Combining Online Clustering and Rank Pooling Dynamics for Action Proposals

Nadjia Khatir[1], Roberto J. López-Sastre[2], Marcos Baptista-Ríos[2], Safia Nait-Bahloul[1], and Francisco Javier Acevedo-Rodríguez[2]

[1] LITIO, Dept. of Computer Science, University of Oran1, Ahmed Ben Bella, Algeria
[2] GRAM, University of Alcalá, Alcalá de Henares, Spain
http://agamenon.tsc.uah.es/Investigacion/gram/

Abstract. The action proposals problem consists in developing efficient and effective approaches to retrieve, from untrimmed long videos, those temporal segments which are likely to contain human actions. This is a fundamental task for any video analysis solution, which will struggle to detect activities in a large-scale video collection without the proposals step, needing hence to apply an action classifier at every time location, in a temporal sliding window strategy, a pipeline which is clearly unfeasible. While all previous action proposals solutions are supervised, we introduce here a novel strategy that works in an unsupervised fashion. We rely on an online agglomerative clustering algorithm to build an initial set of proposals/clusters. Then a novel filtering approach is proposed, which uses the dynamics of the proposals discovered by the clustering, to measure their actioness, and proceeds to filter them accordingly. Our experiments show that our model improves the supervised state-of-the-art approaches when the number of proposals is controlled.

Keywords: action proposals, unsupervised learning, clustering, computer vision, action recognition

1 Introduction

In this work, we focus on the problem of localizing temporal segments in untrimmed videos that are likely to contain human actions. This is the well-known problem of action proposals, *e.g.* [1,2,3,4,5,6]. These proposals can speed-up activity recognition and detection tasks, as well as retrieval and indexing in long videos. Interestingly, in the last ActivityNet challenge [7], *all* the methods for the task of action localization [6,8,9] have tackled the problem following a two-stage pipeline, where the first step always consists in using an action proposal method. So, action proposals are important.

Typically, all action proposals solutions follow a supervised approach during learning. That is, models are trained using fully annotated datasets, such as THUMOS-14 [10], where for each video the time slots corresponding to each action are specified. However, the problem has not been addressed from an *unsupervised* perspective, where action proposal models can be trained without using

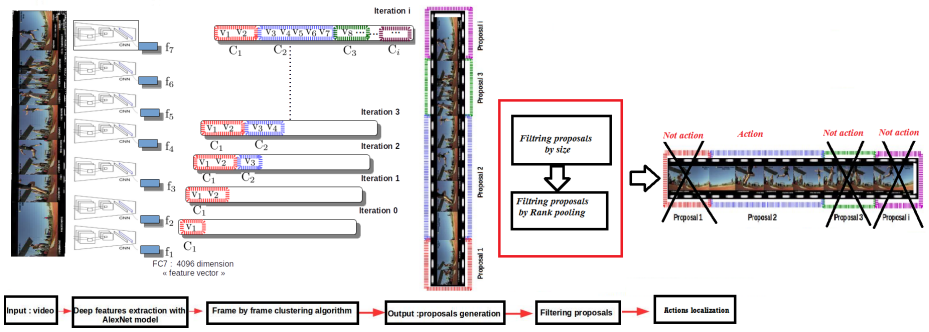


Fig. 1: We propose an unsupervised approach for action proposals. It is based on an online clustering model which works on deep features designed for object recognition. These cluster are later refined by their sizes and dynamics, using a rank-pooling based mechanism.

this information. This new approach offers significant benefits. The first one is that it is not necessary to annotate videos to perform the training. Moreover, the training data becomes unlimited, and data sources such as YouTube can be used, a factor that will lead to solutions that will be able to offer a better generalization capability.

