# TracKlinic: Diagnosis of Challenge Factors in Visual Tracking

Heng Fan[1] Fan Yang[2] Peng Chu[2] Yuewei Lin[3] Lin Yuan[4] Haibin Ling[1]
[1]Department of Computer Science, Stony Brook University, Stony Brook, NY, USA
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, PA USA
[3]Computational Science Initiative, Brookhaven National Laboratory, Upton, NY, USA
[4]Amazon Web Services, Palo Alto, CA USA

## Abstract

*Generic visual object tracking is difficult due to many challenge factors (*e.g.*, occlusion, blur, etc.). Each of these factors may cause serious problems for a tracker, and when they work together can make things even more complicated. Despite a great amount of efforts devoted to understanding the behavior of trackers, reliable and quantifiable ways for studying the per factor tracking behavior remain barely available. Addressing this issue, in this paper we contribute to the community a tracking diagnosis toolkit,* TracKlinic*, for diagnosis of challenge factors of tracking algorithms.*

*TracKlinic consists of two novel components focusing on the data and analysis aspects, respectively. For the data component, we carefully prepare a set of 2,390 annotated videos, each involving one and only one major challenge factor. When analyzing an algorithm for a specific challenge factor, such* one-factor-per-sequence *rule greatly inhibits the disturbance from other factors and consequently leads to more faithful analysis. For the analysis component, given the tracking results on all sequences, it investigates the behavior of the tracker under each individual factor and generates the report automatically. With TracKlinic, a thorough study is conducted on ten state-of-the-art trackers on nine challenge factors (including two compound ones). The results suggest that, heavy* shape variation *and* occlusion *are the two most challenging factors faced by most trackers. Besides,* out-of-view*, though does not happen frequently, is often fatal. By sharing TracKlinic[1], we expect to make it much easier for diagnosing tracking algorithms, and to thus facilitate developing better ones.*

## 1. Introduction

As one of the most fundamental problems in computer vision, visual tracking has been studied for several decades. Despite considerable progress in recent years, object tracking remains difficult due to many challenge factors[2] such as *occlusion*, *rotation*, *background clutter*, *out-of-view*, *motion blur*, etc. Any one of these factors may cause severe variation in target shape and/or appearance in a video, resulting in tracker failures. Moreover, multiple factors may occur at the same time, making things even worse.

Numerous approaches have been proposed to tackle the aforementioned challenge factors. To validate effectiveness, evaluating and comparing these trackers on datasets [42, 43, 7, 31, 20, 19, 27, 23, 39, 17, 36, 30] is an important section. Doubtlessly, these benchmarks greatly advance the tracking community. Nevertheless, if used directly, they are incapable of faithfully investigating tracker behaviors on specific factors, as described below.

In existing datasets, a sequence is typically involved with multiple challenge factors (see Fig. 1 (a) for an example), and these factors are usually annotated for the entire sequence instead of for individual frames. In such case, it is hard to *faithfully* examine a tracker on a specific factor due to disturbances from other ones. Likewise, *reliably* comparing different trackers with respect to a factor is hard, because the trackers may fail due to other factors instead of the one to assess. For instance, when comparing two trackers $\mathcal{T}_A$ and $\mathcal{T}_B$ on a sequence with *background clutter* and *occlusion*, assuming it only contains these two factors and *background clutter* occurs earlier, if $\mathcal{T}_A$ outperforms $\mathcal{T}_B$, can we conclude that $\mathcal{T}_A$ performs better than $\mathcal{T}_B$ in dealing with *occlusion*? Obviously, drawing such a conclusion would be arbitrary since $\mathcal{T}_B$ may fail where *background clutter* happens and it never has a chance to demonstrate its ability in handling the subsequent *occlusion*.

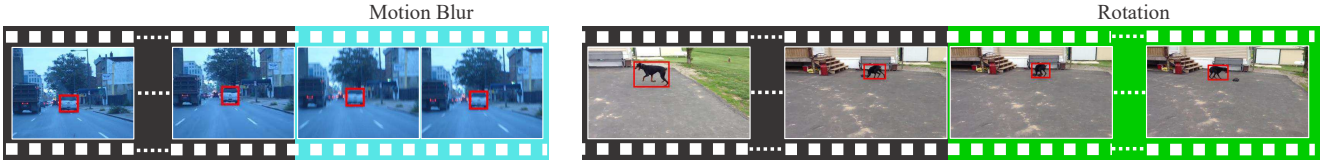
To conduct reliable and quantifiable *per factor* analysis for tracking algorithms, a new type of benchmark is desired, which ideally should meet the following two requirements:

(1) **Purity**. Different from sequences in existing tracking datasets with many challenge factors, each sequence of the desired benchmark should be involved with one and only

[1]TracKlinic is released at `https://hengfan2010.github.io/projects/TracKlinic/TracKlinic.htm`.

[2]Note that, *challenge factor* is often called *attribute* as well [42].


(a) A sequence in existing benchmark with multiple various challenge factors.




(b) Two sequences in our TracKlinic with each containing only one major factor.

Figure 1. Comparison between existing tracking benchmark LaSOT [7] and our diagnosis benchmark TracKlinic. Note that, the *per frame* factor annotations in image (a) are labeled by us, and the original benchmark only provides global factor annotations. Best viewed in color.

one major factor (*i.e.*, *one-factor-per-sequence*). By doing so, we are able to eliminate influences from others when examining tracking algorithms on a specific challenge factor, leading to more faithful analysis.

(2) **Quantity**. Evaluation and analysis of trackers on a benchmark are desired to be general. To this end, a dataset should be large scale to guarantee the statistical significance of analysis, so is the desired dataset. In particular, the desired benchmark should contain *at least* a decent number (30 is used in our work) of videos for each challenge factor for statistically meaningful analysis [22].

### 1.1. Contribution

Motivated by the above discussion, we introduce a novel toolkit, referred to as *TracKlinic*, aiming to offer an in-depth understanding of visual tracking algorithms on each challenge factor. TracKlinic consists of two components focusing on data and analysis aspects, respectively.

