SEMI-SUPERVISED OBJECT DETECTION WITH SPARSELY ANNOTATED DATASET

Jihun Yoon, Seungbum Hong, and Min-Kook Choi

hutom
Republic of Korea

ABSTRACT

When training an anchor-based object detector with a sparsely annotated dataset, the effort required to locate positive examples can cause performance degradation. Because anchor-based object detection models collect positive examples under IoU between anchors and ground-truth bounding boxes, in a sparsely annotated image, some objects that are not annotated can be assigned as negative examples, such as backgrounds. We attempt to solve this problem with two approaches: 1) using an anchor-less object detector and 2) using a single-object tracker for semi-supervised learning-based object detection. The proposed technique performs bidirectional single-object tracking from sparsely annotated bounding boxes as starting points in videos to obtain dense annotations. On applying our method to the EPIC-KITCHENS-55 dataset, we were able to achieve **runner-up** performance in the **Unseen** section, while achieving the **first place** in the **Seen** section of the EPIC-KITCHENS 2020 object detection challenge under IoU > 0.5 on the EPIC-KITCHENS 2020 object detection challenge.

***Index Terms*—** Object detection, semi-supervised learning, object tracking, ensemble learning, sparse annotation

1. INTRODUCTION

Owing to the rapid evolution of convolutional neural networks (CNNs), the performance of object recognition networks, including object detection, has significantly improved [1]. The object detection dataset has also been changed from a low-complexity dataset, such as PASCAL VOC [2], to a high-complexity dataset such as MS-COCO [3]. Among the object detection datasets, the relatively latest EPIC-KITCHENS-55 dataset is the largest egocentric video benchmark, offering a unique viewpoint on people's interaction with objects in multiple action clips. It has the following characteristics that are different from other object detection datasets [4].

- Training annotations for object detection only capture the active objects interacting with people in action clips, which cause sparse annotations.
- The difference in the number of annotations between the few and many shot classes is large, depending on the distribution of the appearances of the objects in the training dataset.

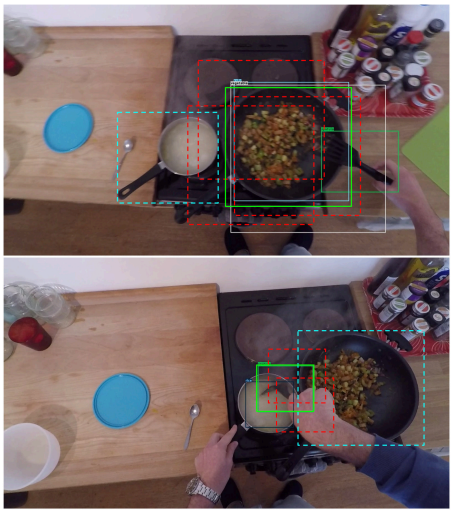


Fig. 1. **Example of anchor-based detector training on a sparsely annotated dataset.** The solid green line boxes represent given ground-truth annotations, and the red dotted line boxes are examples of positive examples based on anchors with IoU and a ground-truth box. Light blue dashed line boxes indicate objects to learn but not annotated in all training images. On the top image, a pot could be assigned as a negative example and also a fan on the bottom image. These missing annotations can degrade the performance of anchor-based detector.

As described above, the annotation of EPIC-KICTHENS-55 dataset for object detection is provided in a different way from the existing datasets, and consequently, it is difficult to apply the method of training the existing object detection model as it is. In general, in the case of detectors that train positive examples based on anchors [5, 6] or detectors that train the objectness of a candidate object with the structure of an RPN [7, 8], batch sampling is performed considering the IoU with the ground-truth bounding box for effective training. However, if anchor-based hard example mining is performed on a sparsely annotated training image, the efficiency of training is hindered by the distribution of objects near the ground truth bounding box. Figure 1 shows the negative effects of collecting positive examples when training anchor-based object detectors with sparse annotations.

