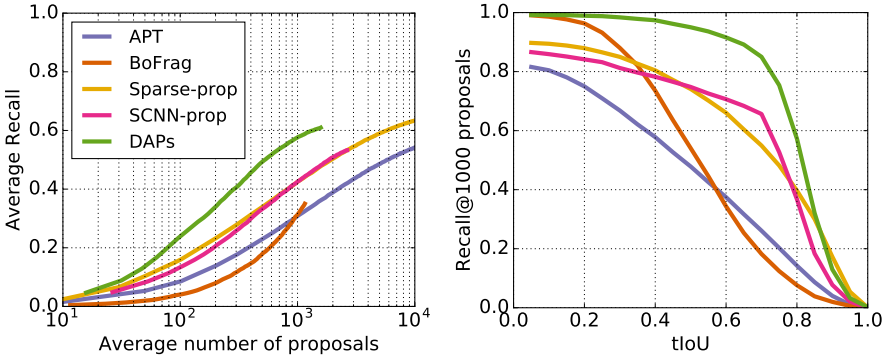

Fig. 4. Our DAPs network outperforms previous temporal and spatio-temporal approaches on *THUMOS-14* in terms of Average Recall as well as in terms of recall of 1000 proposals for a wide range of *tIoU*. This result evidences the importance of effectively encoding the visual sequence as a discriminative sequence of states in relation with previous approaches.

exhaustively exploring multiple rigid temporal window sizes which suggests that our network is effectively encoding multiple action durations instead of sticking to a fixed length. It is worth to notice that the AR of DAPs with 1.6k proposals is better or comparable to the AR of *APT* and *SparseProp* for 10k proposals. We envision clever innovations of DAPs architecture to increase the number of proposals maintaining the same quality. Figure 4 (right) shows that our approach is the best to generate segments tightly localized around the actions up to 85% *tIoU*. From this point, it is interesting that algorithms like DAPs and *SCNN-prop* exhibit a greater decreasing slope than other algorithms. We guess that this effect is partly due to the use of *tIoU* to define the supervisory signals.

On the other hand, we find that all supervised methods outperform the unsupervised ones (*BoFrag* [22] and *APT* [10]) by a considerable margin, especially at high *tIoU* values. This suggests that supervised methods are not over-fitting over their training set and they learn a good function to measure action likelihood. We believe that such *actioness* function may help to boost the performance of unsupervised approaches, especially on methods that do not provide an action likelihood score for each segment, like *BoFrag* and *APT*.

Is the network able to generalize the concept of an action? Proposal approaches are similar to classifier cascades in the sense that they reduce the computational cost of evaluating powerful classifiers on regions that can be "easily" rejected [36]. According to Hosang *et al.* [15], the main difference between these methods is that classifier cascades do not necessarily exhibit an ability to generalize beyond the categories they are trained on. Along these lines, we study the generalization capabilities of our DAPs network to validate that it is a proper proposal approach. We do that by applying our model, trained on 20 sports categories from *THUMOS-14*, on *ActivityNet*, a rich and diverse dataset in terms of actions. For example, just nine actions from *THUMOS-14* have a

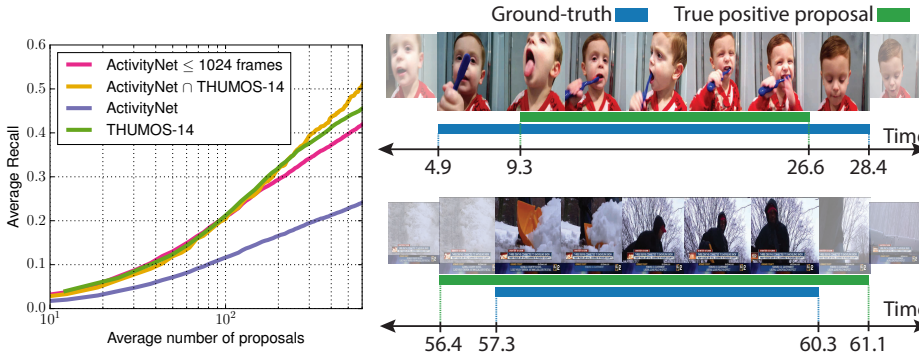

Fig. 5. We measure the generalization power of our DAPs network across dataset for unseen actions by evaluating its performance on *ActivityNet*. Interestingly, the AR performance of our network does not decrease significantly, at 600 proposals, on videos where action durations comes from a similar distribution, *ActivityNet* $\leq$ *1024 frames* line. This suggests that discriminative sequence of states learned by our model capture common patterns that allows it to localize and score segments of unseen actions. On the right, we appreciate segments retrieved by our method focus on *brushing teeth* and *shoveling snow* actions, clearly not related with any sport.

reasonable correspondence with the hundred activities in *ActivityNet*. Moreover, this dataset includes many categories unrelated with sports, such as *Preparing pasta*, *Playing saxophone*, *Shoveling snow*, to name a few.