**Data Component.** At the core of the proposed TracKlinic is an elaborately designed benchmark, which is generated by first *per frame* labeling 461 videos of three benchmarks (OTB-2015 [43], TC-128 [27] and LaSOT$_{\text{tst}}$ [7]) with seven challenge factors. In total, 776K frames are labeled with 7$\times$776K annotations. With these annotations, we construct the TracKlinic diagnosis benchmark by extracting eligible (sub-)sequences from the 461 annotated ones. Eventually, TracKlinic consists of 2,390 sequences with 280K frames. Each sequence in TracKlinic consists of *one and only one* major factor[3], which is essentially different from existing benchmarks (see Fig. 1 (b)). By doing so, we can reliably examine a tracker on a specific factor by eliminating noisy influences from others, providing guidance for pertinent improvements in future research. Besides, TracKlinic allows more fair comparison of trackers on these factors.

[3]In practice, there is a small chance that a sequence still suffers from some unidentified factors, but statistically negligible in our study.

The annotation above involves seven *simple* challenge factors, including *Occlusion*, *Background Clutter*, *Illumination Variation*, *Motion Blur*, *Out-of-View*, *Rotation* and *Shape Variation*. We observe that, not surprisingly, some factors often accompany with each other, and may be worth being investigated together. Thus, we include into TracKlinic two *compound* challenge factors, *Occlusion with Background Clutter* and *Occlusion with Rotation*, both of which are seriously under-studied in previous work. Finally, nine challenge factors are included in TracKlinic.

**Analysis Component.** For the analysis aspect, we present a diagnosis tool. Given the tracking results of all sequences in TracKlinic, this tool analyzes the behavior of a tracker on each individual factor and generates a report automatically. In specific, we demonstrate failure distribution of a tracker and examine its robustness to each challenge factor. In this way, we can better understand the strengths and weaknesses of a tracker, allowing of pertinent improvement. In addition, we conduct more reliable comparisons of trackers on each factor by erasing noise caused by irrelevant ones. Moreover, we measure the consistency of a tracker to different factors to validate its stability.

Integration of the above two components forms our novel diagnosis toolkit TracKlinic. We exemplify its use and capability on ten state-of-the-art trackers. Our analysis reveals that, heavy *shape variation* and *occlusion* are the two most challenging factors faced by most trackers. In addition, *out-of-view*, though does not occur frequently, it is often fatal to visual trackers, and effective strategies should be presented to handle it. More analysis is detailed in later sections.

In summary, our contributions are two-fold:

**(i)** We propose the novel TracKlinic for investigating per factor tracking behaviors. Our toolkit offers the community a diagnosis dataset consisting of 2,390 sequences with 280K frames. To the best of our knowledge, TracKlinic is the first benchmark for diagnosing challenge factors in tracking. In addition, an analysis tool is presented to study trackers and

generate report automatically.

**(ii)** We apply our TracKlinic to ten state-of-the-art trackers, which serves as an example of how researchers can diagnose their own trackers. We include detailed analysis for the exemplification purpose, beyond any insights observed from the diagnosis results.

## 2. Related Work

**Visual Tracking Benchmarks** Benchmarks have played a crucial role in advancing the tracking research. For a long time, tracking algorithms are usually evaluated with a small number of sequences, resulting in the problem of subjective bias [32]. Addressing this issue, various tracking benchmarks are proposed, such as OTB [42, 43], TC-128 [27], VOT series [18], NUS-PRO [23], NfS [17], UAV123 [30], CDTB [28] and ALOV [36]. Recently, to provide enough data for training deep trackers, larger-scale datasets have been proposed, including OxUvA [39], GOT-10K [14], LaSOT [7] and TrackingNet [31]. These benchmarks greatly advance the research of visual tracking. Nevertheless, as discussed above, they are not suitable for reliable diagnosis of tracking algorithms regarding specific challenge factors.

Different from these benchmarks, the proposed TracKlinic is dedicated for challenge factor diagnosis. For this goal, *per frame* challenge factor annotation is first provided on initial sequences. Moreover, one-factor-per-sequence is enforced in TracKlinic for reliable analysis. These characteristics make TracKlinic uniquely suitable for the diagnosis purpose, and enable us, for the first time (to our knowledge), to perform per factor analysis.

It is worth noting that, TracKlinic does not aim to replace existing tracking benchmarks. Instead, it can be viewed as complimentary to existing datasets to focus on challenge factor analysis.

**Diagnosis of Tracking Algorithms** Another important related work is [40]. In [40], a diagnosis approach is proposed to analyze the impact of each tracking component on final tracking performance. Specifically, five constitution parts including motion model, feature extractor, observation model, model updater and ensemble post-processor are examined using different implementations, aiming to offer guidance to researchers in designing robust trackers.

Our work is significantly different from [40]. The work of [40] focuses on studying the impact of a specific tracking component on tracking performance, while ours examines the abilities of a tracker on different challenge factors. In fact, the diagnosis approach of [40] and this work are complementary and can work together to provide directions for improving tracking performance.

**Diagnosis in Other Vision Problems** Our work is also closely related to similar studies [13, 47, 34, 35, 1, 44] for other tasks. The seminal work of [13] explores different failure modes in object detection and showcases the impacts of different characters on detection performance. The approach of [47] analyzes localization errors for pedestrian detection. The methodology of [34] provides an insightful diagnosis of different algorithms in the field of action understanding in videos and discusses relevant directions for future improvements. The method of [35] introduces the diagnosis analysis into human pose estimation with the goal of quantitatively identifying the failure modes of different algorithms and recommending effective strategies for improvements in body part localization. In [1], a diagnostic tool is presented to characterize errors for temporal action localization. The work of [44] diagnoses deep learning models for age estimation.

In a similar spirit with these studies, we present a diagnosis approach to characterize different challenge factors in tracking, and thus help researchers to understand the performance of a tracker on specific challenge factors.