Therefore in this paper we explore if such an unsupervised perspective is viable, offering the following **contributions**. **1)** We propose an action proposals pipeline which starts with an online agglomerative clustering algorithm (Section 3.2). Just using pre-computed deep features for object recognition for every video frame, our hypothesis is that we can localize actions finding clusters in this feature space. For doing so, our clustering solution must work online, *i.e.* grouping contiguous frames if they are visually similar. **2)** We then propose two filtering mechanisms to be performed over the clusters (Section 3.3). One is simply based on the size of the clusters/proposals. The other uses the dynamics of the proposals identified. Dynamics should represent the video-wide temporal evolution of the appearance of the frames. In this paper we introduce an unsupervised approach. It leverages rank pooling based dynamics [11] to build an actioness module which can be used to further filter and refine the obtained proposals/clusters. **3)** Our last contribution consists in offering a thorough experimental evaluation using the THUMOS'14 [10] dataset, with a clear evaluation protocol. This is done in Section 4, where we first compare the performance of our unsupervised model with state-of-the-art supervised solutions. Interestingly, when in the evaluation we control the number of proposals methods can produce, our solution reports the best results. We then analyze the precision of the action proposals, this being an aspect that has not been considered in depth before, even by supervised action proposal models, which generally tend to maximize the recall.

2 Related work

Temporal action proposal generation has recently become of much interest since it has been demonstrated to be a crucial step for temporal action detection [12], as well as helpful at other video understanding tasks [13].

Different types of solutions have been proposed to solve this problem. On the one hand, there are works based on classifying thousands of varied-length candidate segments, being these segments extracted using the sliding window technique. Then, several classification methods have been suggested to consolidate proposals, for example the multi-stage C3D [14] network used by [15], or the dictionary-based method proposed in [1]. Additionally, the works [4,16] propose to refine segment boundaries using temporal regression to generate more precise proposals.

On the other hand, Zhao *et al.* [6] propose to build candidate segments grouping features based on their actionness score. In [3,17] we find approaches that can generate proposals in a single video pass using recurrent networks. Besides, very recent models, *e.g.* [18,19], produce proposals from temporal boundary points, instead of candidate segments. These points are combined to generate precise temporal boundaries.

All previous methods share the fact that they solve the proposal generation task from a supervised perspective. That is, they need the temporal ground truth information during training. However, our method is the first one that operates in an unsupervised fashion. We only rely on video features to generate proposals, hence addressing and proposing a more challenging task.

3 Action Proposals Generation

We here detail our novel solution for the generation of action proposals in videos. Figure 1 shows the main steps of the introduced approach: 1) video frames feature extraction; 2) online hierarchical clustering; and 2) a rank pooling dynamics based filtering.

3.1 Feature Extraction

The input of our action proposal model is a video stream. Therefore, given a video of n frames $V = \{v_1, v_2, v_3, ..., v_n\}$, we proceed to extract, for each frame $v_i \in V$, a deep feature, using any pre-trained deep model for image recognition, such as AlexNet [20]. So, a video V is mapped to a set of high-dimensional deep features, having $V = \{f_1, f_2, f_3, ..., f_n\}$.

3.2 Online clustering for Action Proposals

Our solution to generate action proposals is based on a clustering algorithm. The intuition behind this approach is that action video frames share a visual similarity that can be captured by deep learning features trained for object

Algorithm 1: Frame by frame (FBF) online clustering for AP

Input: video $V = \{f_1, f_2, f_3, ..., f_n\}$;
threshold δ;
struct {float vec; int id;} leafnode;
Output: *Cluster*
//Create a list of leafnodes with the features in V
$features \leftarrow load(V)$;
$L \leftarrow [leafnode(v \leftarrow array(f), id \leftarrow i) for\ \ i, f\ \ in\ \ enumerate(features)]$
//Initialization
$Cluster \leftarrow \{\}$; $n_i \leftarrow L[0]$; $Cluster.append([n_i])$; $newvec \leftarrow n_i.vec$;
for $j \leftarrow 1$ ***to*** $n-1$ **do**
    $n_j \leftarrow L[j]$;
    $d \leftarrow L2dist(newvec, n_j.vec)$
    if $(d < \delta)$ **then**
        $Merge(Cluster[-1], n_j)$ // Merge n_j with the last node of Cluster
        $newvec \leftarrow Cluster[-1].vec$
    else
        //Create new cluster
        $Cluster.append([n_j])$; // n_j is appended as a new cluster
        $newvec \leftarrow n_j.vec$;
return *Cluster*

recognition. Therefore, we can localize actions finding clusters in the feature space where the video frames have been mapped to.

