Guess Where? Actor-Supervision for Spatiotemporal Video Action Localization

Escorcia, V.; Dao, C.D.; Jain, M.; Ghanem, B.; Snoek, C.

DOI
10.1016/j.cviu.2019.102886

Publication date
2020

Document Version
Final published version

Published in
Computer Vision and Image Understanding

License
Article 25fa Dutch Copyright Act (https://www.openaccess.nl/en/in-the-netherlands/you-share-we-take-care)

Link to publication

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 426, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
1. Introduction

The goal of this paper is to localize and classify actions like skateboarding or walking with dog in video by means of its enclosing spatiotemporal tube, as depicted in Fig. 1. Empowered by action proposals (Jain et al., 2014; Weinzaepfel et al., 2015; Zhu et al., 2017), deep learning (Gkioxari and Malik, 2015; Saha et al., 2016) and carefully labeled datasets containing spatiotemporal annotations (Soomro et al., 2012; Rodriguez et al., 2008; Xu et al., 2015), progress on this challenging topic has been considerable (Kalogeiton et al., 2017; Hou et al., 2017). However, the dependence on deep learning and spatiotemporal boxes is also hampering further progress, as annotating tubes inside video is tedious, costly and error prone (Mettes et al., 2016). We strive for action localization without the need for spatiotemporal video supervision.

Others have also considered action localization without spatiotemporal supervision (Siva and Xiang, 2011; Mettes et al., 2017; Li et al., 2018). Recently, Li et al. (2018) proposed a deep learning based model for action classification with an attention LSTM. The attention component highlights regions in the video that correspond to high-responses of certain action class labels. Unfortunately, this scheme does not ensure high-localization accuracy as the model may learn to attend only to discriminative parts of the action, such as the legs and the skateboard for the action skateboarding, but not the entire actor. Siva and Xiang (2011) and Mettes et al. (2017) circumvent this issue and aim to retrieve the entire actor by relying on human detectors, trained on images. These approaches learn a classifier using a multiple instance learning framework. This framework selects the best candidate proposal in the video guided by multiple cues, in particular the detected human actors, which is then used to learn an action classifier. These works are shallow and were not designed to exploit the representation learning principle of deep learning architectures. Our work unifies these alternatives. It infuses the pragmatic and arguably more accurate scheme of localization from detected actors into a novel end-to-end trainable deep architecture.

In this work, we introduce an actor-supervised architecture that exploits the relevance of actors to steer the localization of actions in videos without using spatiotemporal annotations of the training videos. Instead of using the detected actors to select among candidate regions a posteriori (Siva and Xiang, 2011; Mettes et al., 2017), we exploit the detections to define the candidate proposals a priori. Based on them, our architecture learns to rank the potential actor tubes from action labels at the video level. Our technical contributions are twofold. First, we introduce actor proposals; a means to generate candidate regions that are likely to contain an action and that do not require any action video annotations for training. We derive our proposals from a detector for human and non-human actors, intended for images, combined with Siamese similarity matching to account for actor deformations over regions of interest.
Weakly-supervised action proposals do not rely on box-level ground-truth for all the video frames (Kläser et al., 2012; Lan et al., 2011; Tran and Yuan, 2012; Yu and Yuan, 2015). Instead, they exploit object cues such as color or image gradients (Jain et al., 2014, 2017; Oneata et al., 2014). Clustering of motion trajectories is also an effective choice but has an exponentially large search space. More efficient methods have been proposed, such as dynamic programming based on the Viterbi algorithm (Gkioxari and Malik, 2015).

Supervised classification is the default in the action localization literature. Methods train classifiers using box-supervision for each action class and apply it on each action proposal for each test video, e.g. Gkioxari and Malik (2015), Weinzaepfel et al. (2015), Saha et al. (2016), Hou et al. (2017b), Kalogeiton et al. (2017), Saha et al. (2017a,b) and Duarte et al. (2018), Xie et al. (2018). Others, who rely on unsupervised or weakly-supervised action proposals, also train their action proposal classifiers in a supervised fashion using bounding-box ground-truth across frames (Jain et al., 2017; van Gemert et al., 2015; Chen and Corso, 2015).