## 3. TracKlinic

### 3.1. Diagnosis Dataset

In order to validate effectiveness of a tracker in handling a specific challenge factor, it is essential to inhibit disturbance from others in videos. However, the sequences in existing datasets are often involved with more than one challenge factor (see Fig. 1 (a)). In such situation, it is not clear which factor is the cause of a tracking failure. To address this issue, we propose TracKlinic, which is the first diagnosis dataset in the tracking community.

#### 3.1.1 Initial Collection

Our goal is *not* to build a completely new dataset for assessing overall performance of a tracker as in existing datasets. Instead, we aim to diagnose an algorithm on different challenge factors to further improve its performance on existing benchmarks. Considering this, it is natural to adopt existing benchmarks as our data source.

Since this work is focused on generic model-free single-object tracking, we narrow down our choices to OTB-2015 [43], TC-128 [27], NUS-PRO [23], VOT series [18], GOT-10K [14], LaSOT [7] and TrackingNet [31]. With a joint consideration of target diversity, groundtruth availability and the amount of required annotation, we choose OTB-2015 [43], TC-128 [27] and LaSOT$_{\text{tst}}$ [7] as our data source. For conciseness, we use OTL to denote these three benchmarks. In this way, we obtain 461 videos with 770K frames in OTL for initial collection of TracKlinic.

#### 3.1.2 Per-frame Factor Annotation

Tracking is a complex process, and it is difficult to enumerate all challenge factors that may result in tracking failures. In this paper, we study the seven most common factors in tracking, including *Occlusion* (OCC), *Background Clutter*

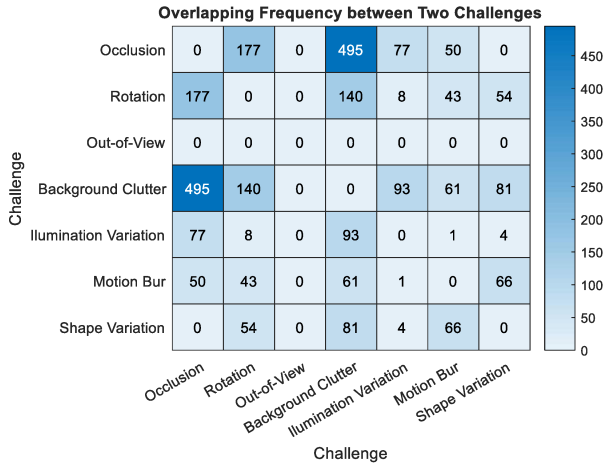



Figure 2. The statistic of overlapping frequency between challenge factors in OTL. Best viewed in color.

Table 1. Descriptions of nine challenge factors in OTL.

| Challenge Factor | Definition |
|---|---|
| Occlusion (OCC) | Target is occluded in a frame |
| Rotation (ROT) | Target rotates in a frame |
| Out-of-View (OV) | Target completely leaves the video frame |
| Bkg. Clut. (BC) | The background is similar to target |
| Illumination Vari. (IV) | The illumination in target region changes |
| Motion Blur (MB) | Target region is blurred due to motion |
| Shape Variation (SV) | The ratio of bounding box or its aspect ratio is outside the range [0.25, 4] |
| Occ.-Bkg. Clut. (O-B) | Target suffers from overlapping OCC and BC |
| Occ.-Rot. (O-R) | Target suffers from overlapping OCC and ROT |

(BC), *Illumination Variation* (IV), *Motion Blur* (MB), *Out-of-View* (OV), *Rotation* (ROT) and *Shape Variation* (SV)[4].

Unlike conventional tracking benchmarks in which each video is labeled with global challenge factors, our goal is to build a new type of dataset where each sequence is involved with *one and only one* major factor. For this purpose, we need to label each video in OTL with *per frame* challenge factors. Based on these annotations, we extract eligible subsequences from OTL and ensure that each one qualifies the *one-factor-per-sequence* rule.

To guarantee the annotation quality, each frame is manually labeled with the aforementioned challenge factors (except for SV) by an expert and then verified by another expert. The annotation of SV is automatically computed based on its definition. It is worth noticing that, since we focus on diagnosing challenges in object tracking, a frame is labeled with challenge factors only when obvious target appearance variations are caused by them. In total, we finish 7×776K manual annotations for all sequences in OTL. An example of our annotation for a video is shown in Fig. 1 (a) and more video examples can be seen in **supplementary material**.

In OTL, each challenge factor represents a type of challenge (referred to as *simple-challenge* factor) for tracking. However, it is rather frequent in tracking that two or more factors occur simultaneously. Considering this, we also study challenge factors containing two types of challenges (referred to as *compound-challenge* factor). In specific, we count the overlapping frequency between two challenges in OTL. Note that, in order to be considered as overlapping, the number of overlapping frames of two challenges has to be greater than $\tau_{\text{op}}$ (set to 3). Fig. 2 demonstrates the statistical results. In this work, we select the two most frequent compound-challenge factors, *i.e.*, *Occlusion with Background clutter* (O-B) and *Occlusion with Rotation* (O-R). After that, we automatically generate annotations for these two compound-challenge factors and meanwhile adjust annotations for OCC, BC and ROT.

In summary, we diagnose nine factors for tracking algorithms. The definitions of these factors are shown in Tab. 1.

#### 3.1.3 Single-factor Sequence Extraction

With the per-frame annotations of each challenge factor, we extract single-factor sequence from OTL under *one-factor-per-sequence* rule. These eligible videos form our diagnosis dataset. We describe the extraction rules as follows.