For doing so, it is fundamental that the solution guarantees two properties: 1) the implemented clustering must be *online*, in the sense that it tends to favour clusters with temporally close or contiguous frames; and 2) a filtering mechanism for discarding non-action clusters has to be designed. In this section, we focus on the online clustering implemented solution.

The input for our clustering solution is the set of deep learning features used to characterize every frame of a given video, *i.e.* $V = \{f_1, f_2, f_3, ..., f_n\}$. We then proceed to execute our online frame by frame (FBF) clustering algorithm in the following fashion.

First, we create a list L where each frame is assigned to a node n_i of the class $leafnode$. This class is implemented with an structure containing and index list id, which identifies all the features belonging to it, and a vector vec, which is the centroid of the cluster represented by the node. The algorithm decides whether to merge two *consecutive* pairs of nodes n_i and n_{i+1} using a distance based criterion computed over the centroids of the nodes, *i.e.* $n_i.vec$ and $n_{i+1}.vec$. Technically, we join two consecutive nodes if $dist(n_i.vec, n_{i+1}.vec) \leq \delta$. In our experiments the Euclidean distance is the one reporting the best results.

Note that our objective is to identify in the video action proposals regions. Following our online clustering solution, we consider that the frames merged in a cluster define an action proposal, the centroid being thus its representative.

If two consecutive nodes do not meet the union criterion, the cluster already formed in the first node is assigned to an action proposal. The algorithm starts again with the last analyzed node. We keep merging nodes until we go through all the frames of the video clip. Finally, our algorithm returns a list of clusters, which define the action proposals for the given video. For a detailed description of the FBF approach, we include the Algorithm 1.

Note that our online FBF model shares some similarities with an Unweighted Pair-Group Method using Centroid averages (UPGMC) for hierarchical clustering. Technically, we also follow a centroid linkage criterion. However, we do not need to perform any hierarchical search. Instead, we process the video frames in an online fashion, building clusters as soon as they occur. In our experiments, we tried other clustering approaches, like standard hierarchical clustering solutions [21] or HDBSCAN [22], but our FBF model reported the best results.

3.3 Filtering proposals

As a result of our online FBF model, every video frame gets assigned to an Action Proposal (AP). In other words, our model fully covers the whole video with proposals. This is due to the unsupervised nature of our approach, in contrast to all the state-of-the-art models for the same problem, which are all supervised approaches. Therefore, we need to incorporate a filtering step with the objective of discarding those incorrect proposals, but again in an unsupervised way.

We first proceed to filter the proposals by their size. The idea is simple, it technically consists in filtering those proposals whose temporal length is shorter than a certain threshold α_t.

Once we have discarded the shorter APs, which typically correspond to video fragments that do not contain actions, we proceed with a novel filtering mechanism, which is based on the computation of the dynamics of the remaining proposals.

The dynamics of a video sequence are defined as the video-wide temporal evolution of the appearance of the frames. Dynamics have been previously used for action recognition, *e.g.* [11,23]. They can be seen as video representations to train a classifier for categorizing each video with an action label. We, instead, propose to use them to measure the *actioness* of a video segment, *i.e.* how likely the video is to contain an action.

For doing so, we start our reasoning with the following hypothesis. Given a video segment V_i, we can compute its corresponding dynamics vector $\mathcal{D}_i$. We now define $\widehat{V}_i$ as a randomly ordered set of frames of V_i. $\widehat{\mathcal{D}}_i$ vector represents the dynamics of this random version of V_i. If we now compute the distance between $\mathcal{D}_i$ and $\widehat{\mathcal{D}}_i$, as $d(\mathcal{D}_i, \widehat{\mathcal{D}}_i)$, we make the basic assumption that this distance will be higher for those video segments that contain actions than for those that just represent background.

