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

Sukjun Hwang[1] Miran Heo[1] Seoung Wug Oh[2] Seon Joo Kim[1]
[1]Yonsei University [2]Adobe Research
{sj.hwang, miran, seonjookim}@yonsei.ac.kr seoh@adobe.com

Abstract

Recently, both long-tailed recognition and object tracking have made great advances individually. TAO benchmark presented a mixture of the two, long-tailed object tracking, in order to further reflect the aspect of the real-world. To date, existing solutions have adopted detectors showing robustness in long-tailed distributions, which derive per-frame results. Then, they used tracking algorithms that combine the temporally independent detections to finalize tracklets. However, as the approaches did not take temporal changes in scenes into account, inconsistent classification results in videos led to low overall performance. In this paper, we present a set classifier that improves accuracy of classifying tracklets by aggregating information from multiple viewpoints contained in a tracklet. To cope with sparse annotations in videos, we further propose augmentation of tracklets that can maximize data efficiency. The set classifier is plug-and-playable to existing object trackers, and highly improves the performance of long-tailed object tracking. By simply attaching our method to QDTrack on top of ResNet-101, we achieve the new state-of-the-art, 19.9% and 15.7% TrackAP$_{50}$ on TAO validation and test sets, respectively. Our code is available at this link[1].

1. Introduction

Object tracking is a long standing problem in computer vision as it plays a key role in surveillance and self-driving applications. There are numerous datasets and benchmarks for tracking [1, 14, 23, 31, 47] and also a long list of tracking algorithms [3, 4, 41, 51, 58]. As with many other computer vision tasks, the performance of tracking algorithms has also taken a leap with deep learning.

Even with the great progress in object tracking, the performance of state-of-the-art trackers starts to degrade in the real-world scenarios with a large vocabulary of objects [9]. This is because most tracking benchmarks include only a small set of objects such as pedestrian, vehicles, and animals, for targeting specific applications like autonomous driving. To deploy the trackers in the real-world in a general environment, it is essential for the trackers to be able to deal with a much larger set of objects as in the image detection problem [15]. For this purpose, a new benchmark called TAO [9] for tracking any object has been recently introduced. This dataset contains over 800 categories, an order of magnitude more than previous tracking benchmarks.

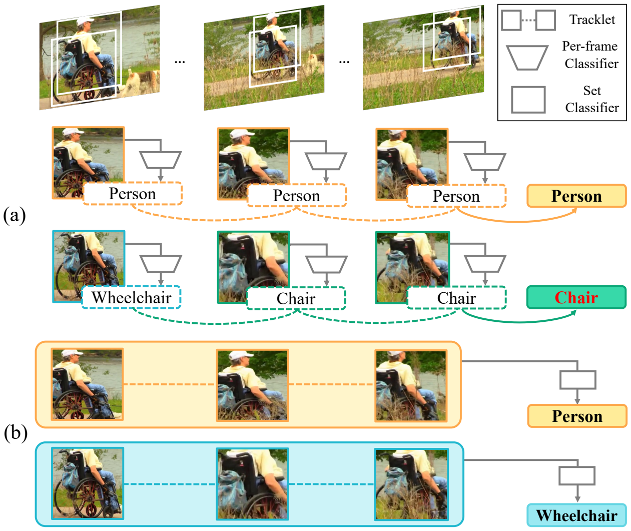


Figure 1. (a) *Per-frame classifier* receiving an instantaneous scene struggles on tail categories (e.g., wheelchair). On the other hand, (b) our proposed *set classifier* shows robustness on tail categories by aggregating multiple viewpoints of a tracklet, taking the whole spatio-temporal feature into account.

In [9], it was shown that most up-to-date trackers do not adapt well with increased number of object vocabulary. While tracking algorithms have focused on accurately finding object boxes and tracking them, less attention has been paid on the classification of objects, primarily due to a small set of vocabulary. As the object category grows to a realistic size, classification becomes crucial for the overall tracking performance. After a thorough analysis, it was suggested in [9] that *"large-vocabulary tracking requires jointly improving tracking and classification accuracy"*.

[1] https://github.com/sukjunhwang/set_classifier

In this paper, we show that aggregating multiple viewpoints of a tracklet is the key to classifying the large vocabulary in videos. A tracklet refers to a set of boxes in different frames that share the same identity. Although appearance of objects in tracklets may go through great changes, existing methods [3, 9, 35] determine the category of a tracklet from the collection of per-frame classification results as shown in Fig. 1 (a). Since changes in scenes from temporal variations are not considered, they are vulnerable to appearance changes including motion blur or occlusion. More importantly such cases bring critical deterioration of performance in tail classes. Specifically, detectors trained on imbalanced data are more confident in frequent classes, and such cases bring critical deterioration of performance in tail classes.

To this end, we propose a *set classifier* that takes the spatio-temporal features of a whole tracklet into account (Fig. 1 (b)). With this design, the set classifier is supplied with sufficient information to determine a category from the large vocabulary. Therefore, the set classifier gains robustness against temporal shifts and the ability to avoid a collapse of final predictions from transient failures, leading to noticeable improvements of accuracy in the tail.