Figure 5 (left) quantitatively summarizes the generalization capability of our approach. We show average recall results of our method on four datasets: *ActivityNet* (all 100 categories), *ActivityNet* $\cap$ *THUMOS-14* (on 9 categories shared between both benchmarks), *ActivityNet* $\leq$ *512 frames* (videos of unseen categories with annotations up to 1024 frames), and *THUMOS-14*. By comparing the performance on *ActivityNet* and *THUMOS-14*, the generalization power of DAPs might not seem encouraging. However, we find that 42% of the activity annotations in *ActivityNet* span more than 1024 frames (i.e. twice the size of our temporal stream T), hence it will be difficult for our model to achieve a high AR in this scenario. Since the distribution of activity durations in *ActivityNet* is very different to the one in *THUMOS-14*, a drop in recall performance is not surprising. In fact, Hosang *et al.* make a very similar observation in the context of generalizing 2D object proposals in images across datasets [15].

Following up on this observation, we study the performance of our approach on *ActivityNet* $\cap$ *THUMOS-14*, where we only consider annotations from common classes seen in training; and *ActivityNet* $\leq$ *512 frames*, where we only consider annotations of unseen classes that have similar duration statistics observed in *THUMOS-14*, *i.e.* annotations that span up to 512 frames. When evaluating on these two datasets, DAPs performance is quite similar in both cases, especially when more proposals are retrieved. In fact, it achieves an AR of 50.9% and 41.9% for 600 proposals respectively, which are close to our performance on

THUMOS-14 for the same number of proposals. This suggests that DAPs does exhibit a desired level of generalization for unseen actions. Note that, *ActivityNet* videos are 50% shorter than *THUMOS-14* videos on average so it is natural that our method produces less number of proposals.

Figure 5 (right) shows qualitative examples of temporal proposals retrieved by our network for activities not related to action categories used in training. We hypothesize that the network can generalize to these activities by discovering common underlying patterns in the encoded visual sequence that helps it to localize a proposal, as well as, score its likelihood.

3.3 DAPs for Action Detection

Inspired by the success of object detection approaches in combining object proposal methods with object classifiers, we study the benefit of applying our temporal action proposals in an action detection pipeline. To this end, we classify the action proposals generated by our approach and competing proposal methods using the same state-of-the-art action classifier trained on *THUMOS-14* [38]. In this section, we describe the action classifier, assess the impact of the number of proposals on detection performance, and compare our method against state-of-the-art approaches.

Action Classifier. Here, we adopt the recent approach of Xu *et al.* [38], which encodes features learned by a conv-net model using *VLAD*. Here, we use the activations from the *fc7* layer from a 3D conv-net [34] as our features. We first learn a codebook using k-means with $k = 256$. Then, we encode the *fc7* features that belong to each temporal segment using *VLAD* with power and L_2-normalization. Finally, we train a *one-vs-all* linear SVM classifier with $C = 100$. At test time, we run our activity classifier over all the generated action proposals and obtain an action confidence score for each of them. We apply non-maximum suppression with a 30% *tIoU* to eliminate near-duplicate detections. As in common detection procedures, we generate a final prediction score by multiplying the classifier and proposal scores.

Detection results. Table 1 shows quantitative detection results comparing our proposal approach against competing methods. Following action detection convention, we report the mAP (mean AP) score at 50% *tIoU*. We consistently outperform the competing methods by a significant margin. This substantiates our claim that our method produces high-quality proposals with a budgeted number of proposals.

Interestingly, *BoFrag* generates good localization results despite its modest recall performance. This suggests that *BoFrag* is producing proposals with a small number of hard negatives, which allows the activity classifier to keep the number of false positives low. We also observe that all methods tend to saturate after using more than 500 proposals. This is in part due to the fact that all proposal methods are decoupled with the final action classifier. Therefore, it is plausible to fluster the action classifier when the ratio of true positives starts to decrease.

Table 1. Results for action detection experiments on *THUMOS-14*. We evaluate the performance of different proposal methods using mAP at P (mAP@P) number of proposals. The *tIoU* threshold for a correct detection is fixed to 50% and "-" is used when a method is not able to produce the P number of proposals. Our method outperform competing methods by a significant margin for all number of proposals.