Unsupervised classification has been addressed as well. Puscas et al. (2015) classify their unsupervised proposals using tube-specific and class agnostic detectors, trained via two-stage transductive learning. Soomro and Shah (2017) start with supervoxel segmentation and automatically discover action classes by discriminative clustering. It localizes actions by knapsack optimization. Jain et al. (2015), classify action proposals in a zero-shot fashion by encoding them into a semantic word embedding spanned by object classifiers. Mettes and Snoek (2017), capture actors, relevant object detections and their spatial relations in a word embedding. All the training happens on images and text, no videos are needed.

Weakly-supervised classification refrains from box-supervision for classifying action proposals. A considerable reduction in annotation effort may be achieved by replacing boxes with point annotations and unsupervised action proposals (Mettes et al., 2016), but it still demands manual labor. An alternative is to rely on human body parts (Ma et al., 2013) or human detectors trained on image benchmarks (Russakovsky et al., 2015; Lin et al., 2014) to steer the localization in video; either by defining the search space of most likely action locations, e.g. Siva and Xiang (2011), or by selecting the most promising action proposal (Mettes et al., 2017).

We also rely on human (and non-human) actor detectors but exploit them to generate a limited set of actor proposals. Among those, we select the best ones per action, based on an actor attention mechanism that only requires action class labels. Without the need for box annotations per video frame, we achieve results not far behind the supervised methods and much better than unsupervised methods.

3. Actor-supervision architecture

To deal with the inherent difficulty of spatiotemporal action localization without box supervision, we introduce actor supervision. We exploit the fact that actors are precursors of actions. Actions result from an actor going through certain transformations, while possibly interacting with other actors and/or objects in the process. This means that actors not only locate the action in the video, but also one can learn to rank the potential actor locations for a given action class.
Based on these premises, we design a novel end-to-end architecture for spatiotemporal action localization guided by actor supervision. Fig. 2 illustrates the two pillars of our architecture, namely actor proposals and actor attention. In a nutshell, our approach enables the localization of actions with minimal supervision by (i) infusing the concept of actors in the architecture, through existing knowledge and progress in object detection and object tracking; and (ii) introducing a powerful attention mechanism suitable for learning a meaningful representation of actions. In the following subsections we disclose full details of each pillar.

3.1. Actor proposals

Our actor proposals receive a video stream and generate a set of tubes, parameterized as a sequence of boxes $T = \{B_t\}$. The tubes outline the most likely spatiotemporal regions where an action may occur based on the presence of an actor. It contains two modules: actor detection and actor linking, as shown in Fig. 3 and detailed next.

**Actor detection.** This module generates spatial locations where the actor of interest appears in the video. Respecting the requirements of our setup, this module adopts a pre-trained conv-net for object detection and actor linking, as shown in Fig. 3 and detailed next.

**Actor linking.** This stage carefully propagates the predictions of our actor detector throughout the video to generate an actor proposal tube. It complements the detector by filling the gaps left during the performance of the action, without demanding any annotation to tune the detector. In this way, our module is more robust to missed detections, characteristic of the performance of the action, may involve significant visual changes that do not fit the canonical model of the object category of the actor.

**Actor proposal classification.** We classify each actor proposal according to a pre-defined set of action classes. This module learns to map the fixed size representation of each actor proposal into the space of actions of interest. In practice, we employ a fully connected layer where the number of outputs corresponds to the number of classes $A$. During training, the main challenge is to learn an appropriate mapping of the action representation into the action space without the obligation of explicit annotations for each actor proposal. For this reason, we propose an attention mechanism over the actor proposals that bootstraps the action labels at the video level. In this way, we encourage the network.
to learn the action classifier by focusing on the actors that contribute to an appropriate classification.

In this work, we explore the use of an attention mechanism based on top-$k$ selection. It encourages the selection of the $k$ most relevant actor proposals per class that contribute to perform a correct classification. In practice, we choose the top-$k$ highest scores from the fully connected layer for each action category, and average them to form a single logit vector for each video. Subsequently, we apply a softmax activation on the logits of each video. Fig. 5 illustrates the particular instantiation described before. Note that our architecture classifies each actor independently, it could easily be extended for multi-class videos. In such case, we would change the softmax nonlinearity by a sigmoid during training.

Learning. We train our actor attention using the cross-entropy loss between the output of the softmax and the video label. It is relevant to highlight that we do not use any spatiotemporal information about the actions for learning the parameters of our model. In practice, we fit the parameters of the actor attention by employing back-propagation and stochastic gradient descent. In the case of the top-$k$ selection module, we use a binary mask during the back-propagation representing the subgradients of the selection process.

