GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild

Lianghua Huang, Xin Zhao, *Senior Member, IEEE,* and Kaiqi Huang, *Senior Member, IEEE*

Abstract—In this work, we introduce a large high-diversity database for generic object tracking, called GOT-10k. GOT-10k is backboned by the semantic hierarchy of WordNet [1]. It populates a majority of 563 object classes and 87 motion patterns in real-world, resulting in a scale of over 10 thousand video segments and 1.5 million bounding boxes. To our knowledge, GOT-10k is by far the richest motion trajectory dataset, and its coverage of object classes is more than a magnitude wider than similar scale counterparts [20], [23]. By publishing GOT-10k, we hope to encourage the development of generic purposed trackers that work for a wide range of moving objects and under diverse real-world scenarios. To promote generalization and avoid the evaluation results biased to *seen* classes, we follow the one-shot principle [35] in dataset splitting where training and testing classes are *zero-overlapped*. We also carry out a series of analytical experiments to select a compact while highly representative testing subset – it embodies 84 object classes and 32 motion patterns with only 180 video segments, allowing for efficient evaluation. Finally, we train and evaluate a number of representative trackers on GOT-10k and analyze their performance. The evaluation results suggest that tracking in real-world unconstrained videos is far from being solved, and only 40% of frames are successfully tracked using top ranking trackers. The database and toolkits are publicly available at https://got-10k.github.io.

Index Terms—Object tracking, benchmark dataset, performance evaluation.

1 Introduction

Generic object tracking refers to the task of sequentially locating a moving object in a given video, without accessing to the prior knowledge (e.g., the object class) about the object as well as its surrounding environment [8], [18]. The task is highly challenging not only because of the class-agnostic nature in its definition, but also due to the unpredictable appearance changes and background distractions occurred during the tracking process, such as occlusion, object deformation and cluttered background. In real life, generic object tracking has a wide range of applications, such as video editing, intelligent surveillance, autonomous vehicle and human-computer interaction [7], [8], [18]. Moreover, generic object tracking requires very few supervision during the tracking process. By exploring this, recent advances [29], [30] have further shown its potential in actively mining the training samples in unlabeled videos, paving the way for more automatic learning system.

Over the past few decades, generic object tracking follows a standard setting that the only available supervision is the annotated object location at the first frame, and no extra labeled videos are employed during the training process [6], [7], [52], [63]. Traditional trackers are based on this setting and in the past several years a wide range of such algorithms have been proposed [47], [49], [51], [52], [67]. For these methods, the tracking models are typically learned from scratch at the first frame, then target localization and model updating are iteratively conducted online to track the object. At the same time, to provide a unified platform for a comprehensive evaluation and comparison of trackers, a number of datasets have emerged and served as benchmarks for generic object tracking [2], [8], [10], [12], [14], [18]. Following traditional setting, these datasets only contain testing videos and there is no split of training data.

More recently, the paradigm of tracking algorithms is experiencing a change. With the growing popularity of using deep learning in a wide range of computer vision tasks, several works start to explore the representation capabilities of deep neural networks to improve tracking performance [56], [60], [62]. Representative methods include convolutional neural networks based trackers [56], [62], siamese trackers [38], [60], [61] and policy learning based methods [59], [66]. These methods introduce a different paradigm to tracking. They typically learn some universal representations offline from a large set of labeled videos, such as feature representation, high dimensional metric space and decision policies, then transfer the knowledge to testing videos and use it for object tracking with minimal or no tuning. Despite their success in improving tracking accuracy, the implicit experimental settings of them are usually casual. For example, varied training data, such as OTB [8], VOT [3], ALOV++ [18], NUS_PRO [14], ImageNet-VID [20] and YouTube-BB [23] are used in the training processes of different trackers [38], [56], [69], making the fair comparison of their results infeasible. In addition, in many cases the object classes in the training and testing videos are overlapped [56], [69]. In that case, the evaluation results could be biased and they cannot reflect the generalization ability of trackers on a wide range

- *L. Huang and X. Zhao are with the Center for Research on Intelligent System and Engineering, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: huanglianghua2017@ia.ac.cn, xzhao@nlpr.ia.ac.cn.*
- *K. Huang is with the Center for Research on Intelligent System and Engineering and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China, and the CAS Center for Excellence in Brain Science and Intelligence Technology, 100190 (kqhuang@nlpr.ia.ac.cn).*

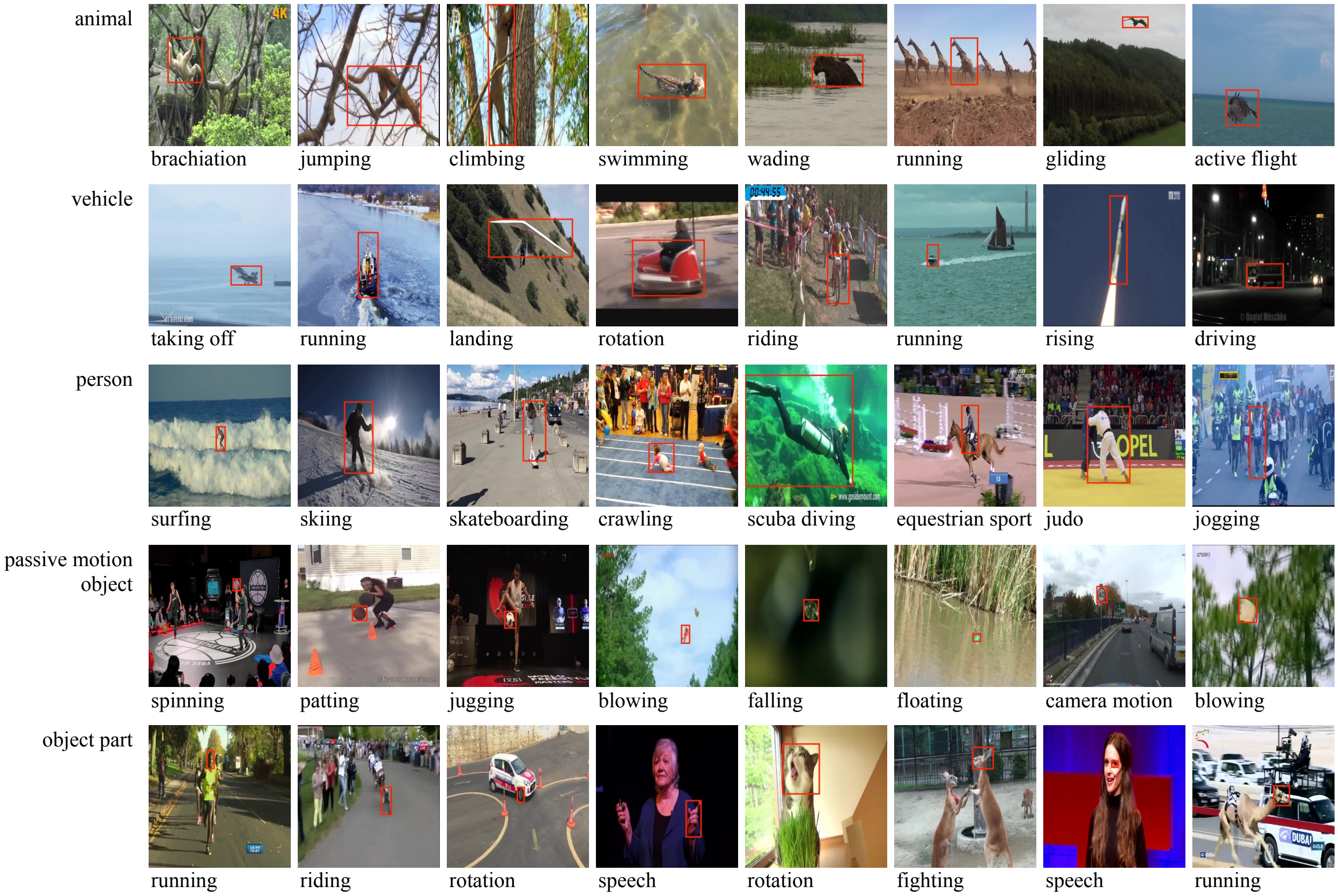


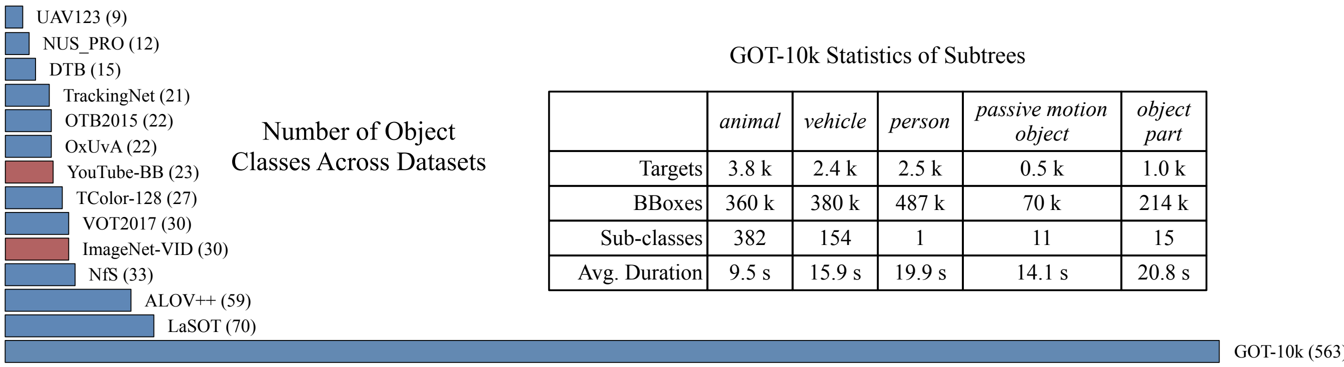
Fig. 1: Screenshots of some representative videos collected and annotated in GOT-10k. Each video is attached with two semantic labels: object and motion classes. The object classes in GOT-10k are populated based on the semantic hierarchy of WordNet [1]. It expands five subtrees: *animal*, *artifact*, *person*, *natural object* and *part* (for clarity, in the figure, we split *artifact* into *vehicle* and *passive motion object* classes and categorize *natural object* as *passive motion object*) from WordNet nouns to cover the majority of both natural and artificial moving objects in real-world. The motion classes are partially backboned by WordNet while others are manually defined by the collectors. GOT-10k populates 563 object classes and 87 motion classes in total. From the screenshots we can also find that introducing motion labels to data collection largely improves the variety of our dataset.