Method	mAP@50	mAP@100	mAP@200	mAP@500	mAP@1000
APT	4.1	5.2	6.2	6.8	6.4
BoFrag	5.3	6.6	7.0	8.5	8.3
SCNN-prop	5.3	5.6	7.8	-	-
Sparse-prop	5.7	6.3	7.6	8.2	8.0
DAPs	**8.4**	**12.1**	**13.9**	**12.5**	**12.0**

Table 2. Action detection state-of-the-art on *THUMOS-14*. We report the detection performance of our method at 200 proposals. Our method is able to achieve a competitive performance using a very limited number of proposals.

Method	Karaman *et al.* [19]	Caba Heilbron *et al.* [4]	Oneata *et al.* [25]	Shou *et al.* [30]	**Ours**
mAP	2.0	13.5	15.0	19.0	13.9

State-of-the-art comparison. Table 2 summarizes the action detection state-of-the-art on *THUMOS-14*. Our method achieves a significantly higher performance than Karaman *et al.* [19] which uses sliding window with a unique fixed temporal length. We attribute this improvement to the fact that our approach scans the video in a much more efficient way. We obtain a similar performance to Caba Heilbron *et al.* [4] and Oneata *et al.* [25]. This result is encouraging given that our detection pipeline operates at a much faster rate of 134 FPS. As compared to Shou *et al.* [30] (*SCNN-prop*), our results are promising considering that less number of windows are scanned to produce the final detection. In future work, we plan to combine directly our proposal network with the classification stage, as well as, fine tune the parameters of the C3D network to achieve further improvement in our detection results.

3.4 Run-time Performance

By definition, action proposals should reduce the effort of applying an accurate and computationally expensive classifier on a large number of windows in a video. This means that a good action proposal method is expected to achieve a high recall rate in the shortest amount of time possible. Table 3 summarizes the run-time performance of different proposal methods. Specifically, we compute the average run-time over all testing videos on a Titan-X GPU and report the time in terms of the average length of videos in *THUMOS-14* (3 minutes). The authors of other methods kindly provided the run-time of their approach.

Table 2 shows that our algorithm is the fastest method to generate temporal action proposals. This is due to: (i) an effective and efficient window scanning approach; and (ii) the use of hardware acceleration units (GPUs) to speed-up computation. A preliminary comparison with *SCNN-prop*, which also benefits from GPUs, shows a relative improvement of 123.5%. Disregarding implementation details that can increase the performance of both approaches, the improve-

Table 3. Our DAPs network is the fastest action proposal method. We report the average time needed to apply DAPs to an average length video from *THUMOS-14* (3 minutes). Methods that we could not benchmark appear with "-", while N.A. refers to methods that do not require a specific stage (see text for more details).

Algorithm	Time [seconds]			Speedup	FPS
	Feature	Proposal	Total		
APT	2828.5	5120.3	7948.8	1.0	0.68
BoFrag	**90**	5.5	95.5	83.23	1.88
Sparse-prop	191.1	342.5	533.6	14.9	10.2
SCNN-prop	N.A	-	-	-	60
DAPs	N.A	**1.34**	**1.34**	**5931.9**	**134.1**

ment on speed-up is a consequence of an effective encoding that reduces the exploration of multiple temporal scales on overlapping regions.

3.5 Qualitative results

Figure 6 shows the top ranked proposal retrieved from videos of *THUMOS-14* as well as two sample videos with the best-matched proposals out of 100. We include examples where our method succeeds (True positive proposals) and fails (False positive proposals) to match the ground truth with a *tIoU* of 50%. We observe that our method can produce tight segments around actions. We detect several failure cases in actions like *Shot put* where either the annotation is ambiguous or is hard to establish the temporal boundaries of the action. Interestingly, our method can retrieve segments semantically relevant around miss-labeled or incomplete actions in *THUMOS-14*. For example, the fourth row in Figure 6 shows a proposal that matches an action where a woman is trying to perform *Pole vault* but fails.

4 Conclusion and Future Work

We present Deep Action Proposals (DAPs), an effective and efficient network that produces temporal segments over a long video sequence where it is likely to find human actions. A comprehensive evaluation shows that our approach not only produces high-quality segments in relationship to the state of the art, it also is the fastest method. A follow-up version of this work will formulate an end-to-end version of our approach in order to fine-tune the low-level filters of the *C3D* architecture for the task of agnostic action localization. Similarly, we expect to design novel architectures that reduce the computational footprint of the current approach and increase the quality of the segments retrieved for a large variety of activity lengths.