Having defined our actor-supervised architecture, we are now ready to report experiments.

4. Experiments

4.1. Datasets and evaluation

We validate our approach on four public benchmarks for spatiotemporal action localization in videos:

**UCF-Sports** (Lan et al., 2011). This dataset consists of 150 videos from TV sport channels representing 10 action categories such as *weightlifting, diving, golf-swing*.

Datasets used in this paper were downloaded and experimented on by primary author.
of VGG-16, followed by a cross-correlation layer (as in Bertinetto et al., 2016) trained on the ALOV dataset (Smeulders et al., 2014). The base network of the actor detector is InceptionV2 (Ioffe and Szegedy, 2015) – followed by six feature layers and box predictors (Liu et al., 2016; Huang et al., 2017). We only track the detections selected by our actor linking algorithm to accommodate for deformation of the actor, that the Viterbi linking is unable to fix. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

JHMDB (Jhuang et al., 2013). This dataset showcases 21 action categories such as push-up, shooting, etc.; and consists of 928 videos from movies and consumer videos from internet portals. Unless stated otherwise, we employ the standard evaluation protocol using three splits. We refrain from using the box annotations in the training set to tune our model.

THUMOS13 (Soomro et al., 2012). This dataset incorporates untrimmed videos and multiple action instances per video. It consists of a subset of 3294 videos derived from UCF101 featuring 24 action categories. We use the training and testing partition from split 1 of UCF101 for evaluating our approach. Note that we do not rely on the spatiotemporal box annotations of the training set.

Evaluation. Following the standard protocol for action localization, we report the intersection over union (IoU) to measure the degree to which a candidate tube is associated with a given spatiotemporal action ground-truth annotation. Depending on the task and dataset of interest, we report the result in terms of Recall or mean Average Precision (mAP). To evaluate action classification performance, we employ the evaluation setup typical for action localization using mean average precision (mAP) over all available classes, given an overlap threshold of 0.2 (THUMOS13) and 0.5 (UCF-Sports, JHMDB).

4.2. Implementation details

Actor proposals. We use a single-shot multi-box detector (Liu et al., 2016) to detect an actor of interest in every frame. We used a pretrained detector trained on all MS-COCO categories (Huang et al., 2017; Lin et al., 2014) and limit the detections used to the actors of interest according to the action categories defined in each dataset. The base network of the actor detector is InceptionV2 (Ioffe and Szegedy, 2015) – up to inception5b pre-trained on ILSVCR-12 (Russakovsky et al., 2015) – followed by six feature layers and box predictors (Lia et al., 2016; Huang et al., 2017). We only track the detections selected by our actor linking algorithm to accommodate for deformation of the actor, that the Viterbi linking approach is unable to fix. Previous approaches e.g. Gkioxari and Malik (2015), Kläser et al. (2012) and Saha et al. (2016) employ supervision at the level of boxes and length of the tubes to overcome this issue. This clearly limits their application under the weakly-supervised setup evaluated in this work. We conclude that actor linking, by similarity-based matching, aids spatiotemporal action localization with weak supervision.

Actor proposals versus others. Table 1 compares our actor proposals with previous supervised and unsupervised action proposals. Compared to the action proposals by Yu and Yuan (2015), our approach achieves an improvement of +34.2% in terms of Recall on THUMOS13. This result evidences the benefit of our actor detection and linking scheme. Interestingly, our approach improves upon previous unsupervised work by +1.7% and +8.4% in terms of Recall on UCF-Sports and THUMOS13, respectively. These methods (Jain et al., 2017; van Gemert et al., 2015) are based on grouping techniques over low-level primitives such as color and motion, which reaffirms our intuition about the relevance of actors as a strong semantic cue for the localization of the actions. Fig. 7 illustrates the recall of our actor proposals for a varying number of proposal in comparison with previous unsupervised approaches. It provides further evidence on the quality of our proposals, especially when considering only a limited number of proposals.

The state of the art for action proposals generation (Zhu et al., 2017; Weinaefelp et al., 2015) are fully supervised approaches based on a mix of convolutional-recurrent networks and supervised instance level tracking, respectively. Our approach achieves competitive results or even outperforms them in JHMDB-split1, following the evaluation protocol of Weinaefelp et al. (2015), without relying on additional action box supervision. Although supervised approaches offer proposals with better quality in two out of the three datasets studied, they do so at the expense of extra annotations. It that sense, these methods have a limited scalability potential, and it opens up a spot for our actor proposals.

