Cannot See the Forest for the Trees: Aggregating Multiple Viewpoints to Better Classify Objects in Videos

Supplementary Material

Figure 5. (Top) QDTrack classifies the tracklet as a **monitor** by looking at only the top part, because QDTrack has an access only to per-frame information. (Bottom) On the contrary, our proposed set classifier gathers all information of the tracklet to make the prediction, resulting in the correct classification: **laptop**.

A. Additional Implementation Details

RoI sampling. For a region proposal from RPN to be regarded as a foreground, its Intersection-over-Union (IoU) overlap with a ground-truth box should be greater or equal to 0.7. We use only positive samples (predicted as foregrounds) for the items of the augmented tracklets.

Length of tracklets. In Table 7, we find that the lengths of tracklets that are generated during training affect the overall performance. We examine that [16, 32] is the most suitable range of lengths as the average length of tracklets predicted by QDTrack is 21.24.

If the length of a given tracklet is greater or equal to 32, we directly use the tracklet without modifications. If a given tracklet of length $L < 32$, we duplicate $\lfloor 32/L \rfloor$ times, so that the length gets similar to 32. For example, let a tracklet of length 6 is given. We modify the tracklet to be composed of $\lfloor 32/6 \rfloor = 5$ duplications of itself, resulting in the new tracklet with the length of $6 \times 5 = 30$. By making the inference be similar to the training setting, the set classifier can better predict the category.

Multiple class assignments. As the set classifier is trained on the soft labels, it shows smooth outputs which sometimes harm the accuracy. Therefore, we assign multiple class scores to a tracklet and top-k scored tracklets are used for the inference.

B. Qualitative Results

We briefly explain the execution steps of our model again using Fig. 5. Due to the large vocabulary of TAO [9], object trackers predict numerous tracklets which harm the visibility. Therefore, we select and visualize only the results of the tail classes to increase visibility.

Because our model is built on top of QDTrack [35], the input video is first fetched by [35]. QDTrack outputs tracklet predictions, each composed of bounding boxes at different frames that share the same identity. In Fig. 5, the top row shows a predicted tracklet from [35], which is wrongly classified as **monitor**. The failure is understandable as QDTrack uses per-frame information and only the top part of the object is visible for the majority of frames.

Taking an input that is composed of the regional features according to the bounding boxes in the tracklet, our proposed set classifier re-classifies the category of the given tracklet. The set classifier does not make predictions by looking only at partial information, but puts all information of the tracklet into consideration. As visualized at the bottom row in Fig. 5, the set classifier successfully predicts the category of the object, **laptop**. While the estimations at bounding boxes of the tracklet is the exact same, using the set classifier brings huge gain of accuracy by correcting categories (Tab. 1 and Tab. 2).

We provide more qualitative results that contain failure cases of QDTrack, but correctly predicted by the set classifier (Fig. 6, Fig. 7, and Fig. 8). Same to Fig. 5, each top row shows wrong predictions of QDTrack that have not yet passed through the set classifier, and the bottom rows show the predictions of our proposed set classifier. We find our set classifier is more robust and accurate in classifying large vocabulary. All qualitative results are best viewed in zoom on screen.

C. Additional Results

Use of better detectors. We further provide a comparison to demonstrate the effect of the set classifier. In Tab. 10, we report the scores of additional approaches that can improve the long-tail detection quality such as more complex back-

	QDTrack	Set Classifier	Tracking (Extension of Table 2)					Detection (Extension of Table 1)								
			AP_{50}	AP_{75}	$AP_{50:95}$	FLOPS(T)	FPS	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP_r	AP_c	AP_f
(1)	R101 + CE	✗	15.8	6.4	7.3	11.17	5.6	17.2	29.1	17.4	5.7	**13.1**	22.0	6.5	11.9	**25.9**
(2)	X101 + CE	✗	17.3	6.8	7.8	17.33	1.9	18.1	29.2	18.9	5.9	12.9	22.5	12.7	**14.0**	23.6
(3)	R101 + SS	✗	16.5	6.1	7.5	11.17	5.6	17.5	**29.9**	17.5	5.3	12.4	21.3	8.5	13.5	24.3
(4)	R101 + CE	✓	19.9	**8.3**	9.6	11.73	5.4	18.3	29.5	18.9	**6.7**	11.9	**23.7**	13.6	**14.0**	23.8
(5)	R101 + SS	✓	**20.5**	8.1	**9.7**	11.73	5.4	**18.9**	**29.9**	**20.1**	6.3	12.9	23.5	**14.3**	13.8	24.9