Acknowledgments. Research in this publication was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research, the Stanford AI Lab-Toyota Center for Artificial Intelligence Research and a Google Faculty Research Award (2015).

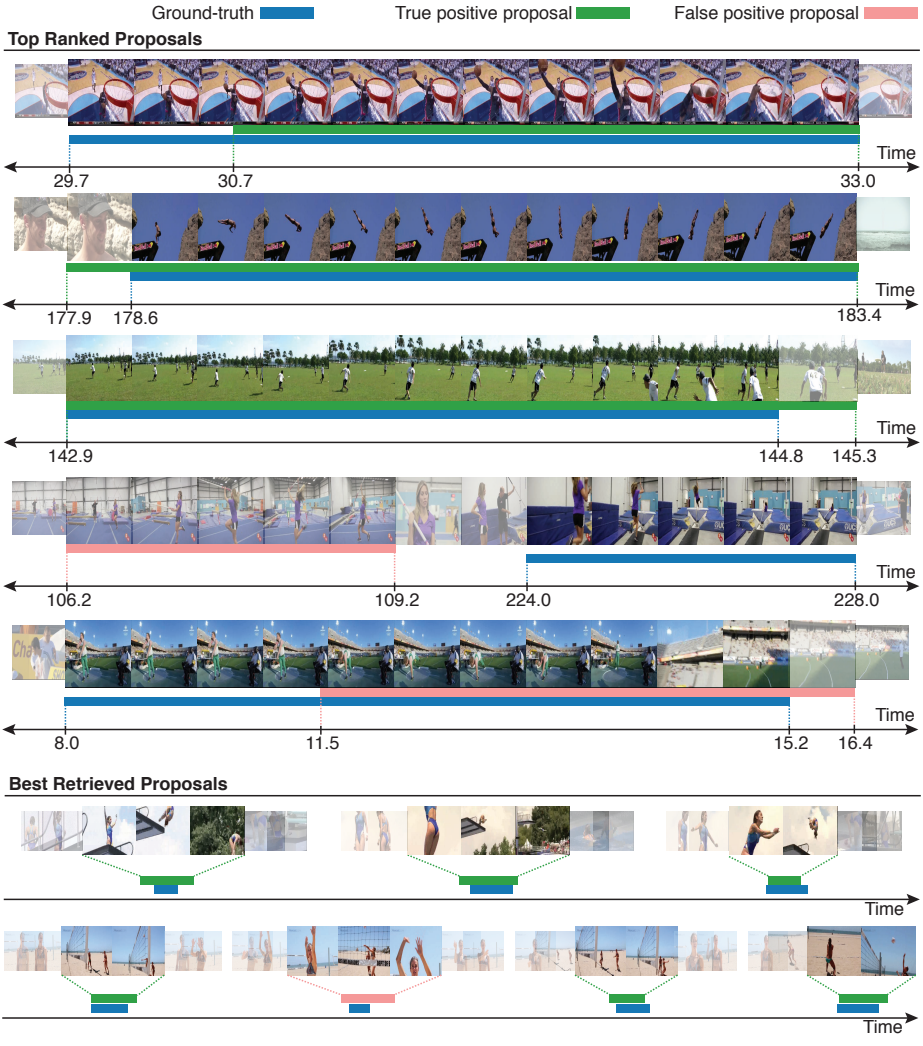

Fig. 6. Qualitative examples of retrieved segments by DAPs algorithm on sample videos from *THUMOS-14*. The first five rows show the top ranked proposal, its nearest ground truth action and the corresponding mapping to time (seconds). The first three rows show examples where our approach generates tightly segments around action instances. On the other hand, the next two rows correspond to failures modes of our model such as an unlabeled occurrence of an incomplete action (fourth row). The last two row visualize the best-matched segments retrieved in two different videos by DAPs out of 100 proposals.