Non-human actor proposals. We also analyze the quality of our proposals for generating spatiotemporal tubes for non-human actors. For this experiment we assume that the class of the actors are known at test time and evaluate the quality of the localization accordingly. Fig. 8
summarizes our findings including two qualitative results taken from the A2D dataset. The leftmost example shows we are able to generate proposals for highly articulated actors like birds. The example in the center exemplifies a common failure case for the ball actor. In this case, the ball changes significantly in appearance during the execution of the action, including several full occlusions caused by the human. From our quantitative analysis, we appreciate that the actor with the highest recall is baby, which is not directly represented in the training set of our actor detector. The analysis also reveals that the gap in recall at 0.5 IoU between adult actors and animal actors like bird and dog is at most +2%. Except for ball, the recall for all the actors using at most 50 actor proposals is greater than 85%. Therefore, we conclude that our method is general and applicable to both human and non-human actors.

Impact of actor attention. We validate the benefits of our actor attention stream, and ablate the operations of our novel actor pooling layer.

How to pool the spatial information of the actors? Table 2(a) compares the proposed bilinear interpolation vs. the RoI pooling strategy proposed by Hou et al. (2017b), by replacing the former with the latter in our actor pooling module. The result shows an improvement of +7.9% in mAP when bilinear interpolation is used. This could be a consequence of the smoother representation with less sparse gradients mentioned in Section 3.2, while training in a regime without box level supervision.

How to aggregate the temporal information of the actor? Table 2(b) compares different types of temporal aggregation function of the actor representation after bilinear interpolation. We observed that Early pooling, averaging the temporal information of V after the bilinear sampling, performs the best. Followed closely by Late pooling, computing the likelihood of each box in the actor tube independently and average. In contrast, No pooling, using fixed size representation of a group of frames V, and 3D-conv, a 3D conv kernel followed by early pooling, do not perform as well in THUMOS13. This result could be a consequence of some degree of overfitting as coarse level information is enough to classify the actions of interest and modeling precise details of the actors is not necessary in the current benchmarks. We used early pooling for the rest of the experiments in the paper.

Benefit of end-to-end training. Finally, we showcase the impact of fine-tuning the video encoder in an end-to-end fashion, as training
proceeds, in Table 2(c) on THUMOS13. When the video encoder is fine-tuned we achieve an improvement of 4.4% in mAP, compared to when employing a fixed visual representation, as done in Li et al. (2018).

Impact of detection and tracking on action localization. Fig. 9 shows per-class correlation between action localization AP at IoU = 0.2 and actor linking (detection + tracking) frame-level performance, Recall IoU = 0.5. We noticed that among all action classes AP is not correlated to the frame-level recall. We observe two clusters of classes, one mostly containing the untrimmed classes (in red) and the other mostly containing trimmed classes (in yellow). For each cluster, the action localization performance is directly proportional to that of actor detection and tracking.

Actor-supervision versus others. Table 3 compares multiple approaches with a varying degree of supervision. Based on these results, we note that actor supervision achieves the state-of-the-art among all approaches that localize action from an action class label only. It improves upon (Li et al., 2018) by +8.1% and upon (Mettes et al., 2017) by +10.4% on the THUMOS13 and UCF-Sports benchmarks, respectively. Compared to Mettes et al. (2017), our approach gives more relevance to the actors during the localization stage instead of using them as cues to improve the ranking of existing action proposals. We hypothesize that our attention mechanism is more effective than the one in Li et al. (2018), because actors are a more powerful cue for guiding the localization of actions than individual pixels. Table 4 illustrates the results of our approach for a broad range of IOU thresholds in contrast with the state of the art approach for weakly-supervised action localization of Mettes et al. (2017). It is evident that our architecture achieves better localization accuracy, especially on higher IOU values. Interestingly, actor-supervision outperforms this multiple instance learning approach in the most challenging dataset, THUMOS13, over all the IOU values reported.