To fully supervise the set classifier to obtain the ability of exploiting spatio-temporal information, the module gets trained with video data. In contrast to existing methods that can only classify the large vocabulary using framewise detections [3, 9, 35], the essence of the set classifier is the ability to evaluate a whole tracklet by aggregating information from multiple sources. The structural design of the set classifier is simple yet powerful; it is a stack of a few transformer layers [46]. Receiving multiple regional features [16] corresponding to the predicted boxes that compose a tracklet, the set classifier attends to relevant information that are necessary for the classification of the large vocabulary.

However, due to immense efforts required to annotate a video [9], the annotation budget is insufficient to give the supervision of classifying tracklets under the complicated long-tailed scenarios. As a solution to this dilemma, we present augmentation methods of generating tracklets that have video characteristics: a variety of viewpoints of an object. Specifically, tracklets are dynamically generated utilizing regional proposals [38] from multiple source videos and images. To further make the most out of the limited number of annotations, the augmented tracklets can be composed of multiple identities, and we introduce a training procedure for the set classifier using such tracklets. With our methods, an enormous number of tracklet samples composed of rare classes can be obtained, and the set classifier gains the ability to successfully distinguish the large vocabulary in videos.

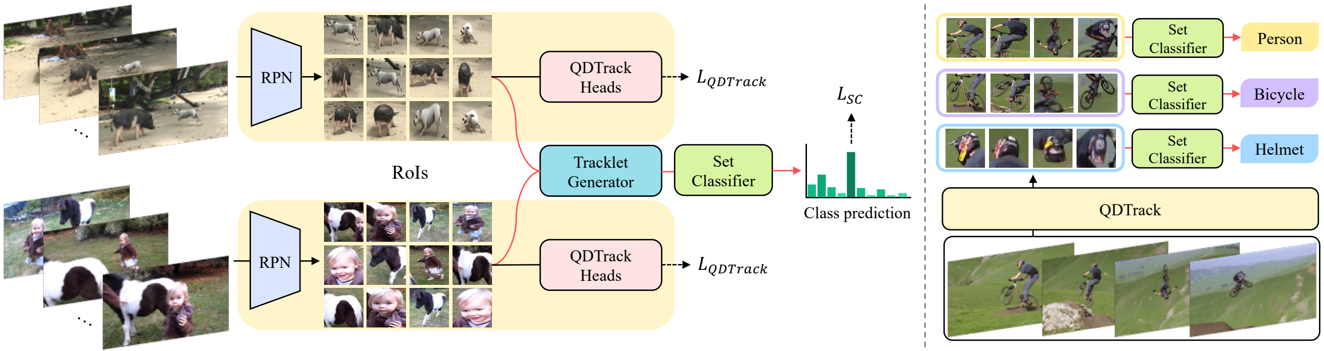
Adoption of our set classifier results in high performance improvement in the long-tailed tracking. With the plug-and-playable design, we show experimental results on top of recently proposed QDTrack [35], and achieve new state-of-the-art on the challenging TAO [9] benchmark: 19.9% and 15.7% TrackAP$_{50}$ on validation and test sets, respectively. Furthermore, taking the same approach, we also achieve a competitive result of 37.7% AP on the video instance segmentation dataset, YouTube-VIS 2019 [54].

Our work can be summarized as follows:

- We propose the set classifier which classifies a tracklet as a whole by aggregating information from multiple viewpoints.
- We introduce augmentation methods that can generate augmented tracklets of near infinite diversities – unlimited number of tracklets of tail classes can be obtained.
- We propose a new training procedure that facilitates the supervision of the set classifier using the augmented tracklets. Moreover, we suggest auxiliary losses that bring further improvements in accuracy.
- We achieve new state-of-the-art on TAO, and also show the effectiveness of our method on YouTube-VIS 2019.

2. Related works

Long-Tailed Recognition. While considerable progress in visual perception has been made on class-balanced datasets [7, 26, 39], naïve migration of cutting-edge models to long-tailed datasets [15, 28, 29] shows severe failures. With growing interests in inherent problems of long-tailed distributions, there have been noticeable explorations to the prevention of minor classes getting dominated by major classes. Such solutions can be largely categorized into three: re-balancing distributions, re-weighting losses, and decoupled training. Re-balancing methods [6, 15, 18, 21] tackle the sparsity of annotations in tail classes, which is the fundamental cause of long-tailed distributions. The methods have shown that adequately sampling minor classes with higher rates can supplement the huge imbalance, showing improvements in the tail. Instead of resolving the imbalance by weighting the sampling rate, the problem in long-tailed distributions can also be alleviated by differently weighting class-wise losses [8, 42, 43, 48]. The weights for different classes are determined with considerations to the number of annotations, mostly putting emphasis on tail classes. BAGS [25] and SimCal [49] introduced attaching additional heads that are specialized in classifying tail classes, and combining the results from various heads during the inference. Having categories grouped with a similar number of annotations, BAGS assigns different heads to each group, which prevents minor classes from getting dominated by major classes. Similarly, the additional head of SimCal avoids the domination by receiving class-balanced proposals.



(a) Training

(b) Inference

Figure 2. Overview of our framework. The set classifier is plug-and-playable to existing object trackers. (a) During training, the set classifier gets trained with QDTrack simultaneously by receiving tracklets from the tracklet generator. (b) For the inference, the object tracker first predicts tracklets. Then, the set classifier takes a tracklet as a whole by fetching RoI tokens that correspond to boxes of the tracklet. Finally, the proposed module comes out with re-classified predictions for each tracklet.

