PUNET: TEMPORAL ACTION PROPOSAL GENERATION WITH POSITIVE UNLABELED LEARNING USING KEY FRAME ANNOTATIONS

Noor ul Sehr Zia　*Osman Semih Kayhan*　*Jan van Gemert*

Computer Vision Lab, Delft University of Technology, The Netherlands

ABSTRACT

Popular approaches to classifying action segments in long, realistic, untrimmed videos start with high quality action proposals. Current action proposal methods based on deep learning are trained on labeled video segments. Obtaining annotated segments for untrimmed videos is time consuming, expensive and error-prone as annotated temporal action boundaries are imprecise, subjective and inconsistent. By embracing this uncertainty we explore to significantly speed up temporal annotations by using just a single key frame label for each action instance instead of the inherently imprecise start and end frames. To tackle the class imbalance by using only a single frame, we evaluate an extremely simple Positive-Unlabeled algorithm (PU-learning). We demonstrate on THUMOS'14 and ActivityNet that using a single key frame label give good results while being significantly faster to annotate. In addition, we show that our simple method, PUNet[1], is data-efficient which further reduces the need for expensive annotations.

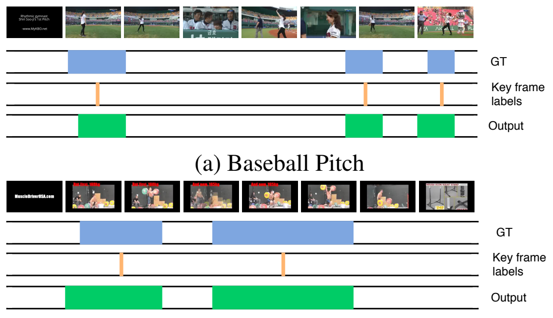
***Index Terms*—** Proposal Generation, Action Localization, Positive-Unlabeled Learning

1. INTRODUCTION

With videos naturally untrimmed and multiple actions per video, doing temporal action localization involves detecting all action labels, with their start and end time. Action localization methods [1, 2] utilize a two stage approach: 1. proposal generation and 2. action classification of each proposal.

Because proposal generation uses machine learning, it relies on annotated data. Such annotations for untrimmed videos have each action instance labeled with a start and end timestamp of the action and each video can have multiple, possibly overlapping, action instances [3]. Obtaining these labels is time consuming and expensive [4]. Moreover, the labeling of the action instances is subjective and error prone [5] due to a different understanding of action duration, thus affecting the results of the model trained using these labels [6]. Recent work in action recognition has shown that performance improves by using most discriminative portions of the video for training [7]. Similarly, work has been done

[1]https://github.com/NoorZia/punet



(a) Baseball Pitch

(b) Clean and Jerk

Fig. 1. Our proposed method. A single frame is labeled for each action instance. The detected results are shown for THUMOS'14 dataset. Using a single frame, the PU learning network is able to detect action boundaries with low error.

to optimize the segment length and recognize human actions with fewer frames [8, 9]. Using a single timestamp instead of start and end time for action recognition has been shown to be a reasonable compromise between performance and annotation effort [10]. In this paper, we question the need for more complex methods, and evaluate an extremely simple idea: We propose labeling a single action frame as "key frame" inside an action's temporal window (Figure 1) and evaluate the simplest approach we could find: Positive Unlabeled (PU) learning to detect action frames.

Our approach requires a single labeled key frame belonging to the action instance. The remaining frames are now a combination of background and unlabeled action frames, referred together as 'unlabeled data'. If we consider the unlabeled data as negative, the problem becomes imbalanced due to the high ratio of unlabeled data to positive which we tackle in a PU learning [11] setting where the true positives are iteratively removed from the unlabeled data.

Our contributions are: 1) Instead of adding complexity, we evaluate the very simple Positive Unlabeled learning setting for action proposal generation using just a single labeled frame per action instance. 2) This simplistic method is able to achieve good results. 3) PU-learning is data-efficient: It does well when using a small number of action instances, allowing another reduction of the annotation effort.