Actor-supervision also outperforms several approaches with varying levels of supervision on the challenging THUMOS13 benchmark (Yu and Yuan, 2015; van Gemert et al., 2015; Mettes et al., 2016). Compared to Weinzaepfel et al. (2015) and Hou et al. (2017b), our visual representation is limited to the RGB video stream. We suspect that flow information can boost the action classification and localization results as earlier shown by Carreira and Zisserman (2017), Gkioxari and Malik (2015) and Saha et al. (2016). Similarly, we compare actor-supervision on JHMDB with Gkioxari and Malik (2015), using their RGB stream only, and observe comparable results (35.8% ± 2.7 versus 37.9%). This proves the relevance of our approach in another challenging benchmark against a strong competitor aided by box-supervision during training.

The state-of-the-art in action localization is dominated by fully-supervised approaches resembling conv-net architectures well established for generic object detection (Kalogeroulion et al., 2017; Saha et al., 2017b), which require strong levels of supervision. These approaches are unable to be trained in the weakly-supervised regime presented in this paper. Interestingly, our approach not only outperforms other weakly-supervised methods but it also has an edge over some of the supervised approaches. Considering the poor scalability of fully-supervised approaches and the tremendous amount of progress in object detection, we envision that our work can inspire the community to seek other forms of supervision during the design and adaption of deep representations for localizing actions in videos. Figs. 10–11 illustrate action localization results of our approach on the THUMOS13 benchmark. The latter showcases that our method can deal with videos exhibiting multiple actors such as Ice Dancing. We also corroborate that the mAP of actions exhibiting multiple actors is similar to those of actions with a single actor.

4.4. Discussion

Our tubes span the entire videos as most datasets for spatiotemporal localization are trimmed or nearly trimmed. Only THUMOS13 contains action tubes that does not expand the entire video. Yet THUMOS13 is nearly trimmed; as the instance temporal coverage, defined as the ratio of the length of an action tube by the length of the video, is 74% on average, and only five out of twenty-four classes have an average coverage lower than 50%. We observed that the AP@IoU = 0.2 in those five classes is 0.25x lower than the other classes. Interestingly, concurrent work on weakly-supervised temporal localization has emerged (Nguyen et al., 2018; Paul et al., 2018) without localizing the actors as presented here. Thus, our work is orthogonal and we envision follow-up work merging both approaches, accompanied by new datasets where the untrimmed aspect of spatiotemporal action instances is widespread.
Table 3
Comparison of spatiotemporal action localization approaches with decreasing amount of supervision. The top half shows supervised approaches, whereas the bottom half shows weakly-supervised approaches relying on action class labels only. Actor-supervision achieves state-of-the-art performance among the weakly-supervised approaches and sometimes even outperforms supervised methods.

<table>
<thead>
<tr>
<th></th>
<th>Action supervision</th>
<th>THUMOS13</th>
<th>UCF-Sports</th>
<th>JHMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Boxes</td>
<td>Points</td>
<td>Labels</td>
<td>mAP@0.2</td>
</tr>
<tr>
<td>Kalogeiton et al. (2017)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>77.2</td>
</tr>
<tr>
<td>Saha et al. (2017b)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>73.5</td>
</tr>
<tr>
<td>Hou et al. (2017b)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>47.1</td>
</tr>
<tr>
<td>Jain et al. (2017)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>48.1</td>
</tr>
<tr>
<td>Weinzaepfel et al. (2015)</td>
<td>✓</td>
<td>✓</td>
<td>46.8</td>
<td>90.5</td>
</tr>
<tr>
<td>Gkioxari and Malik (2015)</td>
<td>✓</td>
<td>✓</td>
<td>–</td>
<td>75.8</td>
</tr>
<tr>
<td>van Gemert et al. (2015)</td>
<td>✓</td>
<td>✓</td>
<td>37.8</td>
<td>–</td>
</tr>
<tr>
<td>Yu and Yuan (2015)</td>
<td>✓</td>
<td>✓</td>
<td>26.5</td>
<td>–</td>
</tr>
<tr>
<td>Mettes et al. (2016)</td>
<td>✓</td>
<td>✓</td>
<td>34.8</td>
<td>–</td>
</tr>
<tr>
<td>Mettes et al. (2017)</td>
<td>✓</td>
<td>✓</td>
<td>37.4</td>
<td>37.8</td>
</tr>
<tr>
<td>Li et al. (2018)</td>
<td>✓</td>
<td>✓</td>
<td>36.9</td>
<td>–</td>
</tr>
<tr>
<td>Cinbis et al. (2014) (from Mettes et al. (2016))</td>
<td>✓</td>
<td>✓</td>
<td>13.6</td>
<td>–</td>
</tr>
<tr>
<td>Sharma et al. (2015) (from Li et al. 2019)]</td>
<td>✓</td>
<td>✓</td>
<td>5.5*</td>
<td>–</td>
</tr>
<tr>
<td><strong>Our paper</strong></td>
<td>✓</td>
<td>✓</td>
<td><strong>45.8</strong></td>
<td><strong>48.2</strong></td>
</tr>
</tbody>
</table>