TABLE 1: Comparison of GOT-10k against other tracking datasets in terms of: scale of manually labeled bounding boxes, number of targets and object classes, dataset splits as well as experimental settings. GOT-10k is magnitudes larger than most datasets and it provides a wider coverage of object classes. Although TrackingNet and LaSOT are on par with our dataset in video scale, they only contain limited number of object classes and their training and evaluation object classes are fully overlapped, leading to biased evaluation results.

	GOT-10k	OTB2015 [8]	VOT2017 [2]	ALOV++ [18]	NUS_PRO [14]	TColor128 [13]	NfS [10]	OxUvA [9]	TrackingNet [16]	LaSOT [17]
BBoxes	**1.5 M**	59 k	21 k	16 k	135 k	55 k	38 k	155 k	509 k	**3.25 M**
Targets	**10 k**	100	60	314	365	129	100	366	$>$ **30 k**	1.4 k
Classes	**563**	22	30	59	12	27	33	22	21	70
Subsets	**train+val.+eval.**	eval.	eval.	eval.	eval.	eval.	eval.	dev.+eval.	train+eval.	train+eval.
Exp. Setting	**one-shot**	casual	casual	casual	casual	casual	casual	open+ constrained	fully-overlapped	fully-overlapped

of unseen objects and scenarios.

Based on the above discussions, the purpose of this work is twofold. On one hand, we would like to provide a unified platform with principled experimental setting to enable practical evaluation and fair comparison of deep trackers. On the other hand, as promising performance is achieved on a number of constrained small datasets, we want to approach a step closer toward the definition of ”generic”, and establish a comprehensive database where the evaluation results can be better generalized to challenging real-world scenarios. To this end, we construct GOT-10k, a large-scale tracking dataset with a wide coverage of objects and scenarios. Like the ImageNet database [21], GOT-10k is backboned by the semantic hierarchy of WordNet [1]



GOT-10k Statistics of Subtrees

	animal	*vehicle*	*person*	*passive motion object*	*object part*
Targets	3.8 k	2.4 k	2.5 k	0.5 k	1.0 k
BBoxes	360 k	380 k	487 k	70 k	214 k
Sub-classes	382	154	1	11	15
Avg. Duration	9.5 s	15.9 s	19.9 s	14.1 s	20.8 s

Fig. 2: BAR CHART: Number of object classes in different tracking (blue) and video image detection (red) datasets. Among all the compared datasets, GOT-10k offers an unprecedentedly wider coverage of moving object classes. TABLE: Statistics of five subtrees populated in GOT-10k. All 563 object classes in GOT-10k are expanded from the five subtrees. The table shows the number of tracking targets, the scale of manually annotated object bounding boxes, sub-class counts and average video length per each subtree.

to ensure a comprehensive coverage. GOT-10k covers a majority of 563 object classes and 87 motion classes in real-world, with a scale of over 10 thousand manually annotated video segments. To our knowledge, GOT-10k is by far the richest video trajectory dataset and its coverage of object classes is more than a magnitude wider than its similar-scale counterparts [16], [20], [23]. Figure 1 shows some representative screenshots of GOT-10k with each video labeled with both object and motion classes. Figure 2 visualizes the comparison of different video datasets in terms of variety while Table 1 shows a comprehensive comparison of GOT-10k against other tracking datasets. Unlike existing datasets, GOT-10k follows the standard experimental setting of *one-shot learning* [35], [36] that the object classes between training and testing videos are *non-overlapped*, while the *support set* [35] corresponds to the ”one-shot example” for each testing video. The principle enables a relatively unbiased evaluation of tracking on anonymous objects and scenarios, and is closer to the definition of generic object tracking.

Before going into the details of our benchmark, we summarize the contributions of this work in the following:

Dataset. We construct a manually labeled large-scale dataset with a wide coverage of real-world moving objects, motion patterns and scenarios for generic object tracking. The dataset enables the training of data-hungry deep models as well as a more practical performance evaluation of generic purposed trackers.

Benchmark. We evaluate a number of state-of-the-art approaches on GOT-10k and analyze their performance in this work. Our results suggest that the performance of real-world tracking is far from reaching human accuracy and many challenging issues remain unsolved. We also analyze the limitations of existing tracking frameworks and discuss possible directions for future works.

Evaluation protocol. The contributions of this work on evaluation protocol of generic object tracking are two-fold. On one hand, we introduce the *one-shot* principle to achieve a relatively unbiased evaluation of deep trackers. On the other hand, we carry out a series of experiments to select a compact while highly representative testing subset, allowing for efficient evaluation.

The remainder of this paper is organized as follows. In Section 2, we make a review of tracking algorithms and related datasets. Section 3 provides the technical details of GOT-10k. The experimental results and analysis are presented in Section 4. We conclude our work and discuss future works in Section 5.

2 Related Work

2.1 Trackers

Recent years have witnessed impressive progress in the field of generic object tracking and its performance has been continuously improved in terms of both accuracy, robustness and tracking speed [2], [8]. In this section, we make a brief review on recent tracking approaches.

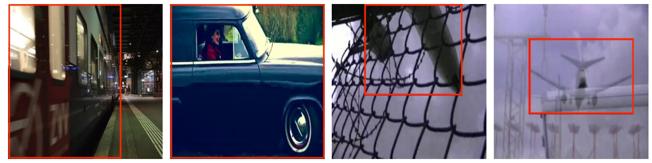
Traditional trackers. Traditional methods use hand-crafted features for tracking and they typically put more efforts on the development of appearance models. Based on the appearance models used, the tracking approaches are roughly categorized as either generative [71], [72] or discriminative methods [42], [51], [52], [68]. Generative methods view tracking as a reconstruction problem and they maintain a template set [72], [73]or a subspace [71], [74] online to represent the moving target. On the other hand, discriminative trackers [42], [50], [51], [52], [68] learn a classifier between the target and its surrounding background. The classifier represents the target and it is updated online to adapt to appearance changes. Empirically, discriminative methods are more robust to background clutters than generative counterparts [2], [8]. This is because the discriminative models explicitly suppress the negative distractions in their classifier learning.

Among the discriminative approaches, correlation filters (CF) [49], [50], [75] exhibit very competitive performance with impressive tracking speed, and are drawing extensive attention in recent years [37], [40], [46], [58], [61]. The key idea behind the correlation filters is to approximate dense image sampling by circulant shift on a single centered image patch [49], [75]. This approximation allows both training and inference to be fast implemented in the Fourier domain. The MOSSE tracker [75] is considered the first approach to introduce correlation filters to object tracking. This approach considers tracking as a regularized least squares problem and reformulates the closed-form solution of it in a correlation filters framework. MOSSE is able to achieve a reasonable tracking performance with a very high speed (∼700 fps). Later on, several improvements on MOSSE have been proposed. They include introducing non-linear kernels [49] and long-term dependencies [55], accurate scale estimation [41], [53], the use of multiple feature channels [37], [46], [50], [58], multiple templates matching [40], [46] and boundary effect removal [44], [45], [76].

Unlike many areas in computer vision where deep learning methods outperform traditional ones by a large margin [77], [78], [79], in generic object tracking, traditional approaches still play an important role and some of them can achieve a performance on par with deep models using only hand-crafted features [37], [45], [76]. Even so, the further improvement of these trackers is largely restricted by the limited representation power of traditional features and the finite prior knowledge they can hold.

Deep trackers. More recently, several attempts have been made to use deep learning to improve tracking performance [54], [60], [62]. Among various deep neural networks, convolutional neural networks (CNNs) [79] are most widely used. Aside from those methods that directly employ pretrained networks for feature extraction [40], [46], [54], researchers have developed a number of effective architectures for end-to-end learning of target appearance models [38], [56], [62]. For example, the MDNet tracker [56] separates domain specific layers of the network from domain agnostic ones, and the domain specific part is re-trained on each testing video to adapt to new environment. The siamese tracker [38], [60] and its variants [61], [69], [80] learn a high dimensional metric space between the exemplar and search patches, thus target searching is reduced to the simple task of feature matching. Some works [61] reformulate the correlation filters as an end-to-end learnable network, thus benefit from both the learning efficiency of correlation filters as well as the representation power of deep neural networks. In addition to convolutional neural networks, the tracking community has also explored other advanced deep models. For example, some works view tracking as a sequential decision making task and learn policies to guide the online actions [59], [66]. Some other works explore recurrent structures for sequential prediction in tracking [64], [65]. To fill the domain gap between training and testing videos, many works explore the attention mechanism [39], [70] or meta learning framework [57] to fast adapt the models to new domains.