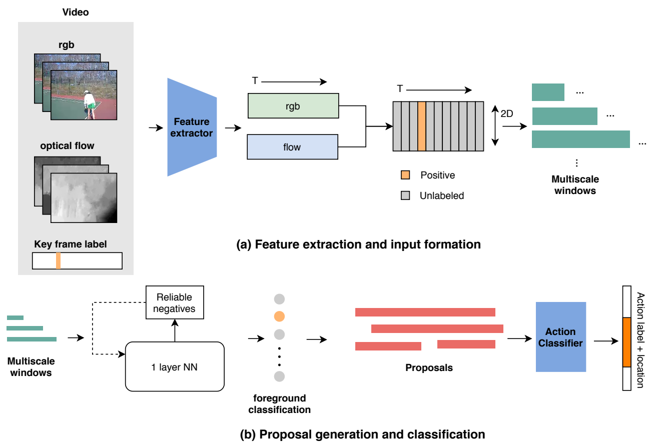


Fig. 2. Overview. We use one labeled point for each action instance. The input is devided in non-overlapping windows for training using PU learning at different scales with I3D encoded features to extract proposals.

2. METHOD

Problem definition. An untrimmed video sequence $X = \{x_n\}_{n=1}^{T}$ has T frames where x_n is the n-th frame in the video. Our single frame action annotations are $\Psi_g = \{\varphi_n = (t_{m,n})\}_{n=1}^{N_g}$ where $t_{m,n}$ is a selected frame at position m of the action instance n which we refer to as our key frame and N_g is the total number of action instances. For proposal generation, we have a binary action vs background classifier. We divide a video in non-overlapping windows, and a window is labeled positive if it contains a key frame.

PU learning. We draw inspiration from the simple and elegant PU-learning algorithm [11] to train the binary action vs background classifier. It finds negative samples that are most dissimilar from the positives by refining such 'reliable negatives'. A Positive versus Unlabeled classifier is trained and tested on the unlabeled training set where high-confidence predicted negative samples are deemed reliable negatives. The remaining unlabeled samples are removed from the training set. The size of the reliable negatives set is reduced iteratively by training a classifier using positive and reliable negative data and evaluating on reliable negative data points. Reliable negatives classified as positives are removed from the training set and this step is repeated until no positive classes are identified or the size of reliable negatives is less than positive samples. This step reduces the size of the negative samples and mitigates class imbalance.

Proposal generation and classification. The proposal generation module uses PU classifier to generate candidate proposals for each window scale. The results from different window scales are aggregated to get the final proposals. We use a state of the art action classifier [12] to classify our action proposals. The overview of PUNet can be seen in Figure 2.

3. EXPERIMENTS

Implementation details. We use I3D [13] pretrained on Kinetics [14] to extract RGB and optical flow features. The feature representations from RGB and optical flow are concatenated to obtain $(T \times 2D)$ features for a video of duration T. From untrimmed videos, we extract temporal windows of varying lengths, 16, 32, 48, 64, and 80 frames; with no overlap. For the proposal classifier, we use a single layer Multi-Layer Perceptron (MLP) with 100 hidden units. The single layer network is trained using adam optimizer and 10^{-4} learning rate. To extract the initial set of reliable negatives, the predicted negatives are thresholded based on their confidence score. The threshold value is set as 0.99.

Experimental Setup. We evaluate on THUMOS'14 [15], ActivityNet v1.2 and v1.3 [16] datasets. The THUMOS'14 dataset has temporal annotations for 20 classes with 200 training and 213 test videos. ActivityNet v1.2 has 100 action classes and 4,819 training, 2,383 validation and 2,480 test videos. ActivityNet v1.3 has 200 action classes, 10,024 training, 4,926 validation and 5,044 test videos. For ActivityNet, we use the validation videos for testing as the groundtruth for test videos is withheld. We measure performance with the F1-score. For temporal action proposal generation, the Average Recall (AR) as calculated at different IoU thresholds is used for evaluation. We also calculate AR with an average number of proposals (AR@AN) to determine relation between recall and number of proposals. For temporal action detection, mean average precision (mAP) is reported.