For extraction purpose, we classify challenge factors into two types $\mathrm{T}_1$ and $\mathrm{T}_2$ based on the status of last frame in a challenge. If the last frame suffers from occlusion or out-of-view, the challenge factor belongs to $\mathrm{T}_1$, otherwise $\mathrm{T}_2$. For challenge factor of $\mathrm{T}_1$, we extract subsequences which satisfy: the subsequence can be decomposed into three consecutive parts $s^1_{\text{nc}}$, $s^2_{\text{c}}$ and $s^3_{\text{nc}}$, where $s^1_{\text{nc}}$ and $s^3_{\text{nc}}$ do not contain any challenge factors and $s^2_{\text{c}}$ exclusively contains this factor, and the lengths $|s^1_{\text{nc}}| \geq \tau_{\text{s}}$ (set to 10) and $|s^3_{\text{nc}}| = \tau_{\text{e}}$ (set to 2), as shown in Fig. 3 (a). Considering that long sequence may cause more cumulative errors, we only retain the last 30 frames in $s^1_{\text{nc}}$ if $|s^1_{\text{nc}}| > 30$ for all factors except for shape variation, because 30 frames are sufficient for stable tracker initialization. For challenge factor of $\mathrm{T}_2$, we extract subsequences that satisfy: the subsequence can be divided into two consecutive parts $s^1_{\text{nc}}$ and $s^2_{\text{c}}$, where $s^1_{\text{nc}}$ does not contain any challenge factors and $s^2_{\text{c}}$ exclusively contains this factor, and the length $|s^1_{\text{nc}}| \geq \tau_{\text{s}}$, as shown in Fig. 3 (b). Likewise, we only retain the last 30 frames in $s^1_{\text{nc}}$ if $|s^1_{\text{nc}}| > 30$. The difference between $\mathrm{T}_1$ and $\mathrm{T}_2$ is that, for challenge factor of $\mathrm{T}_1$, the target may be partly visible (*e.g.*, occlusion) or completely invisible (*e.g.*, out-of-view) in the last frame. In such case, it is not suitable to estimate target status. Therefore, we add an extra part $s^3_{\text{nc}}$ to $\mathrm{T}_1$ type factor to ensure that the target is fully visible.

After the above procedure, we construct TracKlinic with each sequence involving *one and only one* major factor. We show examples for each factor in **supplementary material**. Tab. 2 summarizes TracKlinic and comparisons with other datasets. Fig. 4 shows the numbers of videos for each factor.

[4]It is worth noting that, since the *Deformation* and *Aspect Ratio Change* almost always accompany with *Scale Variation*, we unify all these three challenge factors into *Shape Variation*.


(a) Subsequence for $T_1$ factor type (b) Subsequence for $T_2$ factor type

Figure 3. Generating subsequences of TracKlinic for challenge factors of types $T_1$ and $T_2$. Best viewed in color and by zooming in.

Table 2. Comparison of TKB and other tracking Benchmark. OTL combines OTB-2015 [43], TC-128 [27] and LaSOT$_{tst}$ [7].

| Benchmark | Videos | Min frames | Mean frames | Max frames | Total frames | Chal. factor anno. |
|---|---|---|---|---|---|---|
| VOT-18 [20] | 60 | 41 | 356 | 1,500 | 21K | n/a |
| UAV123 [30] | 123 | 109 | 915 | 3,085 | 113K | global |
| NfS [17] | 100 | 169 | 3,830 | 20,665 | 383K | global |
| GOT-10K$_{tst}$ [14] | 180 | 51 | 127 | 920 | 23K | n/a |
| TrackingNet$_{tst}$ [31] | 511 | 96 | 441 | 2,368 | 226K | n/a |
| OTB-2015 [43] | 100 | 71 | 590 | 3,872 | 59K | global |
| TC-128 [27] | 129 | 71 | 429 | 3,872 | 55K | global |
| LaSOT$_{tst}$ [7] | 280 | 1,000 | 2448 | 9,999 | 690K | global |
| OTL | 461 | 71 | 1683 | 9,999 | 776K | per frame |
| TracKlinic | 2,390 | 11 | 115 | 6,559 | 280K | per frame |

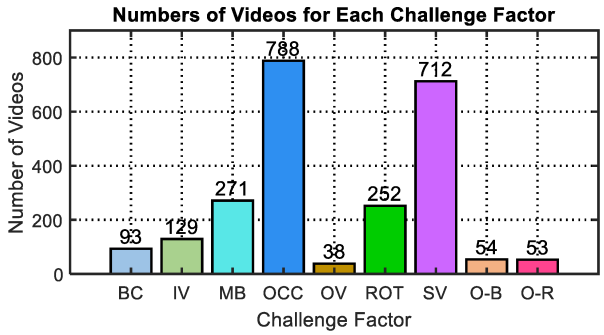



Figure 4. Numbers of sequences for different challenge factors in TracKlinic. There are *at least* 30 videos for each challenge factor, indicating that our diagnosis analysis is statistically meaningful.

### 3.2. Analysis Tool

In addition to the benchmark, an analysis methodology is presented in our toolkit for studying trackers on each factor.

In specific, we demonstrate the proportion of failures due to challenge factors for trackers, which allows us to understand the potential failure causes. In addition, we study the failure rates of trackers for each factor. In this way, we can better understand the strengths and weaknesses of trackers for future improvements. Furthermore, we compare different tracking algorithms on our benchmark. In comparison with other datasets, our analysis is more reliable because noise caused by irrelevant factors are erased. We also measure consistency of a tracker to different factors.

Given the tracking results of all sequences in TracKlinic, our tool analyzes the above per factor tracking behavior and generates the report automatically, with which researchers can further improve their own trackers.

## 4. Diagnosis of Challenge Factors

### 4.1. Algorithms

We apply TracKlinic to ten state-of-the-art trackers. In specific, we choose the top three trackers with available implementations on each of the four datasets, including VOT-19 [21], LaSOT [7], OTB-2015 [43] and TC-128 [27]. We exclude the repeated ones from these trackers. In this way, we obtain eight algorithms, including DiMP [3], ATOM [5], SiamRPN++ [24], SiamDW [49], ASRCF [4], ECO [6], D-STRCF [26] and GFS-DCF [45]. In addition, we also analyze two baseline trackers KCF [12] and SiamFC [2] as these two trackers have led the recent trends of tracking with many extensions [29, 37, 8, 25, 38, 10, 48, 9], and it is worth diagnosing them on different challenge factors.