Deep learning in essential introduces a novel paradigm to generic object tracking – instead of learning from scratch with a single labeled sample, deep learning allows models to gain certain universal skills and representations from a large set of extra training videos, and generalize the prior knowledge to unseen objects and scenarios with slight tuning. Such a mechanism confers on deep models the potential to achieve much higher tracking accuracy than traditional approaches. For example, by acquiring knowledge on high level semantics such as *object*, *motion* and *environment* from training data, deep trackers have the possibility to achieve a tracking robustness close to human vision. It is also the possibility we want to explore through our large high-diversity dataset and benchmark.



(a) ImageNet-VID. (b) YouTube-BB.

Fig. 3: Screenshots taken from the YouTube-BB [23] and ImageNet-VID [20] datasets. Many of their videos contain noisy segments such as incomplete objects and shot changes, making them less optimal for the tracking task.

TABLE 2: Statistical comparison of GOT-10k against video trajectory datasets of related areas, including Multiple Object Tracking (MOT) and Video Image Detection (VID). GOT-10k exhibits a significant superiority in the coverage of diverse object classes.

Datasets	Classes	BBoxes	Videos
KITTI [28]	4	59 k	50
MOT15 [27]	1	101 k	22
MOT16/17 [26]	5	293 k	14
ImageNet-VID [20]	30	1.03 M	5.4 k
YouTube-BB [23]	23	**5.6 M**	**380 k**
GOT-10k	**563**	**1.5 M**	**10 k**

2.2 Datasets

We discuss in this section some of the datasets that are most related to GOT-10k.

Object tracking benchmarks. Since 2013, a number of object tracking datasets have been proposed and served as unified platforms for tracker evaluation and comparison. The OTB [7], [8], ALOV++ [18] and VOT [5], [6] datasets represent the initial attempts to unify the testing data and performance measurements of generic object tracking. The OTB collects 51 and 100 moving objects respectively from previous works in its first [7] and second [8] versions, while the ALOV++ [18] provides a larger pool of over 300 videos. The VOT [2], [3], [6] is an annual visual object tracking challenge held every year in conjunction with ICCV and ECCV workshops since 2013. Later on, several other datasets have been proposed targeting on solving specific issues. They include the large scale people and rigid object tracking dataset NUS_PRO [14], long-term aerial tracking dataset UAV123 [12], color tracking dataset TColor-128 [13], long-term tracking dataset OxUvA [9], thermal tracking datasets PTB-TIR [19] and VOT-TIR [3], RGBD tracking dataset PTB [15] and high frame rate tracking dataset NfS [10]. These datasets play an important role in boosting the development of tracking methods. However, they are all small-scale with limited diversity, and many of the collected videos are captured under constrained scenarios. These disadvantages restrict the further development of tracking algorithms.

More recent datasets TrackingNet [16] and LaSOT [17] offer a scale that is on par with our dataset. TrackingNet chooses around 30 thousand videos from YouTube-BB [23]

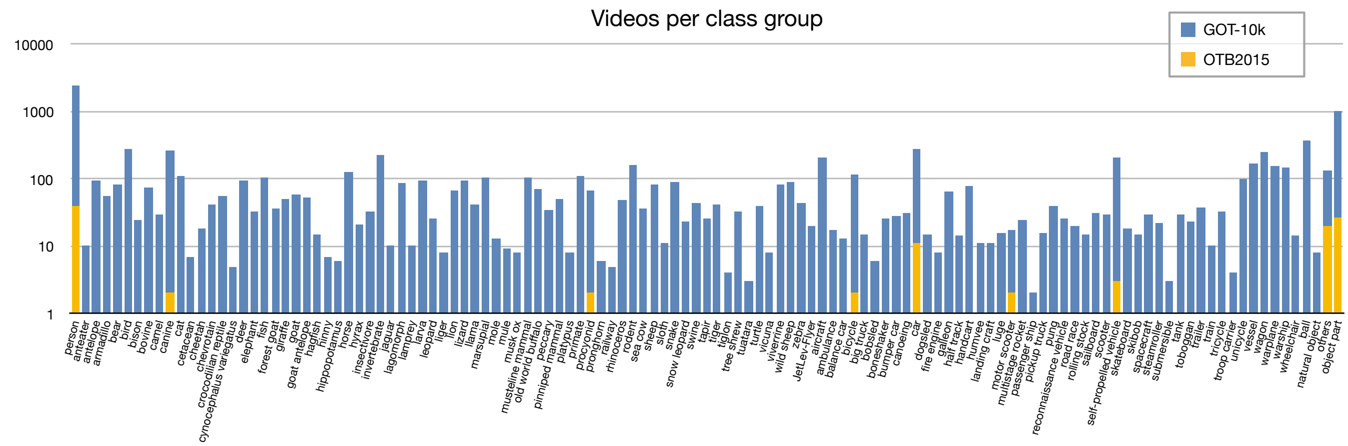


Fig. 4: Number of videos per each group of object classes. In the collection stage, we categorize all potential object classes into 121 groups that we would like to ensure each being collected, as described in Section 3.1. The plot shows the final distribution of these groups in GOT-10k, with OTB2015 dataset as a comparison.

to form its training subset, and it collects another 500 videos with similar class distribution as its evaluation subset, while LaSOT collects and annotates 1.4 thousand videos manually. However, both datasets only contain limited number of object classes (21 and 70 classes respectively), which is not enough for both training and evaluation of generic object trackers. Moreover, their training and evaluation object classes are fully overlapped with close distribution, thus the evaluation results are highly biased to specific object classes and the performance fail to reflect the generalization ability of trackers on a wide range of unseen objects. Table 1 compares GOT-10k with existing tracking datasets in terms of scale, diversity and experimental setting. GOT-10k is magnitudes larger than most tracking datasets and it offers a much wider coverage of moving objects. Furthermore, it follows a *one-shot* setting of tracking experiment to encourage the development of generic purposed trackers.

Large-scale video datasets. In recent years, there have been growing interests in conducting research on videos. A number of large-scale video datasets for classification [24], [25], object detection [20], [23] and multiple object tracking [26], [27], [28] have been proposed. Among them, the ImageNet Video Image Detection (ImageNet-VID) [20] and YouTube Bounding Boxes (YouTube-BB) [23] are the two datasets that are most related to the tracking community – both have been frequently used as extra training videos in deep tracking algorithms [38], [69], [80]. The ImageNet-VID collects more than 5 thousand videos and annotates bounding boxes for 30 classes of moving objects. The YouTube-BB provides a larger video pool, it consists of over 380 thousand videos with 5.6 million bounding boxes annotated at 1 fps. However, the YouTube-BB exhibits a lower diversity with only 23 object classes populated.

Another research field close to generic object tracking is multiple object tracking (MOT) [26], [27], [28]. Nevertheless, MOT is a model-specific tracking task and it focuses on tracking specific classes of objects, typically persons and vehicles. Table 2 compares GOT-10k with popular video trajectory datasets in terms of scale and diversity. GOT-10k offers an annotation scale that is on par with ImageNet-VID and YouTube-BB, but it provides more than a magnitude wider coverage of diverse moving objects. Moreover, the video image detection datasets contain noisy segments such as short trajectories, incomplete objects and shot changes (Figure 3), while GOT-10k always provides clean and continuous long trajectories.

TABLE 3: Quality control pipeline for video collection. The video collection as well as the first 3 stages of verification (marked with $*$) are conducted in a qualified data company.

Stage	Description	Executer	Proportion
1^*	Data collection	Collectors	100%
2^*	Verification	Collectors	$15\% \sim 30\%$
3^*	Verification	Project team	$15\% \sim 30\%$
4^*	Verification	Verification team	$5\% \sim 10\%$
5	Data screening	Our trained verifiers	100%
6	Data acceptance	The authors	20%

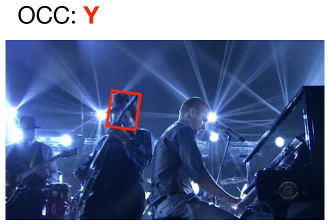
TABLE 4: Quality control pipeline for trajectory annotation. The trajectory annotation as well as the first 3 stages of verification (marked with $*$) are conducted in a qualified data company.

Stage	Description	Executer	Proportion
1^*	Data annotation	Annotators	100%
2^*	Verification	Annotators	$15\% \sim 30\%$
3^*	Verification	Project team	$15\% \sim 30\%$
4^*	Verification	Verification team	$5\% \sim 10\%$
5	Data acceptance	The authors	20%

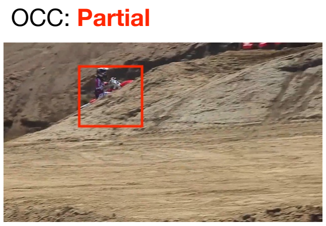
3 Construction of GOT-10k

In this section, we describe technical details on the strategies and pipelines we use to construct GOT-10k, shedding light on how the quality, coverage and accuracy of GOT-10k are ensured. We also show the experiments we carry out for the selection of a comprehensive and compact testing subset.

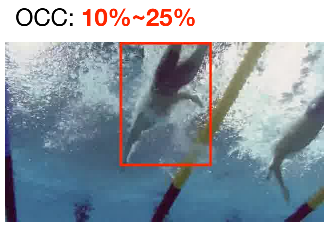
(a) OTB.