References

1. Atmosukarto, I., Ahuja, N., Ghanem, B.: Action recognition using discriminative structured trajectory groups. In: 2015 IEEE Winter Conference on Applications of Computer Vision. pp. 899–906. IEEE (2015)
2. Atmosukarto, I., Ghanem, B., Ahuja, N.: Trajectory-based fisher kernel representation for action recognition in videos. In: Pattern Recognition (ICPR), 2012 21st International Conference on. pp. 3333–3336. IEEE (2012)
3. Caba Heilbron, F., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 961–970 (2015)
4. Caba Heilbron, F., Niebles, J.C., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2016)
5. Chen, C., Grauman, K.: Efficient activity detection with max-subgraph search. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1274–1281 (2012)
6. Chen, W., Xiong, C., Xu, R., Corso, J.J.: Actionness ranking with lattice conditional ordinal random fields. In: IEEE Conference on Computer Vision and Pattern Recognition CVPR. pp. 748–755 (2014)
7. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 2147–2154 (2014)
8. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision (IJCV) 111(1), 98–136 (Jan 2015)
9. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization of actions with actoms. IEEE transactions on pattern analysis and machine intelligence 35(11), 2782–2795 (2013)
10. van Gemert, J.C., Jain, M., Gati, E., Snoek, C.G.: Apt: Action localization proposals from dense trajectories. In: British Machine Vision Conference (BMVC) (2015)
11. Gkioxari, G., Malik, J.: Finding action tubes. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 759–768 (2015)
12. Gorban, A., Idrees, H., Jiang, Y.G., Roshan Zamir, A., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/ (2015)
13. Gupta, S., Girshick, R., Arbeláez, P., Malik, J.: Learning rich features from rgb-d images for object detection and segmentation. In: Computer Vision - ECCV 2014, pp. 345–360. Lecture Notes in Computer Science, Springer International Publishing (2014)
14. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Computer vision - ECCV 2014, pp. 297–312. Lecture Notes in Computer Science, Springer International Publishing (2014)
15. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? PAMI (2015)
16. Hua, Y., Alahari, K., Schmid, C.: Online object tracking with proposal selection. In: IEEE International Conference on Computer Vision (ICCV) (2015)
17. Jain, M., van Gemert, J.C., Jégou, H., Bouthemy, P., Snoek, C.G.M.: Action localization with tubelets from motion. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 740–747 (2014)

18. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (2014)
19. Karaman, S., Seidenari, L., Del Bimbo, A.: Fast saliency based pooling of fisher encoded dense trajectories (2014)
20. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Advances in Neural Information Processing Systems (NIPS). pp. 3128–3137 (2014)
21. Lillo, I., Carlos Niebles, J., Soto, A.: A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
22. Mettes, P., van Gemert, J., Cappallo, S., Mensink, T., Snoek, C.: Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting. In: ACM International Conference on Multimedia Retrieval (ICMR) (2015)
23. Ng, J.Y., Hausknecht, M.J., Vijayanarasimhan, S., Vinyals, O., Monga, R., Toderici, G.: Beyond short snippets: Deep networks for video classification. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 4694–4702 (2015)
24. Oneata, D., Revaud, J., Verbeek, J.J., Schmid, C.: Spatio-temporal object detection proposals. In: Computer Vision - ECCV 2014, pp. 737–752. Lecture Notes in Computer Science, Springer International Publishing (2014)
25. Oneata, D., Verbeek, J., Schmid, C.: The lear submission at thumos 2014 (2014)
26. Oneata, D., Verbeek, J.J., Schmid, C.: Action and event recognition with fisher vectors on a compact feature set. In: IEEE International Conference on Computer Vision, ICCV. pp. 1817–1824 (2013)
27. Oneata, D., Verbeek, J.J., Schmid, C.: Efficient action localization with approximately normalized fisher vectors. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 2545–2552 (2014)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS) (2015)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015)
30. Shou, Z., Wang, D., Chang, S.: Action temporal localization in untrimmed videos via multi-stage cnns. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2016)
31. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 3104–3112 (2014), http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks
32. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. CoRR abs/1412.1441 (2014), http://arxiv.org/abs/1412.1441
33. Tang, K., Yao, B., Fei-Fei, L., Koller, D.: Combining the right features for complex event recognition. In: The IEEE International Conference on Computer Vision (ICCV) (December 2013)
34. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE International Conference on Computer Vision, ICCV. pp. 4489–4497 (2015)

35. Uijlings, J.R.R., van de Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective search for object recognition. International Journal of Computer Vision 104(2), 154–171 (2013), http://dx.doi.org/10.1007/s11263-013-0620-5
36. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPRn. vol. 1, pp. I–511 (2001)
37. Weicheng Kuo, Bharath Hariharan, J.M.: Deepbox:learning objectness with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015)
38. Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative cnn video representation for event detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2015)
39. Yu, G., Yuan, J.: Fast action proposals for human action detection and search. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 1302–1311 (2015)
40. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: Computer vision - ECCV 2014, pp. 297–312. Lecture Notes in Computer Science, Springer International Publishing (2014)