Table 10. X101 denotes ResNeXt-101 32x8d [53]. CE and SS denote Cross Entropy Loss and Seesaw Loss [48], respectively.

Method	RPN [38]	Set Classifier	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN [54]	✓	✗	31.9	53.7	32.3	32.5	37.7
		✓	**34.2**	**55.9**	**36.4**	**35.0**	**41.5**
QDTrack [35]	✓	✗	34.4	55.1	38.4	33.5	41.6
		✓	**37.7**	**60.4**	**39.8**	**35.6**	**45.8**
CrossVIS [35]	✗	✗	36.6	57.3	39.7	36.0	42.0
		✓	**39.1**	**61.5**	**42.6**	**38.0**	**47.7**

Table 11. Effects of the set classifier on top of online methods evaluated on YouTube-VIS 2019. (Extension of Table 3)

#imgs / batch	$TrackAP_{50}$	$TrackAP_{75}$	$TrackAP_{50:95}$
1	17.4	7.7	8.6
2	19.9	8.3	9.6
4	20.6	7.1	9.5

Table 12. Comparison of different number of frames from a video.

#enc layers	$TrackAP_{50}$	$TrackAP_{75}$	$TrackAP_{50:95}$
1	18.1	7.2	8.9
2	18.7	7.8	9.2
3	19.9	8.3	9.6

Table 13. Comparison of different number of encoder layers in the set classifier.

bones and improved classification objectives. ResNeXt-101 [53] outperforms ResNet-101 in most cases, and Seesaw loss [48] is a recently proposed method that has shown improvements over BAGS [25] and SimCal [49] on long-tailed classifications. From the results in Tab. 10, we validate that our set classifier is what brings the most improvements on both tracking and detection compared to the bigger backbone and the advanced loss function.

Inference efficiency. We use a single TITAN Xp GPU to measure FPS of different settings on TAO validation benchmark in Tab. 10. Attaching the set classifier leads to the marginal loss of FPS (-0.2) from 5.6 FPS of the original QDTrack. It is because the set classifier is composed of $N_E = 3$ transformer layers and only executes at the end to finalize the class of tracklets. We also report FLOPS(T)[2] measured while processing a video in TAO, which is composed of 40 frames with 800×1280 resolution. Compared to Tab. 10 (2) which uses the heavier backbone, our set classifier is much more efficient and obtains higher accuracy.

[2]Measured using `flop_count` function of `fvcore==0.1.5`.