We trained the object detector using two approaches to

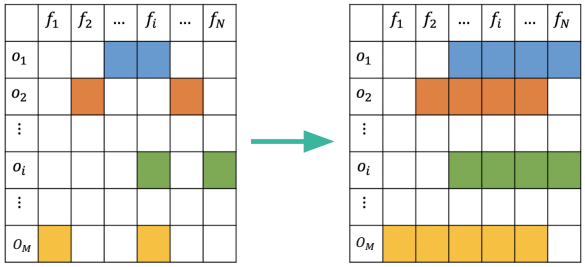


Fig. 2. **The final goal of the proposed supervised learning.** f denotes an action clip composed of N frames, and o denotes a total of M objects present in the action clip. We performed semi-supervised learning through bidirectional tracking to obtain dense labels for all learnable objects present in the action clip.

solve this problem. First, we used an anchor-less object detector, the Fully Convolutional One-Stage Object Detection (FCOS) network [9] and consequently improved the performance considerably more than using anchor-based object detector. Second, as the EPIC-KITCHENS-55 dataset originated from videos, we used single-object tracking to semi-supervise all detectable objects between consecutive frames from sparse annotations. The bounding box label existing in a specific frame is set as an initial bounding box in the time domain and used as an input for a single-object tracker. If the box size of the new predictive output of the tacker changes less than a threshold compared to the previous predictive output, it was assumed to be a pseudo annotation. Figure 2 illustrates the proposed goal of semi-supervised learning.

Using the proposed approach, we were able to train an object detection network effectively using the EPIC-KITCHENS-55 dataset. Subsequently, a joint NMS-based ensemble [10] was performed for the FCOS models with inhomogeneous backbones. Therefore, we achieved the ***first place*** in the *Seen* evaluation set and ***runner-up*** in the *Unseen* evaluation set under the IoU > 0.5 of the EPIC-KITCHENS 2020 object detection challenge.

2. RELATED WORK

Object Detection. CNN-based object detection models are largely divided into one-stage or two-stage models. In the one-stage model, (e.g., YOLO [5, 11], SSD [6, 12], and RetinaNet [13]), the process of predicting the class and position of an object is performed under one structure. In general, it is known that the regression accuracy of the model is lower than that of the two-stage model, because classification and regression are performed in one structure. In the case of the two-stage model, prior knowledge of the location of the object is estimated from the region proposal network (RPN) [7]. RPN determines objectness using a class-agnostic subnet, and class-aware detection is performed through the subsequent head structure. Faster R-CNN [7], R-FCN [8], Cascade R-CNN [14], and Cascade RPN [15] are representative of various head structures with relatively high regression accuracy. Models such as RefineDet [15] that combines one-stage and two-stage models have also been proposed. On the other hand, detectors utilizing other parameterizations for bounding box regression rather than structural advantages, such as FCOS [9], have also been proposed.

Semi-supervised learning for object detection. Object detection using semi-supervised learning is used in situations where it is difficult to manually acquire a sufficient number of annotations to learn, or when pseudo labels are to be obtained from a relatively large number of unlabeled data [16, 17, 18, 19]. In [16], the author's proposed an iterative framework for evaluating and retraining pseudo-labels using pre-trained object detectors and robust trackers to obtain good pseudo-labels in successive frames. In [17], it was possible to achieve improved detection performance in the Open Image Dataset V4 by utilizing part-aware sampling and RoI proposals to obtain good pseudo labels for sparsely annotated large-scale datasets. In [18], to efficiently use unlabeled data from the MS-COCO dataset, co-current matrix analysis was used to generate good pseudo labels by using prior information of the labeled dataset. The proposed single-object tracker-based semi-supervised learning is similar to [16] in that it uses a tracker, but has a difference in obtaining dense annotation information for a specific image by using the existing lean annotation information. Simultaneously, because the object detector is not used as the initial input for tracking, training is not applied as an iterative training scenario.

Single-object visual tracking. In single-object tracking [20, 21, 22, 23], the Siamese network-based visual tracker showed balanced accuracy and speed across various datasets. The Siamese network-based tracker is trained with the similarity of the CNN feature for the target image and the input image for tracking. We used SiamMask [22] as a single object tracker, which uses box and mask information together with the similarity of features to the tracking target.