Understanding Objects in Videos. From the huge success in images, there have been many tasks that shift the focus onto videos [1, 23, 31, 36, 47, 54]. The fundamental characteristic of videos is that they are composites of a number of consecutive frames. Therefore, well utilizing information and predictions from multiple frames has become the de facto factor in improving performance.

To exploit spatial proximity between consecutive frames, many approaches that target tracking tasks associate objects by fusing the motion prior with additional algorithms; optical flow [52, 60], displacement regression [12, 58], and Kalman Filter [3, 20]. However, relying heavily on the motion priors shows vulnerability to scenes with low frame rates or large camera motions. To overcome those problems and handle reappearances, use of implicit features that represent objects can be considered [4,13,30,33,35,41,54].

It has been shown that utilizing information from multiple frames is beneficial not only for tracking, but also for improving the quality of detection and segmentation in videos [2, 19, 22, 34, 50]. For example, in the video instance segmentation task [54], per-clip methods [2, 19, 22, 50] generally show higher segmentation accuracy than approaches using only single frame information [5, 27, 54, 55].

Videos with Large Vocabulary. As mentioned earlier, most previous video-related studies did not pay much attention to improving the classification performance. Rather, major improvement came from tracking [1, 23, 31], detection [39], and segmentation [36, 37, 47, 54]. The primary reason for this trend is that most video related benchmarks include only a few major object classes.

Recently proposed benchmark TAO [9] is a video dataset with large vocabulary. Since long-tailed scenarios were usually dealt only with images, existing methods [3, 9, 35] showed difficulties in classifying tracklets of rare categories in videos. In this paper, we introduce a set classifier which re-evaluates the category of tracklet predictions given from the recent object tracker QDTrack [35]. The plug-and-playable set classifier aggregates information from various viewpoints of an object in a similar way to [24], which is necessary for classifying large vocabulary. By increasing the classification accuracy of the predicted tracklets, noticeable improvements in overall accuracy are achieved in TAO and YouTube-VIS [9, 54].

3. Method

Our method can be easily implemented on top of existing object trackers that use two-stage detectors. In this paper, our framework is built upon the recently proposed QDTrack [35] with integration of the additional head that can be jointly trained: the set classifier (Fig. 2). The input video is first fetched by the object tracker that detects objects per-frame and generates tracklets by associating the predictions. Next, the newly proposed set classifier receives the tracklet as a whole, where each item in a tracklet is a RoI-Aligned [16] feature that corresponds to a predicted box. Finally, with the design of incorporating multiple viewpoints of instances, the set classifier head precisely predicts the category of the given tracklet.

Throughout this paper, k^{th} regional proposal from RPN [38] is defined as $B_k = (b_k, c_k, i_k)$, where b_k denotes the box coordinates. Here, c_k and i_k denote the category and the identity of the ground-truth box that b_k is matched to.

3.1. Set Classifier

Typical trackers [35, 54, 58], which have been specialized to major classes (e.g., car and pedestrian), have not given much consideration to the classification. Therefore, they usually use class predictions from detectors [16,38,59] directly. However, failure of such naïve classification is witnessed when encountering the object tracking benchmark with long-tailed distribution [9]. The most challenging aspect of classification in such a scenario is the change of appearance over time. To overcome this problem, we propose a classifier that aggregates information of multiple views from a tracklet. This simple extension strengthens the tracklet-level classification accuracy by mitigating the inconsistency of most trackers; deriving class prediction from naïve averaging or max-count of per-RoI classification results.

We design our set classifier as a stack of N_E transformer encoder layers [46] (Fig. 3). Inputs for the set classifier come from two sources: a classification token and RoI tokens from a tracklet. The classification token is trainable, similar to recent uses of tokens in transformers [10, 11, 45]. Each RoI feature corresponding to box labels in a tracklet passes through an extra lightweight embedding head to become a RoI token. By inserting the classification token x_0 and RoI tokens $\{x_l\}_{l=1}^{L}$ together into the set classifier, the classification token encodes the overall contextual information of the given tracklet and the set classifier outputs embeddings $\{z_l\}_{l=0}^{L}$.

Using the output embedding of the classification token z_0, logits $\hat{y} \in \mathbb{R}^C$ are predicted, where C is the number of categories of the dataset. With the prediction, loss can be calculated using Cross-Entropy (CE) Loss as

$$L_{SC}(y, \hat{y}) = -\sum_{c=1}^{C} y_c \log(\sigma_c), \qquad \sigma_c = \frac{e^{\hat{y}_c}}{\sum_{k=1}^{C} e^{\hat{y}_k}}, \tag{1}$$

where $y_c \in \{0, 1\}$, $1 \leq c \leq C$ is the one-hot ground truth label.

To train the set transformer, we first extract tracklets from training video clips as done in QDTrack [35]. Then the set transformer learns to predict one most probable class for each tracklet. This approach has several advantages over the conventional method that predicts class per frame (or RoI). It becomes more robust against noise such as motion blur and occlusion, and accurate on classifying the tail classes by aggregating information from multiple sources.