### 4.2. Proportion of Failures by Challenge Factors

It is important to understand failures of a tracker caused by different challenge factors for improvements. In this section, we take ten trackers as an example of showcasing the proportions of failures by each factor.

Since each video in TracKlinic is involved with one major challenge factor, we define tracking failure on a video according to the tracking result in the last frame. If the intersection over union (IoU) of tracking result with groundtruth is less than a threshold $\tau_{\text{IoU}}$ (set to 0.5), the tracker fails for the challenge factor in this video. Note that, although each video contains one major factor, it is possible that a tracker fails before the challenge in a video happens. For this case, we identify the tracking failure caused by others.

Fig. 5 shows the proportions of failures by different challenge factor. Surprisingly, we observe that for all ten trackers, although shape variation does not happen the most frequently in videos (see Fig. 4), it is the most likely to cause a

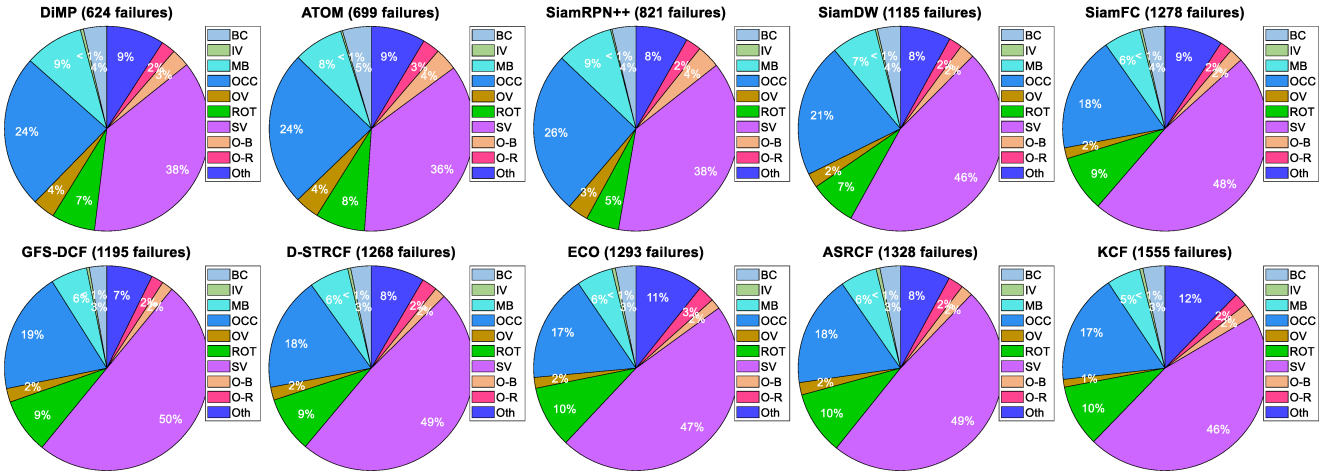



Figure 5. The percents of failures caused by different challenge factors for ten trackers. Best viewed in color and by zooming in.

failure. Specifically, the probabilities of a failure caused by the shape variation are 38%, 36%, 38%, 46%, 48%, 50%, 49%, 47%, 49% and 46% for DiMP, ATOM, SiamRPN++, SiamDW, SiamFC, GFS-DCF, D-STRCF, ECO, ASRCF and KCF, respectively.

Following shape variation, occlusion is second most likely to cause a tracking failure, with chances of 24%, 24%, 26%, 21%, 18%, 19%, 18%, 17%, 18% and 17% for the ten trackers. Note that, shape variation and occlusion account for more than 60% of failures, suggesting that strategies to better handle them may be the most rewarding for future research. Moreover, we can see from Fig. 5 that, illumination variation is the least possible reason for a failure. For all trackers, the chances of a failure caused by illumination variation do not exceed 1%, indicating that future research can focus more on dealing with other factors such as shape variation, occlusion, rotation, motion blur and so forth.

### 4.3. Understanding Trackers on Different Factors

To understand the strengths and weaknesses of a tracker, it helps (1) guide future researches for improvement and (2) deploy the tracker in practical applications as we can choose the tracking algorithms according to the specific scenarios. In this section, we show the abilities of ten representative trackers in response to each challenge factor. We use failure rate, defined as the ratio of the number of tracking failures and the number of videos of a challenge factor, to represent the ability of the tracker. The lower the failure rate is, the stronger the tracker is in handling the challenge factor.

Fig. 6 shows the failure rates of ten trackers on each factor. An interesting observation is that, although out-of-view appears least frequently (see Fig. 4), it is fatal to trackers. For DiMP, ATOM, SiamRPN++, SiamDW, SiamFC, GFS-DCF, D-STRCF, ECO, ASRCF and KCF, their failures rates under the out-of-view are 65.8%, 71.1%, 78.9%, 81.6%, 86.2%, 86.8%, 89.5%, 84.2%, 89.5% and 86.8%, respectively. It is worth noting that DiMP, ATOM, SiamRPN++ and SiamDW utilize very deep networks [11] for feature representation. Nevertheless, they still easily fail when out-of-view happens, which implies the requirement of a special mechanism in handling out-of-view for tracking. A feasible solution is to couple the tracker with a global detector [16, 46] which can re-locate the target when it appears again in the view.

In addition, shape variation is difficult to visual trackers, especially for the baseline SiamFC and correlation filter based ones including GFS-DCF, D-STRCF, ECO, ASRCF and KCF. The failure rates of these trackers are higher than 80% due to the lack of an effective strategy to estimate target state. This issue is alleviated by borrowing techniques such as RPN [33] (*i.e.*, SiamRPN++ and SiamDW) and IoU-Net [15] (*i.e.*, DiMP and ATOM) from object detection. Despite this, a tracker is prone to fail when shape variation occurs. Especially for non-rigid targets, the bounding-box based trackers may introduce a large amount of background information into the tracking model, resulting in gradual degradation. To achieve accurate target state estimation, a better way is to leverage the mask of target through segmentation [41]. Although the chance of an occlusion-causing failure is high, trackers perform surprisingly robustly to this challenge factor. In particular, using deeper feature can further reduce the failure rate. Nevertheless, when accompanied with other factors such as rotation or background clutter, occlusion becomes more challenging. Furthermore, we also see that all visual trackers exhibit robust performance with failure rates of 3.1%, 2.3%, 2.3%, 3.1%, 4.7%, 4.7%, 5.4%, 7.8%, 6.4 and 8.5% in tackling illumination variation, which indicates that the illumination variation issue is almost addressed using deep features.