Following this hypothesis, we can filter our action proposals in an unsupervised way. Technically, we proceed as follows. Let P_i be one of the proposals constructed with the FBF approach. For each proposal P_i, we build its randomly ordered version $\widehat{P}_i$. Then, as in the rank-pooling model [11], we proceed to model the video dynamics of each of these proposals solving a constrained optimization pairwise-learning-to-rank formulation [24]. In particular, we opt for a linear Support Vector Regression (SVR) based formulation. Given the set of ordered features for P_i, *i.e.* $P_i = \{f_1^{(i)}, f_2^{(i)}, f_3^{(i)}, \ldots, f_{n_i}^{(i)}\}$, we seek a direct mapping from the input feature vectors $f_t^{(i)}$ to a *time* variable t using a linear model with parameters $w^{(i)}$, as follows,

$$w^{(i)} = \arg\min_{w^{(i)}} \sum_t |t - w^{(i)} \cdot f_t^{(i)}| \,. \tag{1}$$

This SVR approach is known to be a robust point-wise ranking formulation [24]. In summary, to encode the dynamics of proposal P_i, we use the model parameters vector $w^{(i)}$. For $\widehat{P}_i$ we also compute its corresponding dynamics $\widehat{w}^{(i)}$, following the same SVR based procedure.

Once the dynamics are computed, we proceed to filter the proposals according to the Euclidean distance between these dynamics vectors. If $d_i(w^{(i)}, \widehat{w}^{(i)}) < \Delta$, then the associated proposal is discarded, because it is considered to belong to a non-action video segment.

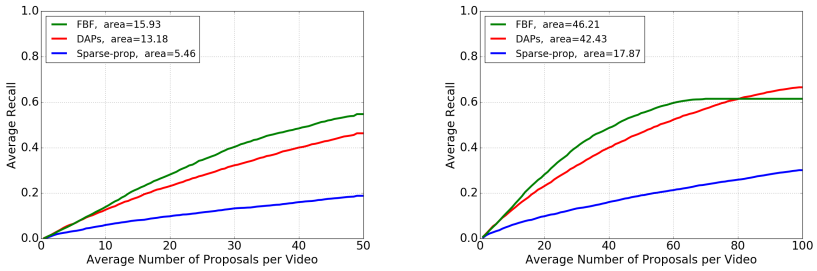
4 Experiments

4.1 Experimental Setup

For the experiments we use the challenging dataset THUMOS-14 [10]. It contains 213 *untrimmed* test videos with temporal annotations of 20 sport action categories. Note that this dataset is also used by all state-of-the-art supervised AP methods, which allows for a direct comparison of our unsupervised model with them.

In particular, we directly compare with the following supervised AP models: DAPs [3] and Sparse-prop [1]. The authors of these papers publicly release their proposals results using THUMOS-14.

With respect to the evaluation metrics, we use two. The first one is the Average-Recall versus the Average-Number of Proposals per Video (AR-AN). This is the standard metric used by all AP models. Technically, we follow the evaluation procedure detailed in [3], where the Average Recall curve is generated for a set of Intersection over Union between 0.25 to 0.5, as a function of the number of proposals. The second metric we propose for the experimental validation is the Average Precision of the Precision-Recall curves. We follow the official implementation of Average Precision released with the THUMOS-14 benchmark [10] for action detection. Note that while with AR-AN the temporal precision of the proposals is not considered, *i.e.* just the recall is evaluated, we aim to focus the attention on the fact that precision also matters. A method



(a) 50 (b) 100

Fig. 2: Comparison of FBF with the state of the art using the AR-AN metric. Our FBF is able to improve in terms of AR when 50 and 100 average number of proposals per video are considered.

that throws thousands of action proposals for each video can get an excellent recall, but very low precision, because many of those action proposals will fall into background zones, or will overlap with each other. However, our unsupervised online solution tends to generate fewer and more precise proposals, an aspect that will be shown by the Average Precision experimental evaluation.

With the aim of making our results reproducible, we detail the parameters of our solution. For the feature extraction, we proceed to extract for each video frame the last fully connected activation layer of the AlexNet CNN architecture [20], named $FC7$ (4096-dimensional), which has been pre-trained using the ImageNet database [25]. For the FBF clustering, $\delta = 100$, and we use an Euclidean distance. Then, for the filtering steps, $\alpha_t = 10$ and $\Delta = 10000$. Note that we tried to avoid any kind of manual parameter tuning as this could be considered a violation of the unsupervised character of our solution. Instead, we selected reasonable parameters in advance and held them fixed for all of the experiments.