However, due to the lack of tracklet annotations in video, using the aforementioned training pipeline alone inevitably encounters over-fitting. Even using the largest scale long-tail video detection dataset [9], we observed that the number of training tracklets is not sufficient to prevent over-fitting, especially for the tail classes. It is unavoidable as annotating for videos requires much more effort and resources than for images. To alleviate this issue, we propose effective data augmentation methods that can generate tracklets with video characteristics. Our augmentation methods can not only increase the number of existing video tracklets, but also synthesize video tracklets from images.

3.2. Tracklet Augmentations

In this section, we describe data augmentation methods for generating tracklets to resolve the data shortage issue (Fig. 4 (a)). The data augmentation is designed with following considerations that are essential for effective supervision of the set classifier.

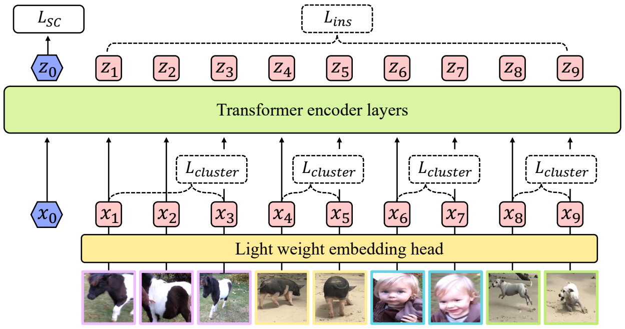


Figure 3. The structural design of the set classifier. L_{SC} is calculated from the prediction of the classification token, which embeds the overall context of a tracklet. The use of two auxiliary losses, L_{ins} and $L_{cluster}$, brings further improvements in accuracy.

Tracklets from region proposals. Instead of using only the ground-truth box labels, we combine numerous regional proposals of RPN in multiple frames to generate tracklets with new views (Fig. 4 (b)). Compared to sparse box annotations, numerous region proposals can be collected where each has a different perspective to the matched ground-truth box. Therefore, this augmentation leads to quantitatively increasing and further diversifying the training samples for the set classifier. Moreover, the augmentation is applicable not only to videos [9], but also to images [15]; tracklets can be generated from images. Even if components originated from a single image, the combinations of possible object regions can imitate appearance transitions that are inherent in videos. With the augmentation, the set classifier can learn to aggregate information from different viewpoints, which is necessary for classifying large vocabulary.

More diverse tracklets by mixing RoIs. We observed that the proposed tracklet augmentation significantly impacts the classification performance. However, it is worth noting that RPN proposes more RoIs on head classes than tail classes, which intensifies the imbalance between the head and the tail. If tracklets are generated by gathering RoIs matched to a single identity, it inevitably leads to tail classes having fewer and less diverse tracklets than that of head classes. To diversify the tracklets and expose more tail classes, we propose to further augment tracklets to have multi-identity and multi-class tracklets (Fig. 4 (c)). Inspired by augmentation techniques that mix two images [56, 57], we extend the idea to our augmentation. In other words, we do not restrict the generated tracklets to be composed of a single class. With this approach, the number of combinations of forming tracklets is countless by mixing all the RoIs within a training batch. This makes our set classifier even more robust to over-fitting.

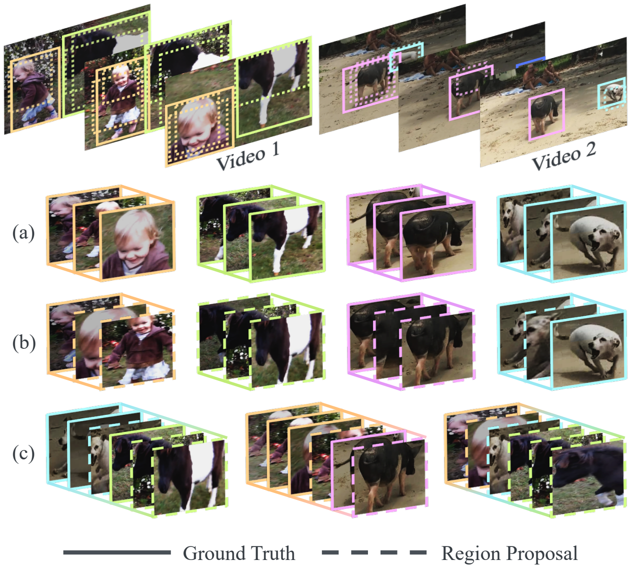


Figure 4. Augmentations of tracklets. The solid lines denote ground-truth box annotations, and the dotted lines are region proposals from RPN. (a) Because annotating a video requires huge efforts, video datasets have sparse labels that are insufficient to train large vocabularies. (b) Tracklets can be diversified from substitution of box labels with region proposals. (c) Further supervision to tail classes can be derived from mixtures of tracklets, having multiple classes.

Resampling for tail classes. Our RoI mix-up method also enables unlimited resampling of sparse tail class RoIs. Recent findings [15,25,32,40] have shown the impact of training data sampling to be critical on datasets showing long-tailed distributions. Likewise, we observed that the strategy of sampling RoIs in generated tracklets also plays an important role. Inspired by RFS [15], the probability of sampling an RoI is determined by the total number of training samples per category in the training set. Let n_c be the total number of training annotations in the training dataset that are labeled to category c. We sample RoIs under the multinomial distribution where p'_k, the probability of sampling B_k, is defined as follows:

$$p'_k = \frac{p_k}{\sum_{j=1}^{B} p_j}, \qquad p_k = \sqrt{\frac{1}{n_{c_k}}}. \tag{2}$$

Compared to uniformly sampling RoIs, the weighted sampling strategy results in the set classifier encountering more tracklets containing RoIs of tail classes.