Results. A good proposal generation method should generate high recall with few proposals. PUNet compares well to most state of the art methods which use full supervision. We list the comparative results for THUMOS'14 in Table 1. We evaluate the quality of our generated proposals by comparing the recall at different tIoU thresholds (Figure 3). Our results have good recall at 100 proposals for tIoU 0.1 to 0.5. The results for action detection indicate that our extremely simple PUNet does well when compared to others. These results on THUMOS'14 are summarized in Table 2. Our method outperforms all weakly supervised methods except BaSNet [17], against which it shows a slight performance decrease while being more data efficient and having a simpler network design. Besides, our iterative approach takes around 4.6 minutes to train even on CPU. Our method can also be used to improve other single frame methods [10]. Compared to fully supervised methods, our method gives good performance while utilizing significantly less annotation effort. Table 3 shows our results on ActivityNet v1.2 and v1.3. For ActivityNet v1.2, we see that our method outperforms all weakly supervised methods except BaSNet and is not too far behind the fully supervised method. On ActivityNet v1.3, our method outperforms all weakly supervised methods including BaSNet.

Qualitative analysis. The qualitative analysis of our approach for key frame annotation is shown in Figure 1. The

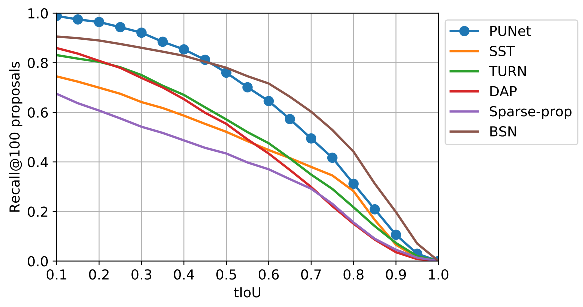


Fig. 3. Comparison of our method with the state of the art fully supervised methods on THUMOS'14 dataset. Recall with 100 proposals at different tIoU thresholds show PUNet has high recall compared to all fully supervised methods when tIoU < 0.5. At higher tIoUs, PUNet outperforms all fully supervised methods except BSN.

Table 1. Comparison of our method with other state of the art proposal generation methods on THUMOS'14 dataset in terms of AR@AN. Our method outperforms all fully supervised methods at AR@50 and AR@100 except BSN.

Supervision	Method	@50	@100
Full	DAPs [18]	13.56	23.83
	Sparse [19]	13.42	21.44
	SST [20]	19.90	28.36
	TURN [1]	21.86	31.89
	BSN [2]	35.41	46.06
Weak	PUNet (ours)	32.72	40.61

GT denotes groundtruth segments and the labels denote the key frame inputs to our network. Without any postprocessing, our proposal evaluation model is able to capture the full extent of the temporal duration and not just the key frames.

Data efficiency. We evaluate how the performance of PUNet changes when trained with a small dataset. We train BaSNet and PUNet with various training set sizes of THUMOS'14 dataset and report the mean average precision. All classes are included in each training set in an equal ratio.

Results are shown in Figure 4. For small training sets, PUNet outperforms BaSNet. As the data size increases, the performance becomes more similar for both. With 20 training samples, PUNet achieves 14.7% performance gain. The performance gain reduces as the training data set size increases.

Generalizability of proposals. We evaluate the generalization ability of PUNet by testing its performance on unseen action classes. We randomly leave one, two and three classes from our training set and test on our test set containing all 20 classes of THUMOS'14 data. As shown in table 4, there is only a slight performance decrease when testing on unseen classes and the method is able to generate high quality proposals on unseen classes.

Table 2. Comparison of our method with the state of the art methods on the THUMOS'14 dataset. Average mAP is reported at IoU thresholds from 0.1 to 0.5. Weak * indicate use of additional information in weakly supervised approach. PUNet outperforms most weakly supervised and some fully supervised methods while utilizing less annotations.

Supervision	Method	AVG mAP
Full	Yuan et al. [21]	35.7
	TAL-Net [22]	52.3
	P-GCN [23]	61.6
Weak (video)	UntrimmedNet [12]	29.0
	Liu et al. [24]	40.9
	BaSNet [17]	43.6
Weak (single frame)*	SF-Net [10]	51.5
	PUNet (ours)	42.1
	SF-Net + PUNet (ours)	53.6

Table 3. Comparison on ActivityNet (Anet) v1.2 and v1.3 with the current state of the art methods. PUNet has comparable performance to fully supervised method and outperforms most weakly supervised methods for action localization.