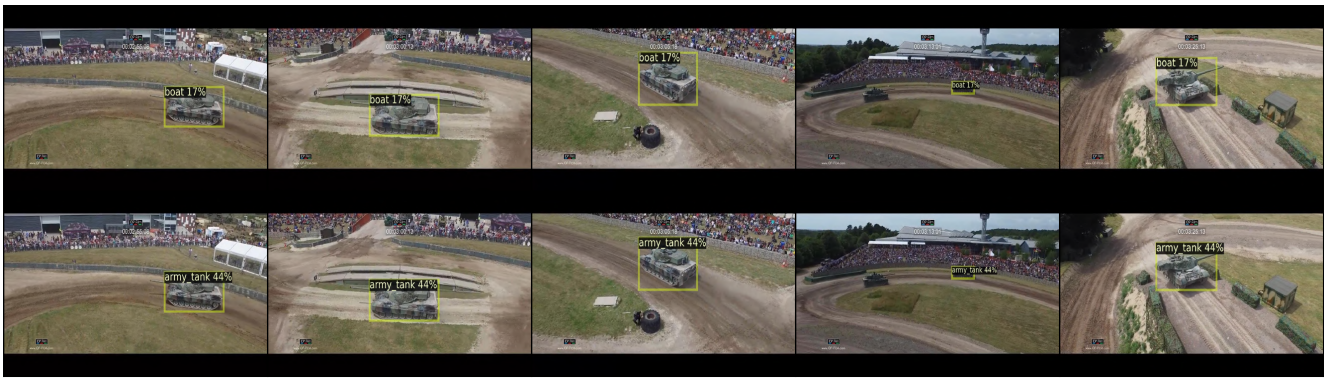
Plug-and-play of our method. To support our claim that the set classifier is plug-and-playable, we show additional results in Tab. 11. As QDTrack [35] is the only open-sourced tracker that targets TAO [9], we validate other trackers using YouTube-VIS 2019 [54]. Specifically, we further evaluate MaskTrack R-CNN [54] and CrossVIS [55] by adding the set classifier on top of each. CrossVIS is based on the one-stage detector FCOS [44], so we make FCOS take the role of RPN by substituting region proposals with predicted boxes from FCOS. As shown in Tab. 11, all methods gain noticeable accuracy improvements with our set classifier. CrossVIS with the set classifier surpasses the offline method, VisTR (38.6 AP).

Number of sampling frames. In Table 6, we discuss the importance of training with videos in order to cover the true appearance changes. In Table 12, we then analyze the impact from the number of frames retrieved from a video per batch. We find that utilizing multiple frames greatly outperforms the use of a single frame as the set classifier can be trained from tracklets having real-video characteristics. With the proposed augmentations, we observe that using multiple distinct frames brings competitive accuracy from having real-video aspects.

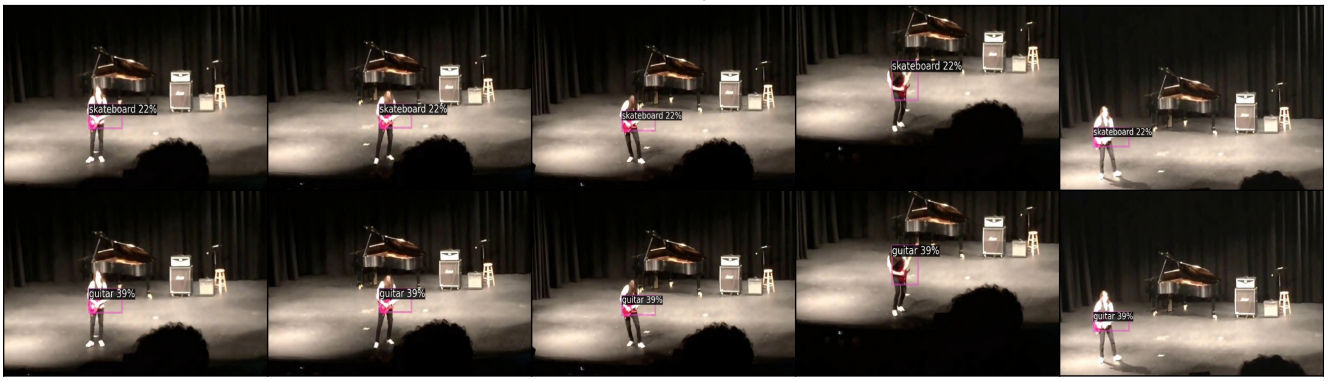
Number of transformer encoder layers. In Table 13, we show that the accuracy of the set classifier increases with respect to the number of transformer encoder layers. Because we find that the number of encoder layers more than 3 does not bring much improvements, we set $N_E = 3$ to be our default setting.