Training with Augmented Tracklets. Using the proposed augmentation methods, let $X = \{B_l\}_{l=1}^{L}$ denote a generated tracklet, where L is the number of items. As mentioned above, the set X can be composed of RoIs that contain numerous different categories. For a dataset of C categories, we define the label y to be the proportion of categories of items within the set X as follows:

$$y = \left\{ \frac{\sum_{l=1}^{L} \mathbb{1}_{\{c_l = c\}}}{L} \right\}_{c=1}^{C} \in [0,1]^C, \quad \sum_{c=1}^{C} y_c = 1. \tag{3}$$

For the training with the augmented tracklets, typical CE Loss (Eq. (1)) can be used with the simple replacement of the one-hot ground-truth label to the multi-class soft label.

3.3. Auxiliary Tasks

In addition to the soft label of augmented tracklets, we can also utilize instance-level labels that correspond to region proposals. We further introduce two auxiliary losses, each of which improves the accuracy of the set classifier.

Instance-wise loss. The structural design of the set classifier is similar to the transformer based ViT [11]; it is composed of a stack of transformer encoder layers and uses a classification token for the final prediction (Fig. 3). Since ViT targets the image classification task, where local patches may not hold meaningful information, only the classification token can receive losses. Contrary to ViT, the set classifier receives tokens that are embedded from object-like regional proposals. Therefore, other tokens, not just the classification token, can be used to calculate the instance-wise loss L_{ins}. Instance-wise loss adopts simple CE Loss to predict the class of the originated region, which accelerates the training of the set classifier.

Clustering loss. A key challenge in training the set classifier with the augmented tracklets is that the tokens originated from various identities. For example, in Fig. 3, token x_1 should find relevance from x_2 and x_3 and aggregate information from the two. However, if x_1 wrongly interprets that x_4 and x_5 also originated from the same instance, the accuracy of the set classifier will decrease. In order to clarify the sources of tokens, supervising the tokens to embed object appearances can be considered following many approaches that tackle tracking tasks [35, 54]. Nevertheless, since tracking and classification are distinct to another, use of such supervision rather harms the accuracy.

From the assumption that class distributions of RoIs that share the same origination should be similar, we lightly cluster the feature representations by utilizing the class distributions. Using the class logits $\{\tilde{y}\}_{l=1}^{L}$ predicted by inserting the RoI tokens $\{x_l\}_{l=1}^{L}$ to a linear classifier, the features can be clustered as follows:

$$L_{cluster}(y_l, \tilde{y}_l) = L_{CE}(y_l, \tilde{y}_l) + KL(\tilde{p}_l \| Q_l), \tag{4}$$

$$Q_l = \frac{\sum_{k=1}^{L} \mathbb{1}_{\{i_k = i_l\}} \tilde{p}_k}{\sum_{k=1}^{L} \mathbb{1}_{\{i_k = i_l\}}}, \tag{5}$$

where y_l is the ground-truth category label of l^{th} RoI, and $\tilde{p}_l$ is a class probability distribution obtained from $\tilde{y}_l$ by the

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP_r	AP_c	AP_f
QDTrack [35]	17.2	29.1	17.4	5.7	13.1	22.0	6.5	11.9	25.9
Ours	18.3	29.5	18.9	6.7	11.9	23.7	13.6	14.0	23.8

Table 1. Results of detection metrics on TAO validation set. Following LVIS, TAO reports AP_r (*rare* categories with 1 to 9 videos), AP_c (*commmon* categories with 10 to 99 videos), and AP_f (*frequent* categories with $\geq$ 100 videos). AP_s, AP_m, and AP_l denote accuracies on boxes of *small*, *medium*, and *large* sizes, respectively.

Method	TAO validation			TAO test		
	$TrackAP_{50}$	$TrackAP_{75}$	$TrackAP_{50:95}$	$TrackAP_{50}$	$TrackAP_{75}$	$TrackAP_{50:95}$
SORT [3, 9]	13.2	-	-	10.2	4.4	4.9
QDTrack [35]	15.8	6.4	7.3	12.4	4.5	5.2
Ours	19.9	8.3	9.6	15.7	6.8	7.4

Table 2. Results of tracking metrics on TAO validation and test sets. Our method outperforms previous methods by a meaningful margin.

Method	AP	AP_{50}	AP_{75}	AR_1	AR_{10}
MaskTrack R-CNN [54]	31.9	53.7	32.3	32.5	37.7
SipMask† [5]	33.7	54.1	35.8	35.4	40.1
CrossVIS [55]	36.6	57.3	39.7	36.0	42.0
VisTR [50]	38.6	61.3	42.3	37.6	44.2
QDTrack [35]	34.4	55.1	38.4	33.5	41.6
Ours	37.7	60.4	39.8	35.6	45.8

Table 3. Comparison of results on YouTube-VIS 2019 with previous methods using ResNet-101. † indicates using ResNet-50.

softmax function. The use of KL divergence brings the class distributions to resemble the centroid distribution Q. With this auxiliary loss, the inputs to the set classifier share similar embeddings if originated from the same object, which assists the set classifier to aggregate relevant information.