*Denotes use of RGB frames solely, as we do.

Table 4
Localization accuracy for multiple IoU. Actor-supervision outperforms the state-of-the-art weakly-supervised approach of Mettes et al. (2017) for a broad range of stringent IoU thresholds.

<table>
<thead>
<tr>
<th>IOU thresholds</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UCF-sports</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mettes et al. (2017)</td>
<td>87.7</td>
<td>81.7</td>
<td>64.4</td>
<td>54.5</td>
<td>37.8</td>
<td>17.5</td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td>84.2</td>
<td>84.2</td>
<td>84.2</td>
<td>64.9</td>
<td>48.2</td>
<td>40.6</td>
</tr>
<tr>
<td><strong>THUMOS13</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mettes et al. (2017)</td>
<td>49.8</td>
<td>37.4</td>
<td>25.8</td>
<td>13.7</td>
<td>6.2</td>
<td>1.3</td>
</tr>
<tr>
<td><strong>This paper</strong></td>
<td>53.4</td>
<td>45.8</td>
<td>38.0</td>
<td>30.7</td>
<td>19.3</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Fig. 10. Qualitative results on the THUMOS13 dataset. Top row shows three successful cases by visualizing the ground-truth and action tubes as well as two highlighted frames. These include action sequences that have deformations of actor as well as multiple actors with complex background. Bottom row visualizes three failed cases which show that crowded background, occlusions and temporally untrimmed action sequences are the most challenging scenarios. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
5. Conclusion

This paper introduces a weakly-supervised approach for the spatiotemporal localization of actions, driven by actor supervision. We show that exploiting the inherent compositionality of actions, in terms of transformations of actors, disregards the dependence on spatiotemporal annotations of the training videos. In the proposal generation step, we introduce actor supervision in the form of an actor detector and similarity-based matching to locate the action in the video as a set of actor proposals. Then, our proposed actor attention learns to classify and rank these actor proposals for a given action class. This step also does not require any per frame box-level annotations. Instead, we design an attention based mechanism that chooses the most relevant actor proposal only for class labels at the video level. Moreover, we introduce a novel actor pooling operation that summarizes the representation of each actor in a more effective way than recent pooling strategies for the weakly-supervised setup of our interest. Our approach outperforms the state-of-the-art among weakly-supervised works and even achieves results that are better or competitive to some of the fully-supervised methods. In future work, we envision clever redesigns of our actor supervised approach to attain further improvement in the spatiotemporal localization of actions without action box annotations.

Acknowledgments

This publication is based upon work supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research (OSR) under Award No. OSR-CRG2017-3405. We thank the team members of the IVUL from KAUST and Qualcomm AI Research for helpful comments and discussion. In particular, we appreciate the support of Amirhossein Habibian during the implementation of our Actor Linking.

Appendix

We complement the manuscript with the following items:

• A video summarizing our work. It showcases more qualitative results generated by our actor-supervised architecture. There you can appreciate that our approach works well in videos with multiple actors, involving considerable deformations and for non-human actors.
  The codec of the video is H264-MPEG-4 AVC, its resolution is 1280 × 720 and the frame rate 60. We recommend using the VLC media player https://www.videolan.org/vlc/index.html.
• In Appendix A.1, we describe the inner details of our actor proposal algorithm and comment about its computational complexity.
• Appendix A.2 gives more details about the training of our actor-supervised architecture concerning initialization and training policy, details about the backpropagation, and robustness of our actor pooling operation.

A.1. Actor proposals

The Algorithm 1 describes all the interactions between the inner blocks involved for the generation of our actor proposals, described in the main paper.