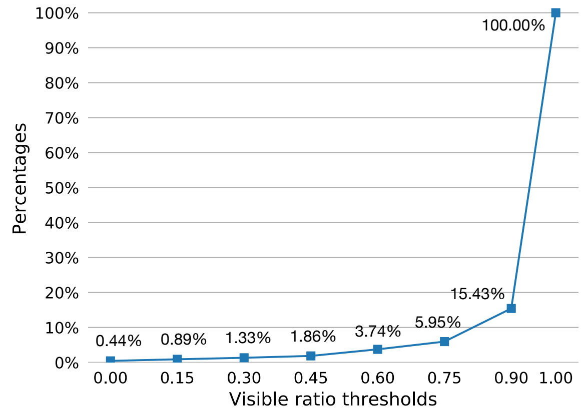
(b) VOT.



(c) NUS_PRO.



(d) GOT-10k.



(e) Cumulative distribution of visible ratios in GOT-10k.

Fig. 5: (a)-(d) Per-frame occlusion labeling in popular tracking datasets and GOT-10k. (a) The OTB dataset provides no frame-wise labeling of occlusion, while (b) the VOT dataset offers a binary label for each frame indicating whether or not the target is occluded. (c) The NUS_PRO dataset distinguishes between partial and complete occlusion, while (d) GOT-10k further provides a more continuous labeling of target occlusion status. (e) Cumulative distribution of object visible ratios in GOT-10k. In around 15.43% of frames the targets are occluded (with less than 90% visible), and in approximately 1.86% of frames, they are heavily occluded (with less than 45% visible). In about 0.43% of frames the targets are absent (being fully occluded or out-of-view).

3.1 Collection of Videos

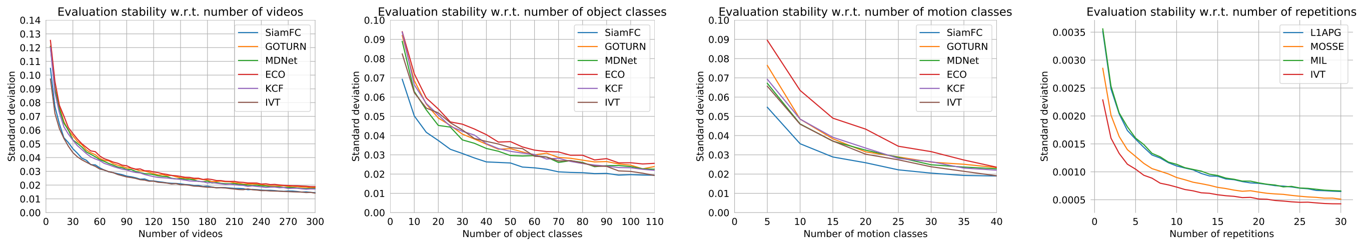
The purpose of this work is to construct a large-scale motion trajectory dataset that covers as many object classes, motion patterns and scenarios in real-world as possible. To achieve such a wide coverage, we use WordNet [1] as the backbone for the selection of object and motion classes. WordNet is a lexical database of English and it groups and organizes words (nouns, verbs, adjectives and adverbs) based on their meanings. For example, nouns in WordNet are linked according to the *hyponymy* relation (i.e., *is a kind/member of*) to form a tree structure. The root node of nouns is *entity* and it has several particulars such as *object*, *thing*, *substance* and *location*.

Each collected video in GOT-10k is attached with 2 dimensional labels: object and motion classes. We expand five nouns: *animal*, *person*, *artifact*, *natural object* and *part* in WordNet to collect an initial pool of potential classes of moving objects, and we expand *locomotion*, *action* and *sport* to collect motion classes. By manually filtering and pruning word subtrees (e.g., removing extinct, static and repeated object classes, grouping close sub-classes, etc.), we obtain a pool of around 2,500 object classes and 60 motion classes. Although we can directly send these words to data collectors for video acquisition, there contain many uncommon object classes (e.g., *broadtail*, *abrocome* and *popinjay*) making the collection process less efficient. To improve the efficacy, we first categorize the 2,500 object classes into 121 groups (e.g., *larva*, *canine*, *invertebrate* and *primate*) that we would like to ensure each being collected, then we rank the object classes in each group based on their corresponding searching volumes on the YouTube website over the last year. The searching volume reflects the popularity and number of uploaded videos of each word, thus the ranking can guide collectors to find qualified videos with a better chance.

We employ a qualified data company for video collection. The overall pipeline of video collection as well as verification is listed in Table 3. In summary, we carry out 1 collection stage and 5 verification stages to ensure the quality of each collected video. Defective videos that contain noisy segments such as shot changes, long-term target absence and incomplete trajectories are filtered out during the verification stages. The data collection and the first 3 stages of verification are conducted in the data company. After receiving the videos, we have 2 trained verifiers to fast go through all the collected videos and determine whether to accept each or not. Finally, the authors of this work will randomly select 20% of the accepted videos and do the last check. The final pool contains 563 classes of moving objects and 87 classes of motion (some of the motion classes are manually defined by the collectors and post-processed by the authors), with a total scale of around 10 thousand videos. Figure 1 illustrates some screenshots of GOT-10k videos labeled with varied object and motion classes. Figure 4 shows the final distribution of video counts of the 121 object class groups we used in the collection stage, with OTB2015 dataset as a comparison.

3.2 Annotation of Trajectories

We follow the standard rules in object detection [22] for the labeling of object bounding boxes in GOT-10k. Note this differs from some visual tracking datasets such as VOT [3], [6], where the optimal object bounding box is defined as the one with minimum background pixels contained. Since object tracking has been frequently used in a number of related areas such as video image detection [31], [32] and segmentation [33], multiple object tracking [34] and



(a) Impact of video number. (b) Impact of object classes. (c) Impact of motion classes. (d) Impact of repetition time.

Fig. 6: The impact of different configurations of testing videos on evaluation stability. A higher standard deviation indicates a less reliable evaluation. Better viewed with zooming in.

self-supervised learning [29], [30], keeping a compatible annotation standard encourages the development of more practical trackers.

In addition to object bounding boxes, GOT-10k also provides the annotation of visible ratios. A visible ratio is a percentage indicating the approximate proportion of an object that is visible. The pixels that are occluded or cut by image correspond to the invisible part. As is indicated in many tracking benchmarks [2], [8], [14], occlusion is one of the most challenging factors that can easily cause tracking failure. We hope the labeling of visible ratios can provide extra supervision and help improve tracking robustness under occlusion. We divide visible ratios into 7 ranges with a step of 15%. Figure 5 (a)-(d) compares the per-frame occlusion labeling of different tracking datasets while Figure 5 (e) shows the cumulative distribution of visible ratios annotated in our dataset. Through visible ratios, GOT-10k provides a more continuous labeling of target occlusion status (the percentage of occlusion can be easily calculated as $(1 - v)$ using the visible ratio v). Similar to video collection, we use 1 annotation stage and 4 verification stages to ensure the quality of each annotation. Table 4 lists out the pipeline. The object annotation and the first 3 stages of verification are conducted in the data company while the authors of this paper will randomly check 20% of the submitted results. The acceptance criterion is a qualified rate of above 95%.

3.3 Dataset Splitting

We split the GOT-10k dataset into unified training, validation and testing subsets to enable fair comparison of tracking approaches. Unlike many other machine learning applications [21], [23], the splitting of generic object tracking dataset is not straightforward (i.e., by randomly sampling a proportion of data). For one thing, we expect the evaluation results to reflect the generalization ability of different approaches on a wide range of objects and scenarios. To achieve this, an explicit domain gap between training and testing videos has to be established. For another, we do not need thousands of videos to assess a tracking algorithm. Besides, the evaluation of trackers is very time-consuming, thus it would be favorable to keep the testing subset compact.

With the first consideration, we follow the *one-shot* principle [35] and set up a strict rule that the object classes in training and testing videos are *non-overlapped*; the *person* class, however, is treated as an exception. Unlike other

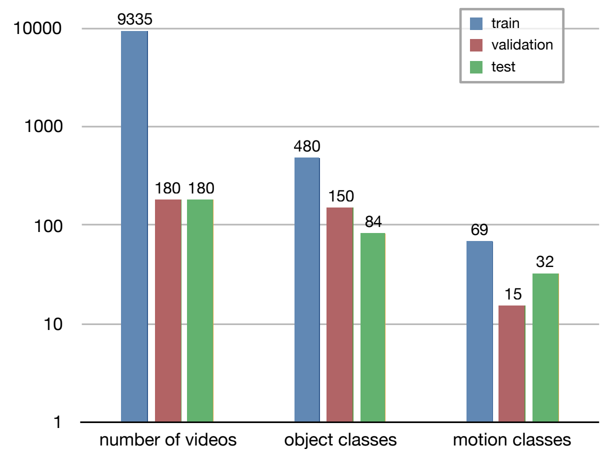


Fig. 7: Dataset splits of GOT-10k. Except for the *person* class, all object classes between training and testing videos are not overlapped; while for *person*s, the motion classes between training and testing are not overlapped. The validation subset is randomly sampled from training videos with uniform distribution across different object classes.

classes of objects, *person*s embrace a rich collection of motion patterns like *jogging, swimming, skiing, crawling, cycling, diving, equitation, judo* and *surfing*, to name a few. Each motion pattern represent a special combination of challenges and thus form a problem domain. We argue that an object class with such a variety is of great interest in both training and evaluation, therefore we include *person*s in both training and testing subsets. To also introduce a domain gap, we ensure in the splitting that the motion classes of *person*s between training and testing are *non-overlapped*.