4. Experiments

In this section, we evaluate the proposed method using TAO [9] and YouTube-VIS 2019 [54]. We demonstrate that our method achieves a huge increase in the overall performance by improving the classification performance, especially for infrequent object categories in the tail. More experiments and details can be found in the supplementary.

4.1. Datasets

TAO. Our experiments are conducted on the long-tailed object tracking benchmark, TAO [9]. TAO is a large-scale dataset that has long-tailed distributions within 482 classes that are a subset of LVIS [15]. The dataset has 500 videos, 216 classes in the training set, 988 videos, 302 classes in the validation set, and 1419 videos, 369 classes in the test set. It is worth noting that the categories in the validation set and the test set are not subsets of the training set, so categories do not overlap. The ability to classify such non-overlapping categories should be trained from LVIS [15].

YouTube-VIS 2019. Most MOT benchmarks have a very limited number of classes: less than ten. Therefore, we further demonstrate the importance of tracklet classification on the video instance segmentation benchmark, YouTube-VIS 2019 [54]. The dataset is composed of 40 categories, and is composed of 2,238/302/343 videos for training/validation/test.

4.2. Implementation Details

All models including previous works and ours use ResNet-101 [17] as the backbone. By default, the number of transformer encoder layers composing the set classifier is $N_E = 3$, where each layer has 8 heads with overall depth of 512. For the inputs of the set classifier, we first flatten region proposals and embed corresponding features by two fully connected layers [38]. Thanks to the plug-and-playable design of the proposed set classifier, we attach the module on top of QDTrack [35]. Our model is fully trained end-to-end with [35] and the configurations are identical unless specified.

The training schedule of QDTrack has two phases; 24 epochs of pretraining on LVIS [15], and 12 epochs of finetuning on TAO [9]. Although the set classifier can be trained only from images, the accuracy higly improves when using real video samples together (Table 6). Therefore, while following the training schedule of [35], we simultaneously insert videos of TAO during the pretraining phase. For each iteration of training, the tracklet generator produces 256 augmented tracklets of varying length from 16 to 32. The weight of L_{SC}, L_{ins}, and $L_{cluster}$ are 0.05, 0.02, 0.1 respectively. During inference, we follow the pipeline of generating tracklets and re-classifying the category as mentioned in Sec. 3. Denoting the predicted classification score from the set classifier as c, and the confidence score of the tracklet from [35] as s, the output score can be obtained as $c^{\lambda_c} s^{\lambda_s}$, where $\lambda_c = \frac{1}{3}, \lambda_s = \frac{2}{3}$ by default. We finalize the output score by multiplying the length of each to penalize redundant short tracklets.

4.3. Main Results

As mentioned in Sec. 3, we receive tracklet predictions from the tracker, and assign a newly predicted category from the set classifier to each box that forms tracklets. Therefore, the only difference between the outputs of QDTrack [35] and ours is the classification labels. Surprisingly, with the simple re-classification of tracklets, our method highly improves the overall accuracy of QDTrack on both TAO and YouTube-VIS benchmarks.

TAO. The results on TAO can be largely divided into two as shown in Table 1 and Table 2, where each refers to the accuracies on detection and tracking, respectively. Since our method is built on top of QDTrack [35], the quality of predicted box coordinates is similar to that of [35]. However, from the reclassification of the boxes with the set classifier, our model outperforms QDTrack with a noticeable gap of 1.1% detection AP as shown in Table 1. More importantly, the score of AP_r is more than doubled from that of QDTrack. AP_r is an important criterion in benchmarks of long-tailed distributions [9, 15] as it represents the ability of classifying rare categories. From the noticeably enhanced classification results, our model achieves the best accuracy of 19.9% on the tracking metric TrackAP_{50}, which is a 4.1% of improvement compared to QDTrack. These results signify that from the utilization of multiple information in tracklets, the set classifier successfully classifies large vocabularies.

YouTube-VIS 2019. The results on YouTube-VIS 2019 benchmark is shown in Table 3. Since QDTrack [35] is an object tracker, it is not capable of generating segmentation masks by default. Therefore, we attach the simple mask head of Mask R-CNN [16] on top of QDTrack. From the small modification, QDTrack shows the accuracy of 34.4% AP, which is comparable to previous methods that mainly target the video instance segmentation task.

We also demonstrate the effectiveness of our method, re-classifying tracklets using the set classifier, on the VIS benchmark. Because the VIS dataset has relatively small vocabulary size (40 categories), an insignificant impact is expected from the use of the set classifier. Nevertheless, the predicted class labels from our set classifier greatly improves the accuracy of QDTrack by 3.3% AP. The achieved score of 37.7% AP is higher than many existing VIS models [5, 54, 55], and also competitive to recently proposed VisTR [50].

4.4. Ablation Studies

Using the validation set of TAO, we show various ablation studies on the set classifier and the tracklet generator. We show how the accuracy of the set classifier can be improved by differentiating tracklet augmentations and training configurations.