Algorithm 1 Actor proposals generation

1: Input: maximum number of proposals N
2: Output: $\mathcal{T}$
3: $D \leftarrow$ run actor detector over all frames
4: $T \leftarrow \emptyset$
5: $i \leftarrow 0$
6: while $D \neq \emptyset \land i < N$ do
7: $b_i \leftarrow$ select actor with highest score from $D$
8: $B_i \leftarrow$ actor tracker tracks $b_i$ forward and backward throughout the video
9: Push $B_i$ onto $T$
10: $D \leftarrow$ filter actors in $D$ with high similarity with boxes in $B_i$
11: $i \leftarrow i + 1$

Regarding the time complexity of our approach, our object detector and Siamese-tracker run at 32 and 60 FPS, respectively, on an Intel Xeon E5-2687 with a GTX 1080. After direct contact with Jain et al. (2017), we establish that our method generates less number of proposals (100), is more accurate (+10% Recall at 0.5 IOU) and 7.91 times faster on THUMOS13. In conclusion, our action proposal approach generates more precise candidates and does so in less time.

A.2. Actor-supervised architecture

Backpropagation details. Fig. 5 illustrates the end-to-end characteristic of the actor attention stream of our Actor-supervised architecture. The modules involved during the training of our actor attention. These are (i) the video encoder; (ii) our actor pooling; (iii) the actor attention classifier; and (iv) the attention mechanism. The latter is composed by a top-k selection operational average among top-k scores per-class; and a softmax activation per video. This diagram also highlights the capability to be trained in a weakly-supervised setup, only from class labels at the video level. We can note that gradients flow backwards down to the raw video updating the parameters of our video encoder. This contrasts with the strategy of Li et al. (2018) that trains a recurrent module on top of pre-computed features from the last convolutional block of VGG16. In that sense, our work is the first deep architecture for weakly-supervised spatiotemporal action localization in videos trained from raw visual information.

On the other hand, it is relevant to mention that we did not employ any supervisory signal nor perform any updates on the parameters of our actor proposal. Note that the use of any additional supervisory signal to tune our actor proposals module, top stream in Fig. 5, goes against the weakly-supervised setup of our interest.

Training details of our actor-attention stream. We initialize the weights of our actor classifier module with Xavier technique (Glorot and Bengio, 2010), and our video encoder with the weights from a VGG-16 model pre-trained on ImageNet-ILSVCR-2012 (Russakovsky et al., 2015).
et al., 2015). Our actor attention stream is trained for 20 epochs annealing the learning rate by a factor of 0.75 after eleven epochs. We use a momentum factor of 0.99 and an initial learning rate of 0.01. As input pre-processing, we employ the segment-based strategy suggested by Wang et al. (2016). In our case, we randomly sample 16 frames uniformly spaced per video. Additionally, we apply a random horizontal flipping of all the sampled frames. Finally, we normalize the input frames such that the intensity values lie on the range between [−0.5, 0.5] using standard scaling with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225] for the RGB channel.

Robustness of actor pooling. As we mentioned in the main paper, the use of our actor pooling yields better results than the tube pooling operation (Hou et al., 2017b) in the weakly-supervised setup of our interest. Fig. 12 compares the robustness of our actor pooling operation in regards to the tube pooling operation. The violin plot, Fig. 12, summarizes the statistics of the top-15 results among forty experiments for each trial i.e. our actor supervised architecture using either actor pooling or tube pooling. One experiment constitutes a variation of the grid size (3 × 3 or 5 × 5), or the optimization hyper-parameters, such as learning rate, momentum, etc., while maintaining the other hyper-parameters of our actor supervised architecture intact. In this manner, we can compare both operations beyond a single hyper-parameter configuration without introducing other confounding variables. For this experiment, we use the THUMOS13 dataset employing the training-testing partition from the split 2 of UCF101. More details of the experiment are provided in the text.

Our actor pooling statistically improves upon the tube pooling operation with a p-value of 1%. Thus, we reaffirm that our actor pooling consistently achieves better localization performance than tube pooling beyond a single hyper-parameter configuration.

References

Girshick, R., 2015. Fast r-CNN. In: ICCV.
Li, Z., Gavrilyuk, K., Garves, E., Jain, M., Snoek, C., 2018. VideoSTM convolves, attends and flows for action recognition. CVPR.
Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K., 2018. Rethinking spatiotemporal feature learning for video understanding. In: ECCV.