To address the second consideration, we carry out a series of experiments to find a reliable while compact testing subset. We take the testing subset as a random variable and draw its samples from a large pool of around a thousand videos. Then we run tracking experiment on each sample and evaluate the tracking performance. The standard deviation of evaluation scores (we use the average overlap for simplicity) is calculated as the indicator of evaluation stability, as practiced in [9]. Results are visualized in Figure 6. We analyze each influence factor in the following.

Impact of video number. We vary the number of testing videos from 5 to 300, with a step of 5. Figure 6a shows that the standard deviation significantly decreases as the video number increases, which indicates an improved evaluation stability. A reasonable stability is observed at the point of around 150 videos, where further raise the video number only marginally reduces the deviation (the reduction is less than 0.01 when video number doubles).

Impact of object classes. We fix the video number to 180 and change the sampled object classes from 5 to 110, results are visualized in Figure 6b. We observe an obvious downward trend of standard deviation as more object classes are included. This suggests the importance of dataset diversity to the stability of performance evaluation. The trend approximately converges at the point of around 80 object classes.

Impact of motion classes. With the number of videos fixed at 180, we vary the number of motion classes from 5 to 40. Figure 6c shows the impact of motion classes on evaluation stability. The evaluation stability generally improves as more classes of motion are included in the testing subset. At the point of around 30 motion classes, the trend converges and stable evaluation can be achieved.

Impact of repetition time. Many tracking benchmarks require trackers to run multiple times (e.g., 15 times in VOT challenges [2]) on their datasets to ensure a reliable evaluation. This significantly multiplies the evaluation costs. In this work, we would like to quantitatively analyze the impact of repetition time on the stability of performance evaluation. Figure 6d shows the improvement of evaluation stability of several stochastic trackers as the repetition time increases from 1 to 30, with video number fixed at 180. Compared to other influence factors, we find the contribution of increasing repetition time to evaluation stability is negligible (at the order of 0.001) when evaluated on a testing subset with 180 videos. Considering the stochastic character of many trackers, we set the number of repetitions to 3, which is fairly enough for a stable evaluation.

According to the above analysis, the final splits of GOT-10k dataset are summarized in Figure 7. The testing subset contains 180 videos, 84 classes of moving objects and 32 forms of motion, where a highly reliable evaluation is observed at such a setting in the above experiments. Except for the *person* class, all object classes between training and testing videos are *non-overlapped*; while for *person*s, the motion classes between training and testing are not overlapped. The validation subset is selected by randomly sampling 180 videos from the training subset, with uniform probabilities across different object classes. For each stochastic tracker, we run 3 times of experiments and average the scores to ensure a reliable evaluation.

4 Experiments

We carry out extensive experiments on GOT-10k and analyze the results in this section. We expect the baseline performance to offer an aspect of overall difficulty of GOT-10k as well as a point of comparison for future works. We also discuss the challenges of real-world tracking, and the impact of training data on the performance of deep trackers.

4.1 Baseline Models

In recent tracking benchmarks and challenges [2], [3], [9], the state-of-the-art performance is mainly prevailed by either deep learning or correlation filters based methods. Therefore, we consider typical trackers of these two categories in our benchmark. We also evaluate some traditional pioneer works of generic object tracking for comparison. The baseline trackers evaluated in our benchmark are briefly described in the following.

Deep learning based trackers. A number of deep learning based tracking algorithms have been developed in recent years. In this work, we consider the convolutional neural networks based trackers MDNet [56], GOTURN [48] and CF2 [54] as well as siamese trackers SiamFC [38] and CFNet [61]. We also evaluate some of their variants, including SiamFCv2 [61], which is the baseline method of CFNet and it wins the realtime challenge of VOT2017 [2]; and CFNetc1 and CFNetc2, the variants of CFNet that use only the first one and two convolutional layers of CFNet respectively but exhibit comparable or even better performance [61]. To be fair, for all these trackers, we retrain them on GOT-10k's training subset with their default parameter settings.

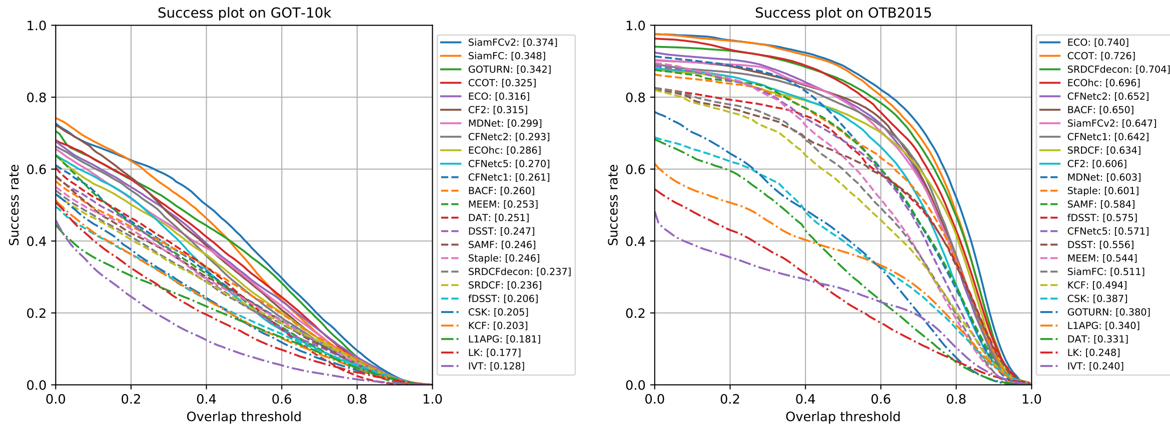
Correlation filters based trackers. We consider pioneer work CSK [49] and its variants KCF [50], DAT [58], LCT [55], SAMF [53], DSST [41], Staple [37], SRDCF [44], SRDCFdecon [45], CCOT [46], BACF [76] and ECO [40] in our evaluation. MOSSE [75] is considered the first approach to introduce correlation filters to object tracking. CSK introduces nonlinear kernels to correlation filters. DAT, KCF and Staple extends CSK with multi-channel visual features. SAMF and DSST propose efficient scale searching schemes for correlation filters tracking. To tackle with boundary effects, SRDCF and SRDCFdecon apply spatial regularization on learned filters, while BACF uses center cropping on larger shifted samples to remove the influence of boundaries. CCOT presents a continuous convolution operator to integrate multi-layer features of convolutional neural networks, while ECO raises both the speed and accuracy of CCOT with several improvements. We also consider ECOhc, the variant of ECO that uses traditional HoG and color-name features.

Traditional trackers. In addition to popular correlation filters and deep learning based trackers, we also evaluate some traditional pioneer works. They include generative methods LK [82], IVT [71] and L1APG [72] and discriminative method MEEM [42]. Although these trackers are not state-of-the-art in recent benchmarks, their algorithm designs may inspire future works, thus we also evaluate them in our benchmark as a reference.

For all the baseline models, we use their public code and default parameter settings throughout our experiments. Although adjusting parameters on the validation subset of GOT-10k may improve their performance, it requires a huge amount of work. In this respect, the evaluation results in this work can be viewed as a lower bound of these algorithms.

4.2 Evaluation Methodology

In this work, we prefer to employ simple metrics with clear meaning for the evaluation of trackers. We choose the widely used average overlap (AO) and success rate (SR) as our indicators. The AO denotes the average of overlaps between all groundtruth and estimated bounding boxes, while the SR measures the percentage of successfully tracked frames where the overlaps exceed 0.5. The AO is



(a) Overall performance on GOT-10k.

(b) Overall performance on OTB2015.

Fig. 8: Overall performance of baseline trackers on GOT-10k and OTB2015, ranked by their average overlap (AO) scores.

TABLE 5: Overall tracking results of baseline trackers on GOT-10k. The trackers are ranked by their average overlap (AO) scores. The first-, second- and third-place trackers are labeled with red, blue and green colors respectively. The *Properties* column denotes the attributes of different trackers that are split into: appearance model (generative/discriminative, i.e., G/D), correlation filters (yes/no, i.e., Y/N), deep learning (yes/no, i.e., Y/N) and feature representation (HOG - Histogram of Gradients, CN - Color Names, CH - Color Histogram, Raw - Raw pixels, IIF - Illumination Invariant Features).