In order to qualitatively analyze each tracker, we demonstrate qualitative results of common failure cases for each tracker on eight difficult factors in Fig. 7.

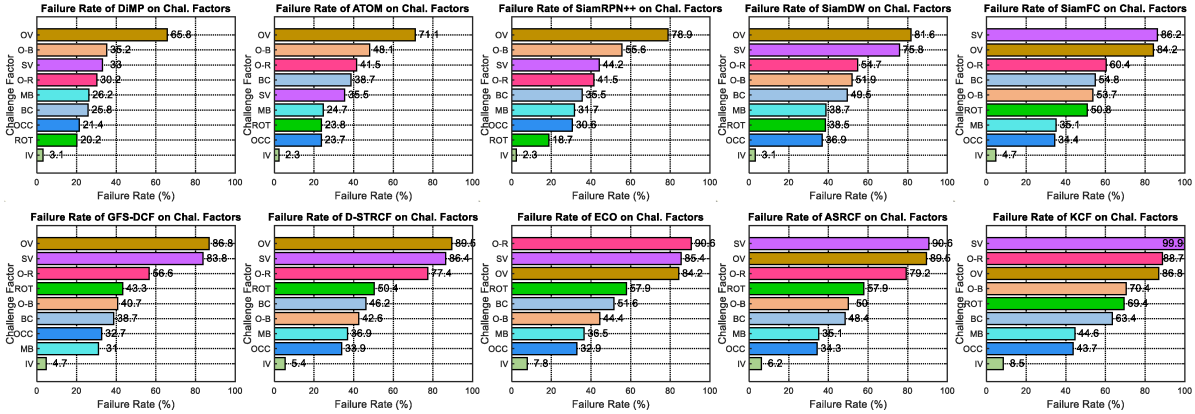
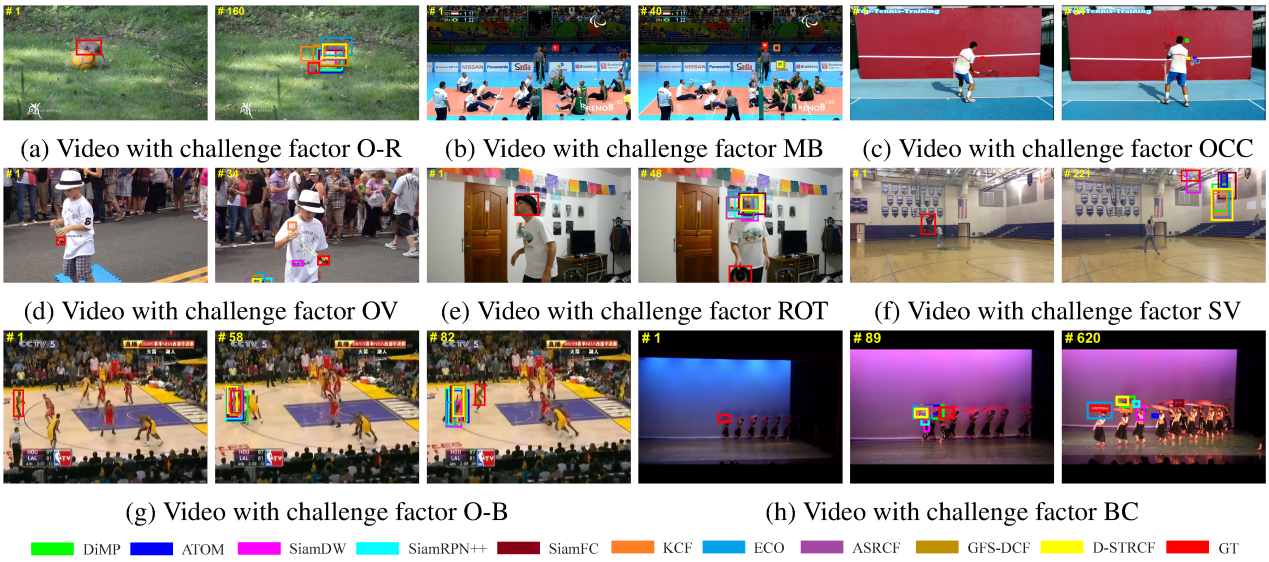



Figure 6. Failure rates of ten state-of-the-art trackers for each challenge factor. Best viewed in color and by zooming in.



(a) Video with challenge factor O-R (b) Video with challenge factor MB (c) Video with challenge factor OCC

(d) Video with challenge factor OV (e) Video with challenge factor ROT (f) Video with challenge factor SV

(g) Video with challenge factor O-B (h) Video with challenge factor BC

DiMP ATOM SiamDW SiamRPN++ SiamFC KCF ECO ASRCF GFS-DCF D-STRCF GT

Figure 7. Qualitative results of failure cases for each tracker under eight challenge factors. Best viewed in color.

### 4.4. Challenge Factor-based Comparison

In addition to diagnosis of each challenge factor in tracking, our TracKlinic allows more reliable comparison of different approaches for each challenge factor. We adopt success score (at the threshold of 0.5) [42] as evaluation metric.