4.2 Comparison with the state of the art

We start with a direct comparison of our unsupervised model, with the *supervised* state-of-the-art models DAPs [3] and Sparse-prop [1]. Figure 2 shows the AR-AN curves for 50 and 100 average number of proposals per video.

Our model reports a higher AR when both 50 or 100 average number of proposals per video are considered. Note that the average number of annotations per video in the THUMOS-14 dataset is of just 15! So, this means we are giving enough margin to the models. DAPs model starts to approach to our performance (AR = 42.43) when we allow it to cast 100 proposals per video. Interestingly, Figure 2b shows that our solution saturates at ~ 70 proposals per video maintaining an AR > 46. This saturation is mainly due to the clustering parameter δ used. Increasing this threshold will produce more (imprecise) proposals.

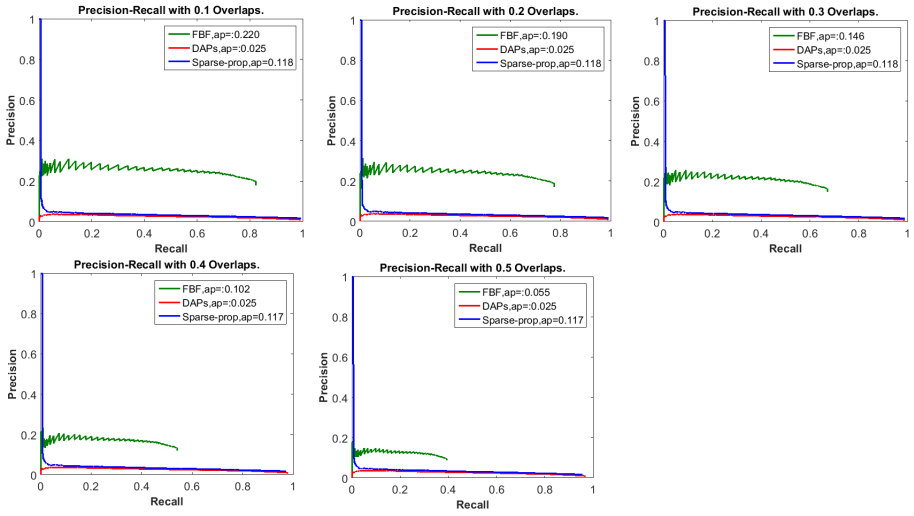


Fig. 3: Comparaison of the FBF with the state of the art using Precision-Recall curves with different Intersection over Union thresholds. Our approach is able to report the best Average Precision for most of them, being trained without any supervision.

The next question that arises is: how precise are the action proposals? In other words, are the AP methods casting action proposals in temporal localizations of the videos where there are actually actions occurring? For performing this analysis, we show in Figure 3 the Precision-Recall curves for our FBF approach and for the rest of the state-of-the-art methods. As one might expect, state-of-the-art models have been designed to maximize the recall, being their precision low. Note that we have incrementally augmented from 0.1 to 0.5 (the standard value for the object detection problem) the Intersection Over Union overlap criterion used in the Precision-Recall formulation to consider a true positive. This means that for 0.1, for instance, an AP is considered a true positive if the area of overlap between the predicted proposal and the ground truth annotation is at least of 10%. The higher this criterion, the more precise in terms of temporal location the proposals should be to be considered as correct.

DAP offers a fixed AP of 2.5% for all the overlap criteria considered, which means that the method is not able to report proposals with an overlap with the ground-truth higher than 1%. We also observe that our approach is able to report the highest Average Precision for most of the overlap criteria, all this in an unsupervised fashion, while the rest of methods are supervised.

Overall, according to the experimental evaluation designed, using the two metrics, the results show that our approach outperforms the supervised models when the number of proposal is controlled (under 100). Finally, in Figure 4

we show some qualitative results of the action proposals obtained by our FBF method.