3. SEMI-SUPERVISED LEARNING WITH SINGLE OBJECT TRACKER

The bounding boxes for the objects in the EPIC-KITCHENS-55 dataset were not densely annotated in all training frames, but rather sparsely in the action sequences. We used a single-object tracker to achieve the goal shown in Figure 2 with an automated procedure. Among the various single-object trackers, SiamMask [22] was used, which showed a balanced performance for tracking accuracy and speed. We performed bidirectional tracking with the SiamMask trained from the DAVIS dataset [24], using each bounding box as the initial value for a single object. The details of forward tracking with SiamMask for an action clip input are described in Algorithm 1. Algorithm 1 is used in the same manner as backward track-

Algorithm 1: Forward tracking

Input: Action clip (A), pre-trained tracking model (T), a set of bbox for initial input (BB), threshold of tracking score (ρ_1), threshold of a box size difference between two pair of tracked bbox (ρ_2)
Output: BB in Q from T
Initialize an empty queue Q
while *bbox* $b_{c,i}$ *with class* c *at* i*-th frame available from* BB **do**
  Get a list of frames FF in forward from i-th frame in A;
  Initialize T with b_i from A;
  Initialize a variable pre_s with a size of $b_{c,i}$ to store a size of object from T at previous frame;
  while *each frame in* FF **do**
    Get a bbox $b_{c,k}$ from T at k-th frame;
    $crnt_s :=$ a size of $b_{c,k}$;
    if $|prev_s - crnt_s| \leq \rho_1$ **then**
      $prev_s := crnt_s$;
      Add $b_{c,k}$ to Q;
    else
      break;
    end
  end
end

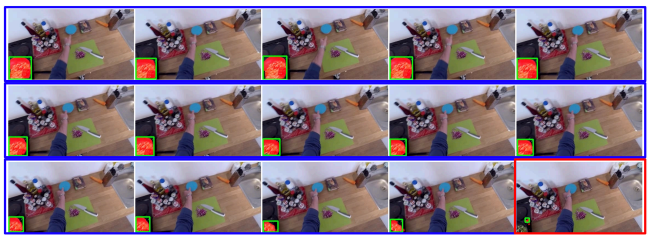


Fig. 3. **Example of the result of Algorithm 1.** Frames marked with blue boxes are frames that have been tracked with the same object since tracking started, and a frame marked with a red box is a frame whose tracking is terminated due to the termination condition of Algorithm 1.

ing to complete the bidirectional tracking. Figure 3 shows an example of the start and end of tracking according to Algorithm 1 on a single object, and Figure 4 shows training images with pseudo labels generated after tracking an object.

4. EPIC-KITCHENS OBJECT DETECTION RESULTS

Training details. We used Faster R-CNN [7] and Cascade R-CNN [14] as anchor-based detectors and FCOS [9] as anchor-less detectors to compare the performance of the EPIC-KITCHENS-55 dataset. ResNet-50, ResNet-101, ResNeXt-101, and HRNetV2p-W32 pre-trained with ImageNet were used for backbones, and training details for each

Fig. 4. **Changes in training images after tracking.** An example of the final annotations to be used for semi-supervised learning is shown on the training images indicated by the red dotted line.

Table 1. **Performance comparison of anchor-based and anchor-less detectors.** If the model name has a '+', it is the result of evaluation using tracker-based semi-supervised learning. The highest performance in a single model and the highest performance in an entire model are shown in bold.

Detector	backbone	Seen			Unseen		
-	-	> 0.05	> 0.5	> 0.75	> 0.05	> 0.5	>0.75
Faster R-CNN	ResNet-101	37.54	28.64	6.92	32.83	23.16	5.55
Cascade R-CNN	HRNet-V2P	30.44	24.17	8.73	23.87	18.05	6.81
FCOS	HRNet-V2P	48.44	34.87	**11.02**	**43.88**	**30.68**	**9.27**
FCOS	ResNet-50	46.96	34.51	10.09	42.46	29.49	7.48
FCOS	ResNet-101	49.77	35.8	10.15	43.39	28.98	7.86
FCOS	ResNeXt-101	48.17	33.95	9.86	41.79	27.27	7.19
FCOS+	ResNet-101	**50.27**	**35.89**	10.57	43.14	29.82	7.76
FCOS Ensemble+	-	**58.27**	**44.48**	**15.36**	**55.72**	**41.12**	**12.5**

combination of backbone and head structure are shown in Supplementary (Supp.) Table 1. All the experiments were conducted using the MMDetection library [25].