Multi-Identity	Multi-Class	TrackAP_{50}	TrackAP_{75}	$\text{TrackAP}_{50:95}$
		17.7	7.4	8.5
✓		17.2	7.9	8.5
✓	✓	**19.9**	**8.3**	**9.6**

Table 4. Comparison of different tracklet augmentations. *Multi-Identity* denotes that tracklets can be composed of regional proposals from different identities. *Multi-Class* denotes that tracklets can be mixtures of multiple classes.

#tracklets / batch	TrackAP_{50}	TrackAP_{75}	$\text{TrackAP}_{50:95}$
32	17.7	8.2	9.1
64	18.3	7.6	8.9
128	18.5	8.2	8.9
256	**19.9**	**8.3**	**9.6**

Table 5. Comparison of number of generated tracklets per batch.

Multi-class tracklet generation. Here, we study the impact of the multi-class tracklet augmentation (Table 4). Due to the sparse number of annotated boxes, using only ground-truth box labels directly leads to severe overfitting. Therefore, we proposed various augmentations that can enormously diversify the tracklets in Sec. 3.2. The augmentation of tracklets using regional proposals, but not allowing them to be composed of multiple identities and multiple classes (Fig. 4 (b)) leads to the accuracy of 17.7% TrackAP_{50}.

Mixing RoIs of different identities (Fig. 4 (c)) can be divided into two, depending on whether composing tracklets with RoIs of multiple classes is granted or not. As shown in Table 4, we find the allowance of multiple classes improves the accuracy by 2.2% TrackAP_{50}, while the preclusion of multiple classes rather decreases TrackAP_{50}. It is because the composition of multiple classes is an important factor to diversify augmented tracklets, especially when dealing with large vocabulary scenarios where only a few regional proposals can be gathered in the tail.

Number of augmented tracklets. Thanks to the multi-class tracklet augmentation, we can now generate training samples for the set classifier without limitations. Therefore, the number of tracklets can be as many as the set classifier requires for training. As shown in Table 5, by increasing the number from 32 to 256, TrackAP_{50} increases by 2.2. This experiment indicates that training the set classifier with more number of tracklets brings further supervision, leading to higher accuracies.

Importance of video training. The main purpose of the set classifier is to aggregate information from multiple viewpoints of an object. Though the ability can be partially gained from augmented tracklets that are generated from images, such tracklets cannot serve the true appearance changes in videos. However, as mentioned earlier, using only the sparse labels from videos leads to an overfitting

Images	Videos	TrackAP$_{50}$	TrackAP$_{75}$	TrackAP$_{50:95}$
✓		17.1	6.9	8.4
	✓	6.5	2.6	2.8
✓	✓	**19.9**	**8.3**	**9.6**

Table 6. Comparison of different domains used for training the set classifier. *Images* and *Videos* denote the use of LVIS [15] and TAO [9], respectively.

Tracklet Length	TrackAP$_{50}$	TrackAP$_{75}$	TrackAP$_{50:95}$
$[\ 8, 16)$	19.0	7.8	8.7
$[16, 32)$	**19.9**	**8.3**	**9.6**
$[32, 64)$	18.3	7.0	8.3

Table 7. Comparison of different lengths of generated tracklets.

(6.5% TrackAP$_{50}$). Therefore, we supply videos of TAO to generate more meaningful tracklets during the pretraining phase. As shown in Table 6, a noticeable improvement of 2.8% TrackAP$_{50}$ is gained by using the augmented tracklets from videos.

Length of augmented tracklets. In Table 7, we investigate how the length of augmented tracklets used during training affect the final accuracy. In order to robustly classify the outputs of the tracker with varying lengths, we provide augmented tracklets of different lengths during training. On the TAO validation dataset, the total average length of tracklets predicted by QDTrack is 21.24. As can be expected from the average length, using tracklets ranging from 16 to 32 for training turns out to bring the best performance.

Sampling ratio of tail categories. Sampling images containing tail classes more frequently similar to RFS [15], we regulate the generation of tracklets to favor including tail classes over that of head classes (Sec. 3.2). The regulation can be controlled by differentiating the probability of sampling RoIs that are used in the tracklet augmentation. The probability is defined using the total number of training annotations per class n_c as denoted in Eq. (2). The results of controlling the probability is shown in Table 8. If $p = 0$, all RoIs are given the same probability to be sampled, which indicates uniform sampling. Since uniform sampling does not consider a huge imbalance of annotations between different categories, it shows a large reduction of accuracy. Among many p values, 0.5 turns out to be the best hyperparameter that well alleviates the imbalance and achieves the highest performance.

Use of auxiliary losses. We study the effects of the introduced auxiliary losses: L_{ins} and $L_{cluster}$. As described in Sec. 3.3, the two losses assist the training of the set classifier. From the baseline which does not use the two losses, the use of L_{ins} brings an increment of 0.2% TrackAP$_{50}$. Further improvement of 1.0% TrackAP$_{50}$ is gained by additionally adopting L_{ins}, which assists the set classifier to easily aggregate relevant information.

p	TrackAP$_{50}$	TrackAP$_{75}$	TrackAP$_{50:95}$
uniform	15.6	7.3	7.8
0.25	17.3	7.2	8.4
0.5	**19.9**	**8.3**	**9.6**
0.75	17.8	7.1	8.3
1.0	17.5	7.5	8.5

Table 8. Comparison of assigning different probabilities used by the multinomial sampling; n_c^{-p} in Sec. 3.2. As p increases, the sampling policy favors RoIs of tail classes.