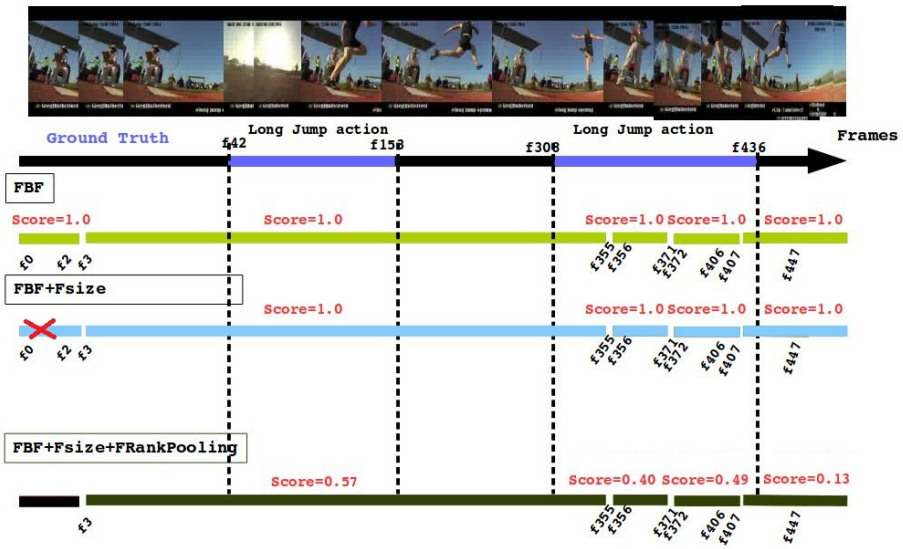


Fig. 4: Qualitative results. Set of action proposals discovered by our FBF model. The illustrative example concerns the video number 62 of the THUMOS'14 dataset which contains two actions of "Long jump" category. The first one starts from frame number 42 up to frame number 153. The second starts from frame number 308 up to frame number 436. Note that our solution FBF + Fsize + FRankPooling correctly covers the ground truth.

4.3 Ablation study

What is the performance of our clustering based solution when no filtering mechanism is used? Do implemented filtering mechanisms really help to increase approach recall? We conclude the experiments section with an ablation study where we address these question.

Figure 5 shows the effects of the incorporation of the different filtering mechanisms in terms of AR. A first observation is that the FBF alone is able to report a decent AR. This gives us confidence that such a clustering based approach is an appropriate solution for the action proposals problem. In other words, one can use deep learning features trained for object recognition to identify groups of frames that belong to an action. Incorporating a filtering by size technique is also beneficial, note how the AR increases for FBF+Fsize. However, the greatest

improvement in terms of recall is achieved thanks to the filtering based on the use of the described dynamics (+FRankPooling in Figure 5).

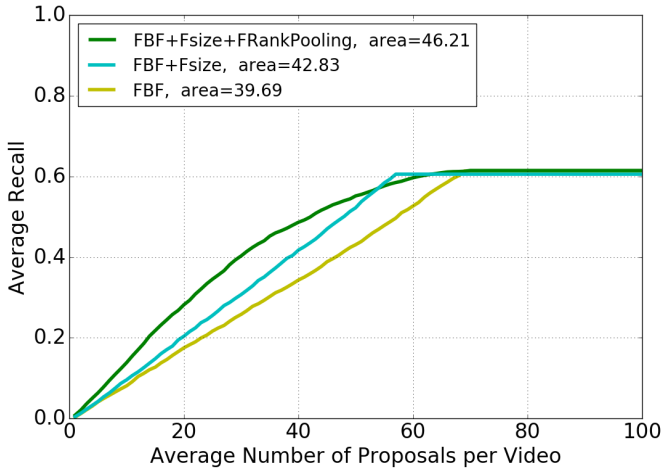


Fig. 5: Ablation study. This figure shows how the AR increases when we incorporate to our clustering solution (FBF) the filtering step using the size of the proposals (FBF+Fsize) and also the rank-pooling dynamics (FBF+Fsize+FRankPooling).

5 Conclusion

To generate action proposals is a difficult task which has not been studied from an unsupervised perspective. In this work we have presented the first attempt, to the best of our knowledge, to cast action proposals in an unsupervised fashion.