Anchor-based vs. anchor-less detector. Table 1 shows the training performance of a single model of an anchor-based detector and an anchor-less detector. According to Table 1, the performance of anchor-less detectors is generally better than that of anchor-based detectors. Supp. Figure 1 shows that the training loss curves and anchor-less detector curves are more stable than anchor-based curves. Simultaneously, Table 1 shows the performance change of the FCOS model according to different backbones. For a single model, it was confirmed that the FCOS model utilizing the ResNet-101 backbone achieved the best generalization performance in the Seen set, and the HRNet backbone model performed the best in the Unseen set.

Semi-supervised learning. We used the pre-trained SiamMask model from the DAVIS dataset [24] to generate pseudo labels, and the labels were used for training the FCOS with ResNet 101. Table 1 shows that the generalization performance of the FCOS models under $IoU > 0.5$, is improved when using semi-supervised learning based on single-object tracking. However, we only generated pseudo-labels in a few shot classes. As shown in Table 2, the performance of our semi-supervised FCOS model only shows an improvement in a few shot classes. If our proposed method were applied in many shot classes, there would be much more improvement.

Inhomogeneous backbone ensemble. We performed ensemble modeling for each trained model in Table 1 to achieve the best detection performance. We used the joint NMS technique

Table 2. Performance comparison of our method and inhomogeneous backbones in few shot classes and many shot classes of the Seen images. Detectors are sorted in descending order by mAP with IoU > 0.5 in few shot classes.

Detector	backbone	Few shot classes			Many shot classes		
-	-	> 0.05	> 0.5	> 0.75	> 0.05	> 0.5	>0.75
FCOS Ensemble+	-	47.44	35.75	14.32	60.77	46.5	15.6
FCOS+	ResNet-101	39.38	27.28	7.43	52.79	37.89	10.83
FCOS	ResNet-101	37.69	25.75	6.26	52.57	38.13	11.05
FCOS	ResNeXt-101	35.43	25.68	8.63	51.13	35.87	10.14
FCOS	HRNet-V2P	36.98	25.03	10.57	51.1	37.16	11.13
FCOS	ResNet-50	32.89	24.14	9.28	50.21	36.91	10.28

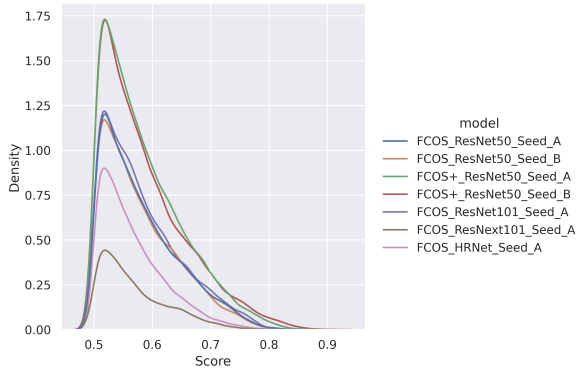
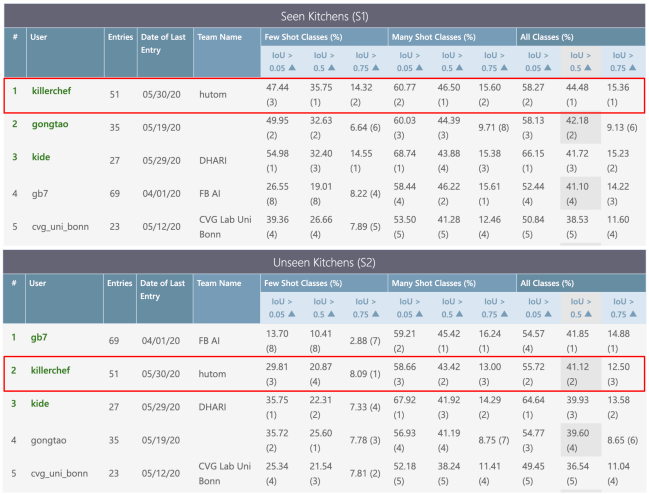


Fig. 5. The probability distributions of prediction score in the test images. For equal comparison, all models were trained for 24 epochs and two random seeds, A and B.