Tracker	**Performance**			**Properties**				**Venue**
	AO	**SR**	**Speed (fps)**	**Appr.**	**C.F.**	**D.L.**	**Repr.**	
SiamFCv2 [61]	**0.374**	**0.404**	22@GPU	D	N	Y	CNN	CVPR'17
SiamFC [38]	**0.348**	**0.353**	**39@GPU**	D	N	Y	CNN	ECCV'16
GOTURN [48]	**0.342**	**0.372**	**87@GPU**	G	N	Y	CNN	ECCV'16
CCOT [46]	0.325	0.328	0.60@CPU	D	Y	N	CNN	ECCV'16
ECO [40]	0.316	0.309	2@CPU	D	Y	N	CNN, HOG	CVPR'17
CF2 [54]	0.315	0.297	5@GPU	D	Y	N	CNN	ICCV'15
MDNet [56]	0.299	0.303	2@GPU	D	N	Y	CNN	CVPR'16
CFNetc2 [61]	0.293	0.265	31@GPU	D	Y	Y	CNN	CVPR'17
ECOhc [40]	0.286	0.276	43@CPU	D	Y	N	HOG, CN	CVPR'17
CFNetc5 [61]	0.270	0.225	22@GPU	D	Y	Y	CNN	CVPR'17
CFNetc1 [61]	0.261	0.243	**35@GPU**	D	Y	Y	CNN	CVPR'17
BACF [76]	0.260	0.262	10@CPU	D	Y	N	HOG	CVPR'17
MEEM [42]	0.253	0.235	19@CPU	D	N	N	Lab, IIF	ECCV'14
DAT [58]	0.251	0.242	47@CPU	D	Y	N	CH	CVPR'15
DSST [41]	0.247	0.223	14@CPU	D	Y	N	HOG	BMVC'14
SAMF [53]	0.246	0.241	6@CPU	D	Y	N	HOG, CN, Raw	ECCV'14
Staple [37]	0.246	0.239	21@CPU	D	Y	N	HOG, CH	CVPR'16
SRDCFdecon [45]	0.237	0.220	2@CPU	D	Y	N	HOG	CVPR'16
SRDCF [44]	0.236	0.227	5@CPU	D	Y	N	HOG	ICCV'15
fDSST [43]	0.206	0.187	26@CPU	D	Y	N	HOG	PAMI'17
CSK [49]	0.205	0.174	**109@CPU**	D	Y	N	Raw	ECCV'12
KCF [50]	0.203	0.177	**72@CPU**	D	Y	N	HOG	PAMI'15
L1APG [72]	0.181	0.174	17@CPU	G	N	N	Raw	CVPR'12
LK [82]	0.177	0.144	4@CPU	G	N	N	Raw	CVPR'02
IVT [71]	0.128	0.084	**77@CPU**	G	N	N	Raw	IJCV'08

recently proved [8] to be equivalent to the area under curve (AUC) metric employed in OTB [7], [8], NfS [10], UAV [12], TrackingNet [16] and LaSOT [17] datasets. Besides, the expected average overlap (EAO) metric used for overall ranking in VOT challenges is an approximation of AO on larger video pool. The SR metric is also used in the OTB-2015 [8] and OxUvA [9] datasets. It clearly indicates how many frames are tracked or lost, which is the concern of many applications. We use frame-pool to gather all tracking results, as in [2], then the AO and SR are calculated based on the stacked long sequence. Note this differs from OTB criteria [7] where sequence-wise scores are calculated first and then averaged over all sequences.

The success curve [7], [8] is used in our benchmark to visualize the tracking results. Each point of success curve shows the percentage of frames where the overlaps exceed a threshold. The success curve offers a continuous measurement of tracking results ranging from robustness (lower overlap rate but more tracked frames) to accuracy (higher overlap rate) [8], [81]. As discussed in Section 3.3, for each stochastic method, we run 3 times of tracking experiments and average the evaluation results to achieve a stable evaluation.

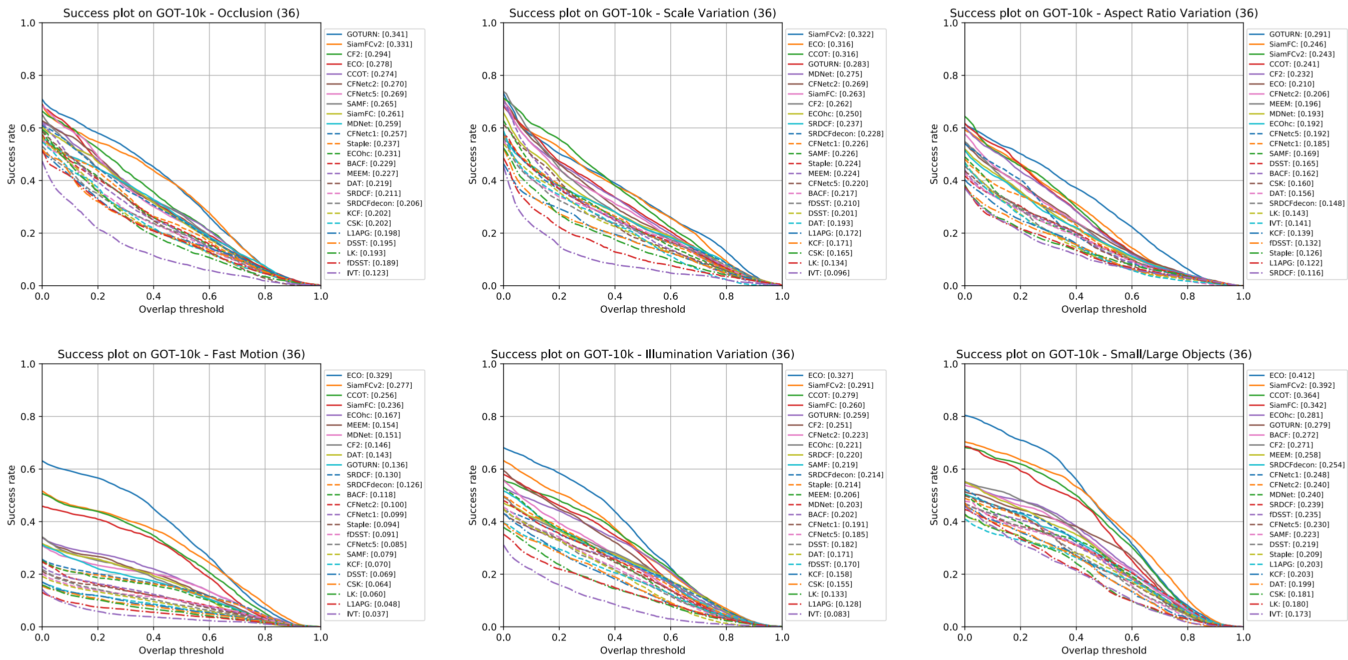


Fig. 9: Performance of baseline trackers on 6 challenging subsets: *occlusion*, *scale variation*, *aspect ratio variation*, *fast motion*, *illumination variation* and *small/large objects*. For each challenging attribute, we select the top 20% hardest videos (i.e., 36 videos) from testing subset for evaluation.

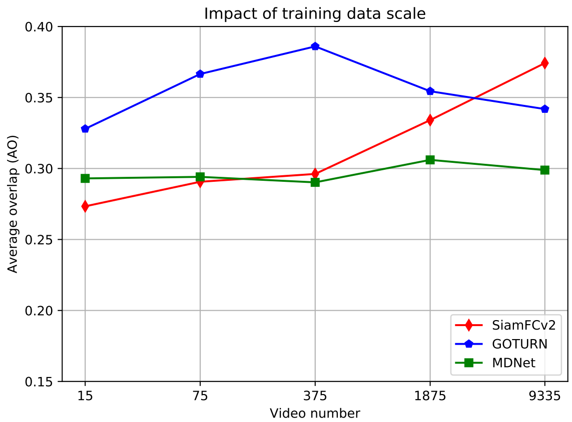
4.3 Overall Performance

We employ the average overlap (AO) and success rate (SR) for the overall evaluation of trackers, as described in Section 4.2. For deep trackers, we retrain each of them on GOT-10k to achieve a fair comparison. All experiments are run on a server with a 56 core Intel(R) Xeon(R) 2.0GHz CPU and 4 GeForce GTX TITAN X graphic cards. Table 5 illustrates the evaluation results of all baseline models, ranked by their AO scores. Figure 8a shows their success curves, with OTB2015 results in Figure 8b as a comparison.

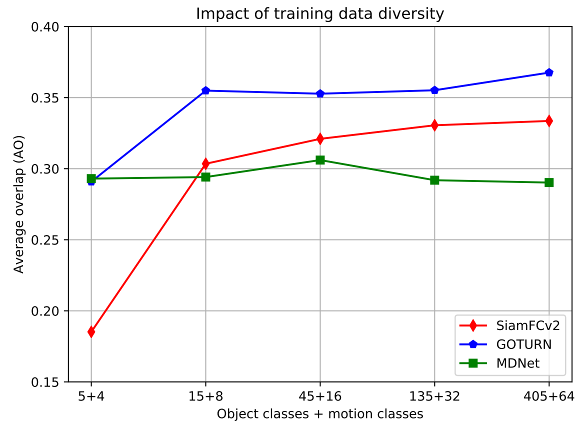
The top three trackers on GOT-10k are SiamFCv2, SiamFC and GOTURN. They all are end-to-end trainable deep trackers. SiamFCv2 outperforms others by a relatively large margin (2.6% in AO and 5.1% in SR) while SiamFC and GOTURN achieve close performance. The following three trackers CCOT, ECO and CF2 are correlation filters based methods and they use pretrained CNNs for feature extraction. Among all traditional trackers using only hand-crafted features, ECOhc, BACF and MEEM obtain the top three evaluation scores. Although no deep features are used, their results are comparable with some deep trackers, such as MDNet and CFNet. From the evaluation results, we note that the highest AO score on GOT-10k only reaches 37.4%, compared to 74.0% on OTB2015. The SR scores also indicate that the best tracker only successfully tracks 40.4% of frames, suggesting that tracking in real-world unconstrained videos is difficult and still far from being solved.

By comparing evaluation results on GOT-10k and on OTB2015, we observe a significant change in the ranking of methods. For example, SRDCFdecon achieves top performance on OTB2015 but it performs much worse on GOT-10k. On the other hand, GOTURN obtains very low AO score on OTB2015 while it outperforms most trackers on GOT-10k. In addition, by comparing some methods with their improved versions, we can also observe the differences between OTB2015 and GOT-10k evaluation results. For example, ECO improves CCOT in several ways and it achieves a much better performance than CCOT on OTB2015, but on GOT-10k it's AO score is worse than CCOT. The same phenomenon can also be observed by comparing DSST and its improved version fDSST, and SRDCF and the improved tracker SRDCFdecon. The possible reason of such differences may be that some high-performance trackers are overfitted to small datasets, or they need certain amount of hyperparameter tuning to achieve better performance, while methods with straightforward frameworks may have better generalization ability in challenging scenarios.