Fig. 8 demonstrates the challenge factor-based comparison of different algorithms. We observe that DiMP achieves the best performance under eight out of nine challenge factors. In particular, compared with its baseline ATOM, the improvement under background clutter is obvious, which evidences the effectiveness of exploring background information in distinguishing target from distractors. For shape variation, the three trackers DiMP, ATOM and SiamRPN++ significantly outperform others, showing the importance of deeper features and the advantages of bounding box regression and IoU prediction in target state estimation. Besides, an interesting finding is that, as the two most popular approaches for state estimation, our analysis shows that IoU-Net based strategy performs slightly better than RPN based method. For illumination variation, all trackers show robust performance, which is consistent with previous diagnosis.

### 4.5. Consistency Analysis

In real-world sceneries, it is difficult to estimate the complexities of challenge factors in a video. Therefore, a practical algorithm should demonstrate consistent (good) performance no matter how complicated the factor is. For this purpose, we conduct consistency analysis for each tracker. In detail, we adopt the variance of all success scores for consistency measurement of a tracker to a challenge factor. The smaller the variance is, the more consistent the performance is. Notice that, the consistency of a tracker may be high if it performs poorly on all factors. Thus, consistency analysis should be applied with other metrics such as success score.

Fig. 9 demonstrates the consistency of each tracker on different factors. We observe that deep trackers performs

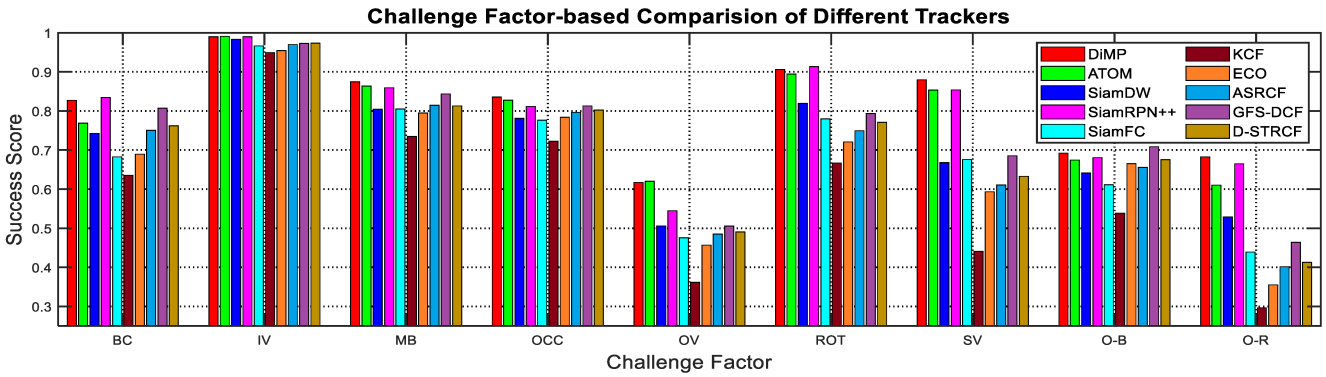



Figure 8. Challenge Factor-based comparison of ten state-of-the-art trackers. Best viewed in color.

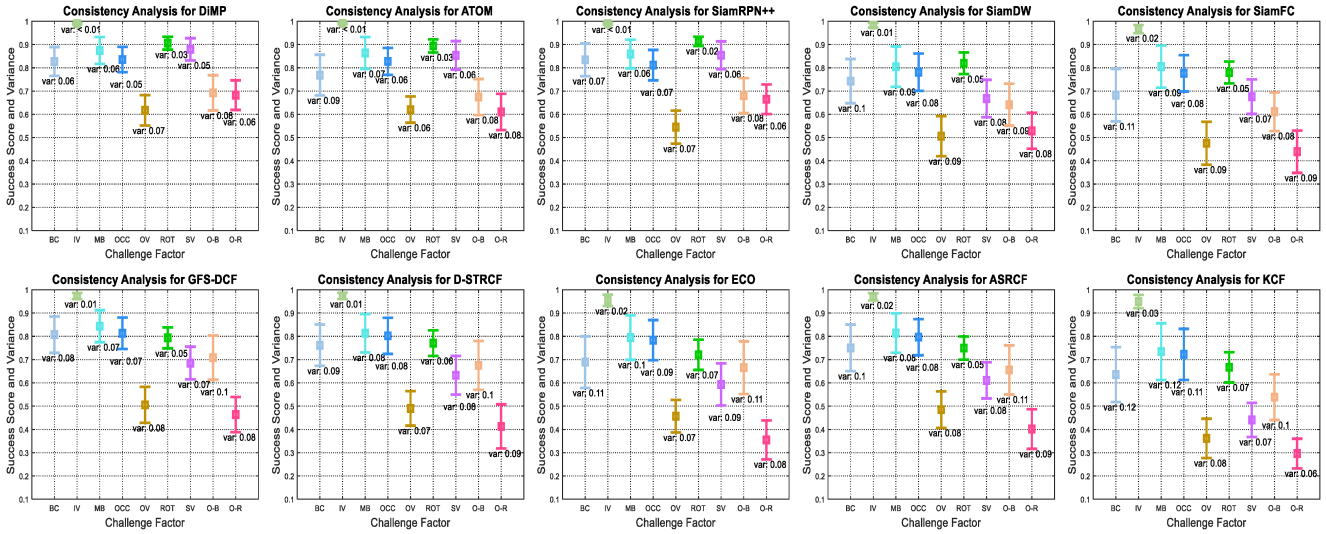



Figure 9. Consistency measurement to each challenge factor. The square represents the average success score of a tracker. The difference between the top (or bottom) and the square denotes the variance under the challenge factor. Best viewed in color.

consistently on illumination variation. However, the performance of trackers on out-of-view and background clutter is inconsistent due to their extreme complexities. For instance for out-of-view, the target may re-enter the view from a distant position. In such case, trackers easily fail. In a video with complicated background clutter, there may exist many distractors with identical appearances to the target. As a result, trackers may drift to other similar distractors. In summary, more efforts are needed to improve the consistency to challenge factors to make visual trackers practical.