[10], where we can achieve an ensemble by applying NMS to bounding boxes and chose 300 boxes in order of high scores, owing to a submission rule of the challenge that allows only 300 boxes in a test image for the evaluation. Table 1 shows the performance change of the Seen and Unseen sets according to the ensemble combination. We confirmed that the ensemble model can achieve a very large performance improvement compared to a single model, and our SSL method is also helpful for the ensemble. Figure 6 shows the performance published on the EPIC-KITCHENS object detection challenge page. Finally, we achieved the ***first rank*** rank in the *Seen* set and ***runner-up*** performance in the *Unseen* set through an inhomogeneous backbone ensemble under IoU > 0.5.

Discussion. We analyzed the performances in detail by breaking down into few-shot and many shot classes, as shown in Table 2. FCOS with ResNet 101 was the best in many shot classes but not in a few shot classes, and FCOS with HRNet was third in many shot classes but not in few-shot classes. We found that object detectors with different backbones not only exhibited general performance differences, but also had different views of recognition. This also occurred for the detector trained using our method. Our semi-supervised FCOS with the ResNet-101 model performed the best in a few shot classes and the second best in many shot classes. We visualized the inference results for each model in Supp. Figure 2, and as each model shows different predictions rather than just for a better performance model, including predictions of a worse model. We also compared how the prediction score


Seen Kitchens (S1)

#	User	Entries	Date of Last Entry	Team Name	Few Shot Classes (%)			Many Shot Classes (%)			All Classes (%)		
					IoU > 0.05	IoU > 0.5	IoU > 0.75	IoU > 0.05	IoU > 0.5	IoU > 0.75	IoU > 0.05	IoU > 0.5	IoU > 0.75
1	killerchef	51	05/30/20	hutom	47.44 (3)	35.75 (1)	14.32 (2)	60.77 (2)	46.50 (1)	15.60 (2)	58.27 (2)	44.48 (1)	15.36 (1)
2	gongtao	35	05/19/20		49.95 (2)	32.63 (2)	6.64 (6)	60.03 (3)	44.39 (3)	9.71 (8)	58.13 (3)	42.18 (2)	9.13 (6)
3	kide	27	05/29/20	DHARI	54.98 (1)	32.40 (3)	14.55 (1)	68.74 (1)	43.88 (4)	15.38 (3)	66.15 (1)	41.72 (3)	15.23 (2)
4	gb7	69	04/01/20	FB AI	26.55 (8)	19.01 (8)	8.22 (4)	58.44 (4)	46.22 (2)	15.61 (1)	52.44 (4)	41.10 (4)	14.22 (3)
5	cvg_uni_bonn	23	05/12/20	CVG Lab Uni Bonn	39.36 (4)	26.66 (4)	7.89 (5)	53.50 (5)	41.28 (5)	12.46 (4)	50.84 (5)	38.53 (5)	11.60 (4)

Unseen Kitchens (S2)

#	User	Entries	Date of Last Entry	Team Name	Few Shot Classes (%)			Many Shot Classes (%)			All Classes (%)		
					IoU > 0.05	IoU > 0.5	IoU > 0.75	IoU > 0.05	IoU > 0.5	IoU > 0.75	IoU > 0.05	IoU > 0.5	IoU > 0.75
1	gb7	69	04/01/20	FB AI	13.70 (8)	10.41 (8)	2.88 (7)	59.21 (2)	45.42 (1)	16.24 (1)	54.57 (4)	41.85 (1)	14.88 (1)
2	killerchef	51	05/30/20	hutom	29.81 (3)	20.87 (4)	8.09 (1)	58.66 (3)	43.42 (2)	13.00 (3)	55.72 (2)	41.12 (2)	12.50 (3)
3	kide	27	05/29/20	DHARI	35.75 (1)	22.31 (2)	7.33 (4)	67.92 (1)	41.92 (3)	14.29 (2)	64.64 (1)	39.93 (3)	13.58 (2)
4	gongtao	35	05/19/20		35.72 (2)	25.60 (1)	7.78 (3)	56.93 (4)	41.19 (4)	8.75 (7)	54.77 (3)	39.60 (4)	8.65 (6)
5	cvg_uni_bonn	23	05/12/20	CVG Lab Uni Bonn	25.34 (4)	21.54 (3)	7.81 (2)	52.18 (5)	38.24 (5)	11.41 (4)	49.45 (5)	36.54 (5)	11.04 (4)

Fig. 6. EPIC-KITCHENS 2020 object detection challenge evaluation page. The entry marked with a red box is the final performance evaluated by the our proposed approach. Each entry is ranked under IoU > 0.5 evaluation.

distributions were different in the test images, as shown in Figure 5.