The *Speed (fps)* column in Table 5 shows the tracking speeds of different approaches. Among the GPU trackers, GOTURN achieves the highest speed of 87 frames per second (fps), followed by SiamFC and CFNet. GOTURN and SiamFC benefit from their extremely simple architectures and tracking pipelines, while CFNet reformulates the efficient correlation filters in an end-to-end learnable module to achieve high-speed tracking. On the CPU platform, CSK is the fastest tracker that runs at around 109 fps, followed by IVT and KCF. The learning and inference efficiency of correlation filters plays a key role in the high speed of CSK and KCF, while the fast incremental subspace updating scheme contributes to the efficiency of IVT. Note the tracking speeds evaluated on GOT-10k are usually slower than their reported results on OTB or VOT. This is because that the video and object resolutions in GOT-10k dataset are much higher (3$\sim$9 times larger) than in OTB and VOT datasets.



(a) Impact of scale.



(b) Impact of diversity.

Fig. 10: Ablation study on how the scale and diversity of training data impact the performance of deep trackers. (a) Impact of training data scale on tracking performance. The number of videos is exponentially increased from 15 to 9335, with a multiplier of 5. (b) Impact of training data diversity on tracking performance. The number of training object classes is exponentially increased from 5 to 405, while motion classes from 4 to 64.

4.4 Evaluation by Challenges

Although the overall performance indicates the general quality of trackers, it cannot differentiate them according to different attributes and thus reflect the strength and weakness of each method. In this section, we analyze the performance of trackers on subsets of testing data which are labeled with different challenging attributes. Nevertheless, manual annotation of binary challenging attributes is quite subjective and thus inaccurate. For example, its hard to determine the boundaries of *object deformation*, *scale variation* or *fast motion*. Instead, similar to [9], we setup several informative indicators that can be directly computed from annotations. The indicators are defined as follows:

Occlusion. The occlusion indicator can be directly deduced from the labeling of visible ratios. We define occlusion indicator for a video as the percentage of frames where visible ratio $v \leq 0.6$.

Scale variation. The variation of object scales is measured by $\max_i s_i / min_i s_i$, where $s_i = \sqrt{w_i h_i}$ denotes the object size at ith frame.

Aspect ratio variation. Object deformation and rotation can be characterized by the variation of aspect ratios. We measure the range of aspect ratio variation in a video as $\max_i r_i / \min_i r_i$, where $r_i = h_i / w_i$.

Fast motion. We measure the object motion speed relative to its size as:

$$d_i = \frac{\|p_i - p_{i-1}\|_2}{\sqrt{s_i s_{i-1}}}, \tag{1}$$

where p_i denotes the object center location. The indicator of fast motion is defined as the average of d_i across all frames.

Illumination variation. The illumination variation in each frame can be measured by the change of average colors $u_i = \|c_i - c_{i-1}\|_1$, where c_i is the average object color at frame i. Then we compute the indicator of illumination variation as the average of u_i across all frames.

Small/Largs objects. Objects with very small or very large resolutions can affect the tracking performance. We first measure object size s for each video as the average of s_i across all frames, then we use the median of these sizes s^{median} to represent normal object size. The indicator of small/large object is defined as:

$$f(x) = \begin{cases} s/s^{median}, \text{if } s > s^{median}, \\ s^{median}/s, \textit{otherwise}. \end{cases} \tag{2}$$

According to the above indicators, we select the top 20% hardest videos (i.e., 36 videos) for each challenging attribute from testing data for analysis. Results are visualized in Figure 9. From the figure we observe that the most challenging attribute is *aspect ratio variation*, which causes 10%∼15% absolute drop in AO for most trackers. This indicates that tracking under object deformation and rotation is still difficult for current trackers. Another highly challenging factor is *fast motion*, where the evaluation results are around 7%∼10% lower than overall performance.

Among all baseline methods, GOTURN performs considerably better than other trackers on *aspect ratio variation* subset. By learning conditioned bounding box regressor, GOTURN is able to accurately locate the boundary of target when it deforms or rotates. It also helps GOTURN to distinguish "object" from "background", which improves its robustness on *occlusion* subset – when target reappears, GOTURN is more likely to recover the target instead of locating to background occluders. ECO achieves the best performance on *fast motion*, *illumination variation* and *small/large objects* subsets. ECO employs a very large search area (4.5 times target size) in both learning and tracking stages, enabling it to locate fast moving objects with a better chance. The rich feature hierarchy (HOG, shallow and deep layer outputs of CNNs) also improves the accuracy of ECO on different sizes of objects and under illumination variation.

4.5 Impact of Training Data

In this section, we discuss the impact of training data on the performance of deep trackers. We analyze two aspects of training data, the scale and the diversity, on three different deep trackers, namely SiamFCv2, GOTURN and MDNet. SiamFCv2 learns a matching function between exemplars and instances, GOTURN directly trains a conditional bounding box regressor, while MDNet learns a binary classifier to distinguish target from background patches. Figure 10 shows the performance of the three trackers w.r.t. different settings of training data.

Impact of scale. We train the three deep trackers with video number exponentially increases from 15 to 9335, with a multiplier of 5, and evaluate their tracking performance on the testing subset of GOT-10k. Results are displayed in Figure 10a. We surprisingly find that, the dependencies of different trackers on the scale of training data differ significantly. The performance of SiamFCv2 consistently improves as more videos are used for training, and the trend does not converge at 9335 videos. It seems that SiamFCv2 can benefit from even larger training data. GOTURN achieves higher AO scores at the start as more data are used, but the performance then drops when data scale gets larger, which may indicate a under-fitting. By contrast, MDNet seems to be insensitive to the scale of training data. Its performance is largely saturated at only 15 training videos, while further increasing training data has minor influence on its evaluation scores.

Impact of diversity. We fix the number of training videos to 2000 and exponentially vary the number of sampled object classes from 5 to 405, and motion classes from 4 to 64. The corresponding evaluation results are shown in Figure 10b. We observe that the performance of SiamFCv2 continuously improves as more training classes are included, and the trend approximately converges at the last point. The performance of GOTURN also shows an upward tendency with the improvement of dataset diversity. However, the impact of dataset diversity on these two trackers seems to be less significant than that of dataset scale. For MDNet, similar to scale, its performance is also insensitive to the number of classes included in the training data.

One possible reason for the different impacts of dataset scale and diversity on different trackers lies in the model capacity. Although all three deep trackers use shallow CNNs as their backbone, SiamFCv2 allows all parameters trainable, while MDNet and GOTURN fix the first three layers to pretrained weights. In this respect, SiamFCv2 has a larger model capacity and can fit training data with more videos and higher diversity. Instead, limited by their model capacities, it is difficult for GOTURN and MDNet to benefit from larger-scale datasets.

5 Conclusion

In this paper, we introduce GOT-10k, a large-scale tracking dataset with an unprecedentedly wide coverage of real-world moving objects. GOT-10k collects over 10,000 videos of 563 object classes, and annotates 1.5 million tight bounding boxes manually. We first describe the construction of GOT-10k, showing how the diversity and quality are ensured in our collection and annotation stages. Then we present the principle we follow and the analytical experiments we carry out for the establishment of an efficient and relatively unbiased evaluation platform for generic purposed trackers. Finally, we train and evaluate a number of recent tracking approaches on our dataset and analyze their results. We show the major challenges of generic object tracking in real-world unconstrained scenarios and discuss the impact of training data on tracking performance. We believe GOT-10k will provide a platform for training and principled evaluation of deep trackers, as well as guide research on generic object tracking.

References

[1] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[2] M. Kristan et al., The Visual Object Tracking VOT2017 Challenge Results, In IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, 2017, pp. 1949-1972.

[3] M. Kristan et al., The Visual Object Tracking VOT2016 Challenge Results. In European Conference on Computer Vision Workshops (ECCVW), Amsterdam, 2016. (Lecture Notes in Computer Science vol 9914. Springer, Cham)

[4] M. Kristan et al., The Visual Object Tracking VOT2015 Challenge Results, IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, 2015, pp. 564-586.

[5] M. Kristan et al., The Visual Object Tracking VOT2014 Challenge Results. In European Conference on Computer Vision Workshops (ECCVW), 2014.. (Lecture Notes in Computer Science, vol 8926. Springer, Cham)

[6] M. Kristan et al., "The Visual Object Tracking VOT2013 Challenge Results," 2013 IEEE International Conference on Computer Vision Workshops, Sydney, NSW, 2013, pp. 98-111.

[7] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In IEEE Conference on *Computer vision and pattern recognition*, pages 2411–2418. Ieee, 2013.

[8] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[9] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. Smeulders, P. Torr, and E. Gavves. Long-term tracking in the wild: A benchmark. *arXiv preprint arXiv:1803.09502*, 2018.

[10] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1134–1143. IEEE, 2017.

[11] S. Li and D.-Y. Yeung. Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models. In *AAAI*, pages 4140–4146, 2017.

[12] M. Mueller, N. Smith, and B. Ghanem. A benchmark and simulator for uav tracking. In *European Conference on Computer Vision*, pages 445–461. Springer, 2016.

[13] P. Liang, E. Blasch, and H. Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.

[14] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan. Nus-pro: A new visual tracking challenge. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):335–349, 2016.