Supervision	Method	AVG mAP	
		ANet v1.2	ANet v1.3
Full	S-CNN [25]	26.6	-
	CDC [26]	-	23.8
Weak	Liu et al. [24]	22.4	21.2
	BaSNet [17]	24.3	22.2
Weak (single frame)*	PUNet (ours)	23.7	22.5

Annotation speed for different settings. Annotation time required to label a single key frame and the full segment is measured for some videos from THUMOS'14 dataset. Five videos are selected from THUMOS'14 dataset with different classes and six annotators are chosen. Three annotators are asked to label the full segment and the remaining three are asked to label a single frame for every action occurrence. On average, one minute video takes 65 seconds for single frame labeling and 250 seconds for full segment labeling.

3.1. How many annotations per video are actually needed?

Videos in THUMOS'14 have 15 action instances on average which are spread unevenly among the videos with a standard deviation of 24, and range from 1 to 128 per video. The total labeled action instances in the training set are shown in Figure 5. We evaluate whether annotations for all instances are needed to get an effective action proposal network. F1-score is used to compare the maximum annotations per video ranging from 1 to 128.

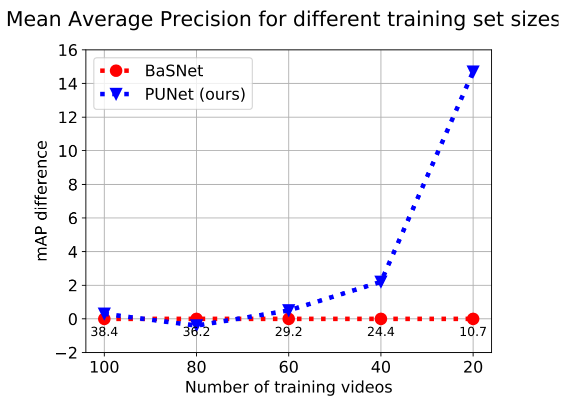


Fig. 4. Data Efficiency. We compare the performance of BaSNet and PUNet when training data is reduced per class from 1 to 5 videos. For small training set, PUNet has a higher relative performance. The performance becomes similar when training set size increases.

Table 4. Generalization evaluation of PUNet on THUMOS'14 dataset. Action classes are removed from the training set and the resulting model is evaluated on the full test set (seen + unseen classes) containing 20 classes.

# classes in training set	AR@50	AR@100
17	31.8	38.5
18	32.4	39.3
19	32.5	40.2
20	32.7	40.6

After a maximum limit of 6 annotations per video, F1-score has low variance (Figure 5). PUNet is able to identify the unlabeled key frames effectively. Results indicate that not all annotations are necessary to achieve a good performance.

In Table 5, we show that not all action annotations are required for good detection performance by training fully and weakly supervised action localization networks. Thus, we set the number of maximum annotations per video to 6 action instances. The number of action instances reduces from 3007 to 947. We train PUNet with a maximum of 6 annotations per video and obtain a slight performance drop of 0.8%. Similarly, BaSNet [17] is trained with the reduced video size and the results show a 1.5% reduction in mAP. Fully supervised method, GTAD [27], is trained with only six labeled action instances and the rest of the data is unlabeled. Interestingly, the performance increases by 0.9%. The results indicate that the methods do not need all the labels to obtain good results.

4. CONCLUSION

We use key frame level supervision for training temporal action proposal model in a PU-learning algorithm on three untrimmed datasets. Compared to fully supervised methods

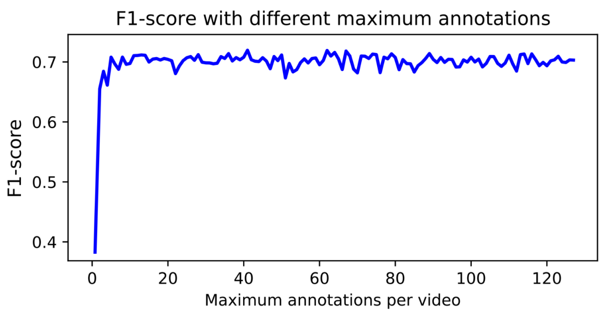


Fig. 5. Effect of changing the maximum number of annotations per video on the binary classifier performance. After 6 annotations per video, the performance does not change much and the standard deviation reduces. The mean value of F1-score from 1-128 annotations is 0.69 $\pm$ 0.05, and mean F1-score from 6-128 annotaions is 0.70 $\pm$ 0.008. Our method can achieve good results without using all the annotations.