5. CONCLUSION

We performed single-object tracker-based semi-supervised object detection to effectively train a dataset with sparse annotations on sequence images. The EPIC-KITCHENS-55 dataset was used to verify the utility of the proposed technique, and it showed good performance in the ensemble as well as in the single model. However, it needs to be analyzed more closely with semi-supervised learning about the advantages and disadvantages of the anchor-based model, and there is a limitation in that a simple rule-based engine is used to obtain a pseudo label. For future improvement, it is necessary to perform quantitative analysis on the effect of anchors and RPN on sparse annotation data training, while simultaneously considering how to improve the tracking rules, or utilize the results obtained in the tracking process during training.

Acknowledgement. This work was supported by the Korea Medical Device Development Fund grant funded by the Korea government (the Ministry of Science and ICT, the Ministry of Trade, Industry and Energy, the Ministry of Health & Welfare, the Ministry of Food and Drug Safety) (Project Number: 202012A02-02)

6. REFERENCES

[1] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen, "Deep learning for generic object detection: A survey," *arXiv:1809.02165*, 2018.

[2] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2015.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro PeronaDeva, Ramanan, Piotr Dollár, and C. Lawrence Zitnick, "Microsoft coco: Common objects in context," in *In Proc. of ECCV*, 2014.

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *In Proc. of ECCV*, 2018.

[5] Joseph Redmon and Ali Farhadi, "Yolo9000: Better, faster, stronger," in *In Proc. of CVPR*, 2017.

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, "Ssd: Single shot multibox detector," in *In Proc. of ECCV*, 2016.

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *In Proc. of NIPS*, 2015.

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *In Proc. of NIPS*, 2016.

[9] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "Fcos: Fully convolutional one-stage object detection," in *In Proc. of ICCV*, 2019.

[10] Heechul Jung, Min-Kook Choi, Jihun Jung, Jin-Hee Lee, Soon Kwon, and Woo Young Jung, "Resnet-based vehicle classification and localization in traffic surveillance systems," in *In Proc. of CVPR*, 2017.

[11] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, 2018.

[12] Cheng-Yang Fu, Wei Liu, Ananth Ranga, Ambrish Tyagi, and Alexander C. Berg, "Dssd : Deconvolutional single shot detector," *arXiv:1701.06659*, 2017.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *In Proc. of ICCV*, 2017.

[14] Zhaowei Cai and Nuno Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *In Proc. of CVPR*, 2018.

[15] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li, "Single-shot refinement neural network for object detection," in *In Proc. of CVPR*, 2018.

[16] Ishan Misra, Abhinav Shrivastava, and Martial Hebert, "Watch and learn: Semi-supervised learning for object detectors from video," in *In Proc. of CVPR*, 2015.

[17] Yusuke Niitani, Takuya Akiba, Tommi Kerola, Toru Ogawa, Shotaro Sano, and Shuji Suzuki, "Sampling techniques for large-scale object detection from sparsely annotated objects," in *In Proc. of CVPR*, 2019.

[18] Min-Kook Choi, Jaehyeong Park, Jihun Jung, Heechul Jung, Jin-Hee Lee, Woong Jae Won, Woo Young Jung, Jincheol Kim, and Soon Kwon, "Co-occurrence matrix analysis-based semi-supervised training for object detection," in *In Proc. of ICIP*, 2018.

[19] Yuzheng Xu, Yang Wu, Nur Ssabrina Binti Zuraimi, Shohei Nobuhara, and Ko Nishino, "Video region annotation with sparse bounding boxes," in *Proc. of BMVC*, 2020.

[20] Luca Bertinetto, Jack Valmadre, João F. Henriques, Andrea Vedaldi, and Philip H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *In Proc. of ECCV*, 2016.

[21] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu, "High performance visual tracking with siamese region proposal network," in *In Proc. of CVPR*, 2018.

[22] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip H.S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *In Proc. of CVPR*, 2019.

[23] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *In Proc. of CVPR*, 2019.

[24] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbelaez, Alexander Sorkine-Hornung, and Luc Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv:1704.00675*, 2017.

[25] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv:1906.07155*, 2019.