D. Submission Questions Response

We believe the studies on long-tail object tracking should be aware of potential violations of personal privacy. Our proposed method is specialized to offline inference settings. Therefore, for future work, we plan to devise a model that is feasible for online inferencing while being accurate and robust within the large vocabulary. The licenses of used data [9, 15, 54] are CC BY-NC-SA 3.0, CC 4.0, and CC 4.0 respectively.



boat → army tank



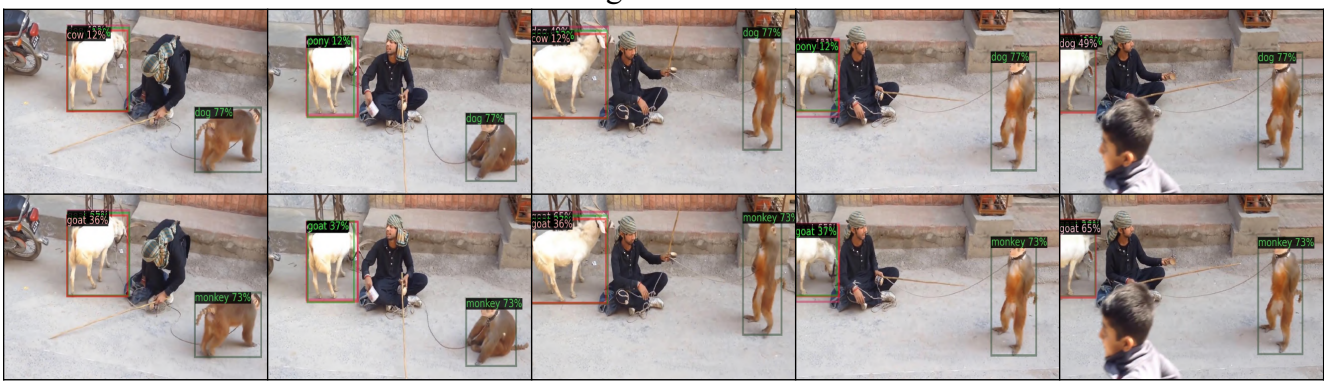
skateboard → guitar

kite → parachute

motorcycle → dirt bike

Figure 6. Visualization of predictions from QDTrack (top) and our model (bottom).

log → lizard

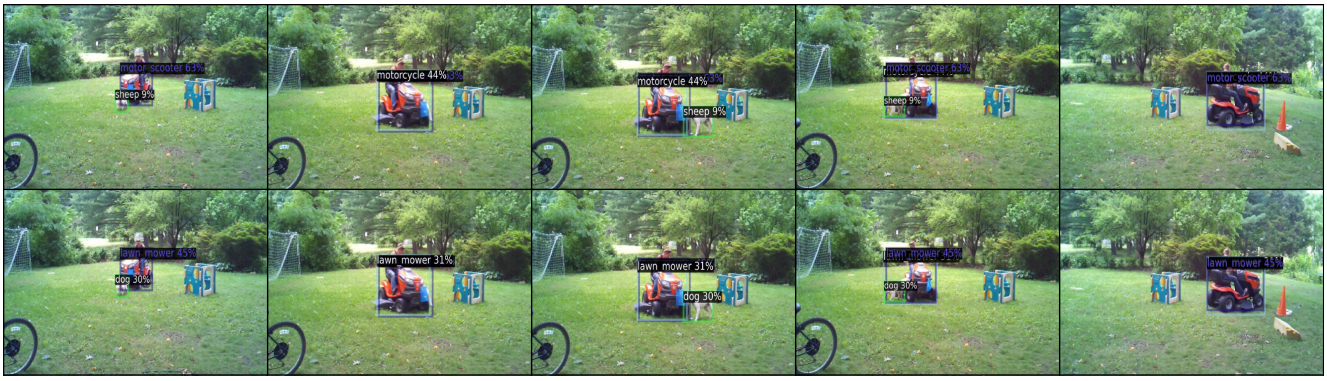


dog, cow, pony → goat / dog → monkey

cow → bull

surfboard → kayak

Figure 7. Visualization of predictions from QDTrack (top) and our model (bottom).



motorcycle, motor scooter → lawn mower / sheep → dog

boat → tarp

basket → birdcage

toy → vacuum cleaner / toy → carton

Figure 8. Visualization of predictions from QDTrack (top) and our model (bottom).