## 5. Conclusion

In this work, we present a novel diagnosis toolkit, TracKlinic, for studying per factor tracking behaviors and exemplify its use on ten state-of-the-art trackers. We show how TracKlinic helps identify potential challenge factors for a tracker. Our results suggest that, heavy shape variation and occlusion are worth more attentions in future research. In addition, we examine the ability of a tracker in handling different challenge factors, which is crucial in understanding the strengths and weaknesses of a tracker. We investigate the failure rate of a tracker to each factor and observe that some rare factors such as out-of-view and occlusion with ration are fatal to visual trackers. We measure consistency of a tracker to different challenge factors and show that more efforts should be made to improve the stability of a tracker. By releasing TracKlinic, we aim to empower the community with more in-depth understanding of tracking algorithms, beyond a single scalar metric evaluation. Most importantly, we expect that our diagnostic tool inspires the development of innovative tracking models that address the issues in current ones.

**Acknowledgment**. This work is supported in part by NSF Grants IIS-2006665 and IIS-1814745. This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research under contract number DE-SC0012704.

## References


[1] Humam Alwassel, Fabian Caba Heilbron, Victor Escorcia, and Bernard Ghanem. Diagnosing error in temporal action detectors. In *ECCV*, 2018. 3

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCVW*, 2016. 5

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *ICCV*, 2019. 5

[4] Kenan Dai, Dong Wang, Huchuan Lu, Chong Sun, and Jianhua Li. Visual tracking via adaptive spatially-regularized correlation filters. In *CVPR*, 2019. 5

[5] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *CVPR*, 2019. 5

[6] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017. 5

[7] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019. 1, 2, 3, 5

[8] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *ICCV*, 2017. 5

[9] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *CVPR*, 2019. 5

[10] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *CVPR*, 2018. 5

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

[12] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2014. 5

[13] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV*, 2012. 3

[14] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv:1810.11981*, 2018. 3, 5

[15] Borui Jiang, Ruixuan Luo, Jiayuan Mao, Tete Xiao, and Yuning Jiang. Acquisition of localization confidence for accurate object detection. In *ECCV*, 2018. 6

[16] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *TPAMI*, 34(7):1409–1422, 2011. 6

[17] Hamed Kiani Galoogahi, Ashton Fagg, Chen Huang, Deva Ramanan, and Simon Lucey. Need for speed: A benchmark for higher frame rate object tracking. In *ICCV*, 2017. 1, 3, 5

[18] 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. *TPAMI*, 38(11):2137–2155, 2016. 3

[19] Matej Kristan et al. The visual object tracking vot2017 challenge results. In *ICCVW*, 2017. 1

[20] Matej Kristan et al. The sixth visual object tracking vot2018 challenge results. In *ECCVW*, 2018. 1, 5

[21] Matej Kristan et al. The seventh visual object tracking vot2019 challenge results. In *ICCVW*, 2019. 5

[22] Erich Leo Lehmann. *Elements of large-sample theory*. Springer Science & Business Media, 2004. 2

[23] Annan Li, Min Lin, Yi Wu, Ming-Hsuan Yang, and Shuicheng Yan. Nus-pro: A new visual tracking challenge. *TPAMI*, 38(2):335–349, 2015. 1, 3

[24] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *CVPR*, 2019. 5

[25] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018. 5

[26] Feng Li, Cheng Tian, Wangmeng Zuo, Lei Zhang, and Ming-Hsuan Yang. Learning spatial-temporal regularized correlation filters for visual tracking. In *CVPR*, 2018. 5

[27] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *TIP*, 24(12):5630–5644, 2015. 1, 2, 3, 5

[28] Alan Lukezic, Ugur Kart, Jani Kapyla, Ahmed Durmush, Joni-Kristian Kamarainen, Jiri Matas, and Matej Kristan. Cdtb: A color and depth visual object tracking dataset and benchmark. In *ICCV*, 2019. 3

[29] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *ICCV*, 2015. 5

[30] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *ECCV*, 2016. 1, 3, 5

[31] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018. 1, 3, 5

[32] Yu Pang and Haibin Ling. Finding the best from the second bests-inhibiting subjective bias in evaluation of visual tracking algorithms. In *ICCV*, 2013. 3

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

[34] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *ICCV*, 2017. 3

[35] Gunnar A Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017. 3

[36] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *TPAMI*, 36(7):1442–1468, 2013. 1, 3

[37] Yuxuan Sun, Chong Sun, Dong Wang, You He, and Huchuan Lu. Roi pooled correlation filters for visual tracking. In *CVPR*, 2019. 5

[38] Jack Valmadre, Luca Bertinetto, João Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, 2017. 5

[39] Jack Valmadre, Luca Bertinetto, Joao F Henriques, Ran Tao, Andrea Vedaldi, Arnold WM Smeulders, Philip HS Torr, and Efstratios Gavves. Long-term tracking in the wild: A benchmark. In *ECCV*, 2018. 1, 3

[40] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. Understanding and diagnosing visual tracking systems. In *ICCV*, 2015. 3

[41] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 6

[42] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *CVPR*, 2013. 1, 3, 7

[43] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *TPAMI*, 37(9):1834–1848, 2015. 1, 2, 3, 5

[44] Junliang Xing, Kai Li, Weiming Hu, Chunfeng Yuan, and Haibin Ling. Diagnosing deep learning models for high accuracy age estimation from a single image. *Pattern Recognition*, 66:106–116, 2017. 3

[45] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Joint group feature selection and discriminative filter learning for robust visual object tracking. In *ICCV*, 2019. 5

[46] Bin Yan, Haojie Zhao, Dong Wang, Huchuan Lu, and Xiaoyun Yang. 'skimming-perusal'tracking: A framework for real-time and robust long-term tracking. In *ICCV*, 2019. 6

[47] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. How far are we from solving pedestrian detection? In *CVPR*, 2016. 3

[48] Yunhua Zhang, Lijun Wang, Jinqing Qi, Dong Wang, Mengyang Feng, and Huchuan Lu. Structured siamese network for real-time visual tracking. In *ECCV*, 2018. 5

[49] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *CVPR*, 2019. 5