Instance	Cluster	TrackAP$_{50}$	TrackAP$_{75}$	TrackAP$_{50:95}$
		18.7	7.8	8.8
✓		18.9	8.0	9.0
✓	✓	**19.9**	**8.3**	**9.6**

Table 9. Comparison of using the auxiliary losses. *Instance* and *Cluster* denote the use of L_{ins} and $L_{cluster}$, respectively.

5. Conclusions

In this paper, we demonstrated that classification is a key factor in tracking performance on benchmarks with large vocabulary, and introduced the set classifier that takes the whole spatio-temporal features of a tracklet. The set classifier precisely classifies large vocabularies by aggregating information from multiple viewpoints. In order to bring out the potential of the set classifier, we also proposed tracklet augmentations that greatly diversify sparse annotations. Furthermore, we strengthen the supervision with the suggested auxiliary losses. The set classifier achieves the new state-of-the-art accuracy on the challenging benchmark TAO, and also shows competitive results on YouTube-VIS 2019. For future work, we plan to design a video-targeted classifier that can precisely classify large vocabulary while being capable of online inferencing.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT), Artificial Intelligence Innovation Hub under Grant 2021-0-02068, Artificial Intelligence Graduate School Program under Grant 2020-0-01361, and Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis under Grant 2014-3-00123.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 1, 3

[2] Gedas Bertasius and Lorenzo Torresani. Classifying, segmenting, and tracking object instances in video with mask propagation. In *CVPR*, 2020. 3

[3] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *ICIP*, 2016. 1, 2, 3, 6

[4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *CVPR*, 2020. 1, 3

[5] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*, 2020. 3, 6, 7

[6] Nadine Chang, Zhiding Yu, Yu-Xiong Wang, Anima Anandkumar, Sanja Fidler, and Jose M Alvarez. Image-level or object-level? a tale of two resampling strategies for long-tailed detection. *arXiv preprint arXiv:2104.05702*, 2021. 2

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2

[8] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 2

[9] Achal Dave, Tarasha Khurana, Pavel Tokmakov, Cordelia Schmid, and Deva Ramanan. Tao: A large-scale benchmark for tracking any object. In *ECCV*, 2020. 1, 2, 3, 4, 6, 7, 8

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 4

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4, 5

[12] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 3

[13] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Graph convolutional tracking. In *CVPR*, 2019. 3

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013. 1

[15] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 2, 4, 5, 6, 7, 8

[16] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 3, 7

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[18] Xinting Hu, Yi Jiang, Kaihua Tang, Jingyuan Chen, Chunyan Miao, and Hanwang Zhang. Learning to segment the tail. In *CVPR*, 2020. 2

[19] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. Video instance segmentation using inter-frame communication transformers. In *NeurIPS*, 2021. 3

[20] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *J. Fluids Eng.*, 1960. 3

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2

[22] Lei Ke, Xia Li, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Prototypical cross-attention networks for multiple object tracking and segmentation. In *NeurIPS*, 2021. 3

[23] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomáš Vojíř, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *PAMI*, 2016. 1, 3

[24] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019. 3

[25] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *CVPR*, 2020. 2, 5

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 2

[27] Dongfang Liu, Yiming Cui, Wenbo Tan, and Yingjie Chen. Sg-net: Spatial granularity network for one-stage video instance segmentation. In *CVPR*, 2021. 3

[28] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 2

[29] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 2

[30] Zhichao Lu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Retinatrack: Online single stage joint detection and tracking. In *CVPR*, 2020. 3

[31] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 2020. 1, 3

[32] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 5

[33] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 3

[34] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 3

[35] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *CVPR*, 2021. 2, 3, 4, 5, 6, 7, 1

[36] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 3

[37] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip Torr, and Song Bai. Occluded video instance segmentation. *arXiv preprint arXiv:2102.01558*, 2021. 3

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 3, 6

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2, 3

[40] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016. 5

[41] Peize Sun, Jinkun Cao, Yi Jiang, Rufeng Zhang, Enze Xie, Zehuan Yuan, Changhu Wang, and Ping Luo. Transtrack: Multiple-object tracking with transformer. *arXiv preprint arXiv: 2012.15460*, 2020. 1, 3

[42] Jingru Tan, Xin Lu, Gang Zhang, Changqing Yin, and Quanquan Li. Equalization loss v2: A new gradient balance approach for long-tailed object detection. In *CVPR*, 2021. 2

[43] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *CVPR*, 2020. 2

[44] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 2

[45] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 4

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 4

[47] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *CVPR*, 2019. 1, 3

[48] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *CVPR*, 2021. 2

[49] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, 2020. 2

[50] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2020. 3, 6, 7

[51] Zhongdao Wang, Hengshuang Zhao, Ya-Li Li, Shengjin Wang, Philip Torr, and Luca Bertinetto. Do different tracking tasks require different appearance models? In *NeurIPS*, 2021. 1

[52] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 3

[53] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2

[54] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, 2019. 2, 3, 5, 6, 7

[55] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Crossover learning for fast online video instance segmentation. In *ICCV*, 2021. 3, 6, 7, 2

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 4

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4

[58] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *ECCV*, 2020. 1, 3

[59] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 3

[60] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. In *CVPR*, 2018. 3