[15] S. Song and J. Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In IEEE International Conference on *Computer Vision*, pages 233–240. IEEE, 2013.

[16] Mller, M., Bibi, A., Giancola, S., Al-Subaihi, S. and Ghanem, B. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In *arXiv preprint arXiv:1803.10794.*, 2018.

[17] Heng Fan, Liting Lin, Fan Yang, et al. LaSOT: A High-quality Benchmark for Large-scale Single Object Tracking. *arXiv preprint arXiv:1809.07845*, 2018.

[18] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: An experimental survey. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1442–1468, 2014.

[19] Liu, Qiao, and Zhenyu He. "PTB-TIR: A Thermal Infrared Pedestrian Tracking Benchmark." arXiv preprint arXiv:1801.05944 (2018).

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on *Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[22] H. Su, J. Deng, and L. Fei-Fei. Crowdsourcing annotations for visual object detection. In *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, volume 1, 2012.

[23] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *CoRR*, abs/1702.00824, 2017.

[24] Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 961-970).

[25] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S. and Suleyman, M. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[26] Milan, A., Leal-Taix, L., Reid, I., Roth, S., and Schindler, K.. MOT16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831, 2016.

[27] Leal-Taix, L., Milan, A., Reid, I., Roth, S., and Schindler, K.. Motchallenge 2015: Towards a benchmark for multi-target tracking. arXiv preprint arXiv:1504.01942, 2015.

[28] Geiger, A., Lenz, P., and Urtasun, R.. Are we ready for autonomous driving? the kitti vision benchmark suite. In IEEE Conference on Computer Vision and Pattern Recognition, 2012 (pp. 3354-3361).

[29] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1329–1338, 2017.

[30] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pages 2794–2802. IEEE Computer Society, 2015.

[31] Kang, K., Ouyang, W., Li, H., and Wang, X. Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp. 817-825.

[32] Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., et al. T-CNN: Tubelets with convolutional neural networks for object detection from videos. In *IEEE Transactions on Circuits and Systems for Video Technology.* 2017.

[33] Cheng, J., Tsai, Y. H., Hung, W. C., Wang, S., and Yang, M. H. Fast and Accurate Online Video Object Segmentation via Tracking Parts. In *arXiv preprint arXiv:1806.02323.* 2018.

[34] Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., and Yu, N. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *IEEE International Conference on Computer Vision.* pp. 4846-4855. 2017.

[35] Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning. Advances in Neural Information Processing Systems. 2016: 3630-3638.

[36] Fe-Fei, L. A Bayesian approach to unsupervised one-shot learning of object categories. In Proceedings of IEEE International Conference on Computer Vision, 2003, pp. 1134-1141.

[37] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2016.

[38] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[39] J. Choi, H. J. Chang, S. Yun, T. Fischer, Y. Demiris, J. Y. Choi, et al. Attentional correlation filter network for adaptive visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2017.

[40] M. Danelljan, G. Bhat, F. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *30th IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6931–6939. IEEE, 2017.

[41] M. Danelljan, G. Häger, F. Khan, and M. Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5*. BMVA Press, 2014.

[42] Zhang, Jianming, Shugao Ma, and Stan Sclaroff. MEEM: robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*. Cham, 2014.

[43] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1561–1575, 2017.

[44] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4310–4318, 2015.

[45] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1438, 2016.

[46] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*, pages 472–488. Springer, 2016.

[47] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2016.

[48] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer, 2016.

[49] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[50] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[51] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 749–758, 2015.

[52] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2012.

[53] Y. Li and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision*, pages 254–265. Springer, 2014.

[54] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3074–3082, 2015.

[55] C. Ma, X. Yang, C. Zhang, and M.-H. Yang. Long-term correlation tracking. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5388–5396. IEEE, 2015.

[56] H. Nam and B. Han. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4293–4302. IEEE, 2016.

[57] E. Park and A. C. Berg. Meta-tracker: Fast and robust online adaptation for visual object trackers. *arXiv preprint arXiv:1801.03049*, 2018.

[58] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2113–2120, 2015.

[59] J. Supancic III and D. Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 322–331, 2017.

[60] R. Tao, E. Gavves, and A. W. Smeulders. Siamese instance search for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1420–1429. IEEE, 2016.

[61] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5000–5008. IEEE, 2017.

[62] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3119–3127, 2015.

[63] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia. Understanding and diagnosing visual tracking systems. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 3101–3109. IEEE, 2015.

[64] T. Yang and A. B. Chan. Recurrent filter learning for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2010–2019, 2017.

[65] Gordon, Daniel, Ali Farhadi, and Dieter Fox. Re3: real-time recurrent regression networks for object tracking. In *arXiv preprint arXiv:1705.06368 2 (2017).*

[66] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi. Action-decision networks for visual tracking with deep reinforcement learning. In IEEE Conference on *Computer Vision and Pattern Recognition*, pages 1349–1358. IEEE, 2017.
[67] T. Zhang, C. Xu, and M.-H. Yang. Robust structural sparse tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
[68] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In IEEE Conference on *Computer vision and pattern recognition*, pages 1838–1845. IEEE, 2012.
[69] Li B, Yan J, Wu W, et al. High Performance Visual Tracking With Siamese Region Proposal Network, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 8971-8980.
[70] Z. Zhu, W. Wu, W. Zou, and J. Yan. End-to-end flow correlation tracking with spatial-temporal attention. *arXiv preprint arXiv:1711.01124*, 2017.
[71] Ross, D. A., Lim, J., Lin, R. S., and Yang, M. H. (2008). Incremental learning for robust visual tracking. International journal of computer vision, 77(1-3), 125-141.
[72] Bao, C., Wu, Y., Ling, H., and Ji, H. (2012, June). Real time robust l1 tracker using accelerated proximal gradient approach. In IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1830-1837.
[73] Zhang, T., Ghanem, B., Liu, S., and Ahuja, N. (2012, June). Robust visual tracking via multi-task sparse learning. In IEEE conference on Computer vision and pattern recognition (CVPR), 2012.
[74] Wang, Dong, Huchuan Lu, and Ming-Hsuan Yang. ”Online object tracking with sparse prototypes.” IEEE transactions on image processing 22.1 (2013): 314-325.
[75] Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010, June). Visual object tracking using adaptive correlation filters. In IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544-2550.
[76] Galoogahi, H. K., Fagg, A., and Lucey, S.. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA (pp. 21-26), 2017.
[77] Ren, S., He, K., Girshick, R. and Sun, J.. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99), 2015.
[78] Xiao, T., Li, H., Ouyang, W., and Wang, X.. Learning deep feature representations with domain guided dropout for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1249-1258.
[79] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
[80] Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., and Wang, S.. Learning dynamic siamese network for visual object tracking. In The IEEE International Conference on Computer Vision. (Oct 2017).
[81] Bhat, G., Johnander, J., Danelljan, M., Khan, F. S., and Felsberg, M. Unveiling the Power of Deep Tracking. arXiv preprint arXiv:1804.06833, 2018.
[82] Shi, J, and C. Tomasi. Good features to track. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, 2002.
[83] Zhang, Tianzhu, et al. Robust visual tracking via multi-task sparse learning. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition IEEE, 2012:2042-2049, 2012.

Lianghua Huang received his BSc from Dalian University of Technology (DUT) and MSc from Beijing Institute of Technology (BIT). In September 2017, he joined the Institute of Automation of the Chinese Academy of Sciences (CASIA), where he is currently studying for his doctorate. He has published 6 papers in the areas of computer vision and pattern recognition at international journals and conferences such as IEEE TIP, SMC Part B and ICME. His current research interests include pattern recognition, computer vision and machine learning.

Xin Zhao received the B.E. degree from Anhui University of Technology (AHUT) in 2006 and Ph.D. degree from University of Science and Technology of China (USTC) in 2013. In July 2013, he joined the Institute of Automation of the Chinese Academy of Sciences (CASIA), where he is currently an Associate Professor. Dr. Zhao has published over 20 research papers in the areas of computer vision and pattern recognition at international journals and conferences such as CVIU, CVPR, ICCV, IJCAI, ICPR, ICIP and ACCV. In 2011, he won the IAPR (the International Association of Pattern Recognition) Best Student Paper Award. His current research interests include pattern recognition, computer vision and machine learning.

Kaiqi Huang received his BSc and MSc from Nanjing University of Science Technology, China and obtained his PhD degree from Southeast University, China. He has been full professor in National Lab. of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science (CASIA) since 2005. He is IEEE Senior member and deputy general secretary of IEEE Beijing Section (2006-2008). His current researches focus on computer vision and pattern recognition including object recognition, video analysis and visual surveillance and so on. He has published over 180 papers in the important international journals and conferences such as IEEE TPAMI, T-IP, T-SMCB, TCSVT, Pattern Recognition (PR), CVIU, ICCV, ECCV, CVPR, ICIP, ICPR. He received the winner prizes of the object detection tasks in both PASCAL VOC’10 and PASCAL VOC’11, object classification task in ImageNet-ILSVRC2014, He serves as co-chairs and program committee members over forty international conferences such as ICCV, CVPR, ECCV, IEEE workshops on visual surveillance and so on. He is an executive team member of IEEE SMC Cognitive Computing Committee as well as Associate Editor of IEEE Trans on Systems, Man, and Cybernetics: Systems (TSMCs) and Pattern Recognition (PR). He has obtained some awards including CCF-IEEE Young Scientist Award (2016), the Excellent Young Scholars Award of National Science Foundation of China (NSFC) (2013), The National Science an Technology Progress Award (2011).