Table 5. Effect of using limited annotations on action localization for THUMOS'14 dataset. We set the maximum annotations per video to 6 to train these models. The action instances needed reduce by one-third from 3007 to 947. The performance only decreases slightly for weakly supervised methods and increases by 0.9% for fully supervised method.

Supervision	Method	Whole	Partial
Full	GTAD [27]	55.4	56.3
Weak	BaSNet [17]	43.6	42.1
Weak*	PUNet (Ours)	42.1	41.3

and other weakly supervised methods, this extremely simple approach generates proposals with high recall and high temporal overlap. Experimental evaluation on THUMOS'14 shows that: (i) Using a key frame annotation gives comparable performance to using fully supervised annotation which uses start and end annotations, (ii) All action instances from one video are not necessary to achieve good detection results, (iii) Our results are comparable to the state of the art methods and data efficient. We conclude that annotation effort can be significantly reduced by labeling key frames and for long untrimmed videos, only a limited number of action instances need to be labeled and trained to achieve similar results.

5. REFERENCES

[1] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *ICCV*, 2017.

[2] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang,

and Ming Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *ECCV*, 2018.

[3] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *IJCV*, 2017.

[4] Soroosh Poorgholi, Osman Semih Kayhan, and Jan C van Gemert, "t-eva: Time-efficient t-sne video annotation," *arXiv preprint arXiv:2011.13202*, 2020.

[5] Davide Moltisanti, Michael Wray, Walterio Mayol-Cuevas, and Dima Damen, "Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video," in *ICCV*, 2017.

[6] Scott Satkin and Martial Hebert, "Modeling the temporal extent of actions," in *ECCV*, 2010.

[7] Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce, "Automatic annotation of human actions in video," in *ICCV*, 2009.

[8] Konrad Schindler and Luc Van Gool, "Action snippets: How many frames does human action recognition require?," in *CVPR*, 2008.

[9] Xiaodong Yang and YingLi Tian, "Effective 3d action recognition using eigenjoints," *Journal of Visual Communication and Image Representation*, 2014.

[10] Fan Ma, Linchao Zhu, Yi Yang, Shengxin Zha, Gourab Kundu, Matt Feiszli, and Zheng Shou, "Sf-net: Single-frame supervision for temporal action localization," *ECCV*, 2020.

[11] Donato Hernández Fusilier, Manuel Montes-y Gómez, Paolo Rosso, and Rafael Guzmán Cabrera, "Detecting positive and negative deceptive opinions using pu-learning," *Information processing & management*, 2015.

[12] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in *CVPR*, 2017.

[13] Joao Carreira and Andrew Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al., "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[15] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[16] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *CVPR*, 2015, pp. 961–970.

[17] Pilhyeon Lee, Youngjung Uh, and Hyeran Byun, "Background suppression network for weakly-supervised temporal action localization.," in *AAAI*, 2020.

[18] Victor Escorcia, Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem, "Daps: Deep action proposals for action understanding," in *ECCV*, 2016.

[19] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *CVPR*, 2016.

[20] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles, "Sst: Single-stream temporal action proposals," in *CVPR*, 2017.

[21] Zehuan Yuan, Jonathan C Stroud, Tong Lu, and Jia Deng, "Temporal action localization by structured maximal sums," in *CVPR*, 2017.

[22] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *CVPR*, 2018.

[23] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan, "Graph convolutional networks for temporal action localization," in *ICCV*, 2019.

[24] Daochang Liu, Tingting Jiang, and Yizhou Wang, "Completeness modeling and context separation for weakly supervised temporal action localization," in *CVPR*, 2019.

[25] Zheng Shou, Dongang Wang, and Shih-Fu Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *CVPR*, 2016, pp. 1049–1058.

[26] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *CVPR*, 2017.

[27] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem, "G-tad: Sub-graph localization for temporal action detection," in *CVPR*, 2020.