For doing so, we have introduced an approach which jointly integrates an online agglomerative clustering algorithm with a filtering mechanism that uses the dynamics of the clusters as an actioness measurement.

Our experimental evaluation shows that our model is able to outperform the average recall of supervised state-of-the-art approaches when the number of proposals is limited for all the methods. We have also shown that although our solution is unsupervised, the precision of the action proposals we generate is better than for the fully supervised models. Finally, an ablation study confirms our hypothesis and the adequateness of the designed approach.

Acknowledgments

This work is supported by project PREPEATE (TEC2016-80326-R), of the Spanish Ministry of Economy, Industry and Competitiveness. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPU used for this research. Cloud computing resources were kindly provided through a Microsoft Azure for Research Award.

References

1. Caba Heilbron, F., Carlos Niebles, J., Ghanem, B.: Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In: CVPR. (2016) 1914–1923
2. Chao, Y.W., Vijayanarasimhan, S., Seybold, B., Ross, D.A., Deng, J., Sukthankar, R.: Rethinking the faster R-CNN architecture for temporal action localization. In: CVPR. (2018)
3. Escorcia, V., Heilbron, F.C., Niebles, J.C., Ghanem, B.: Daps: Deep action proposals for action understanding. In: ECCV, Springer (2016) 768–784
4. Gao, J., Yang, Z., Chen, K., Sun, C., Nevatia, R.: TURN TAP: Temporal unit regression network for temporal action proposals. In: ICCV. (Oct 2017)
5. Jain, M., Van Gemert, J., Jégou, H., Bouthemy, P., Snoek, C.G.: Action localization with tubelets from motion. In: CVPR. (2014) 740–747
6. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D.: Temporal action detection with structured segment networks. In: CVPR. (2016)
7. Ghanem, B., Niebles, J.C., Snoek, C., Caba-Heilbron, F., Alwassel, H., Escorcia, V., Khrisna, R., Buch, S., Duc-Dao, C.: The ActivityNet large-scale activity recognition challenge 2018 summary. arXiv:1808.03766 (2018)
8. Lin, T., Zhao, X., Shou, Z.: Temporal convolution based action proposal: Submission to activitynet 2017. arXiv preprint arXiv:1707.06750 (2017)
9. Xu, H., Das, A., Saenko, K.: R-C3D: Region convolutional 3D network for temporal activity detection. In: ICCV. (2017)
10. Jiang, Y., Liu, J., Zamir, A.R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: Thumos challenge: Action recognition with a large number of classes (2014)
11. Fernando, B., Gavves, E., Oramas, J., Ghodrati, A., Tuytelaars, T.: Rank pooling for action recognition. IEEE TPAMI **39**(4) (2017) 773–787
12. Alwassel, H., Caba Heilbron, F., Escorcia, V., Ghanem, B.: Diagnosing error in temporal action detectors. In: ECCV. (September 2018)
13. Gao, J., Ge, R., Chen, K., Nevatia, R.: Motion-appearance co-memory networks for video question answering. In: CVPR. (2018) 6576–6585
14. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: ICCV. (Dec 2015) 4489–4497
15. Shou, Z., Wang, D., Chang, S.F.: Temporal action localization in untrimmed videos via multi-stage CNNs. In: CVPR. (2016)
16. Gao, J., Yang, Z., Nevatia, R.: Cascaded boundary regression for temporal action detection. In: BMVC. (2017)
17. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Niebles, J.C.: Sst: Single-stream temporal action proposals. In: CVPR. (2017)
18. Gao, J., Chen, K., Nevatia, R.: CTAP: Complementary temporal action proposal generation. In: ECCV. (2018)

19. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: BSN: Boundary sensitive network for temporal action proposal generation. In: ECCV. (2018)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. (2012) 1097–1105
21. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall (1988)
22. McInnes, L., Healy, J., Astels, S.: HDBSCAN: Hierarchical density based clustering. The Journal of Open Source Software **2**(11) (2017)
23. Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., Gould, S.: Dynamic image networks for action recognition. In: CVPR. (2016)
24. Liu, T.Y.: Learning to rank for information retrieval. Found. Trends Inf. Retr. **3**(3) (March 2009) 225–331
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3) (2015) 211–252