IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 13, 2021, accepted December 23, 2021, date of publication January 31, 2022, date of current version February 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3145799

Quality Aware Features for Performance Prediction and Time Reduction in Video Object Tracking

ROGER GOMEZ-NIETO[1], (Member, IEEE), JOSÉ FRANCISCO RUIZ-MUÑOZ[2], JUAN BERON[1], CÉSAR A. ARDILA FRANCO[1], HERNÁN DARÍO BENÍTEZ-RESTREPO[1], (Senior Member, IEEE), AND ALAN C. BOVIK[3], (Fellow, IEEE)

[1]Departamento de Electrónica y Ciencias de la Computación, Pontificia Universidad Javeriana, Seccional Cali, Cali 760014, Colombia
[2]Universidad Nacional de Colombia, Sede de La Paz, Cesar 202017, Colombia
[3]Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA

Corresponding author: Roger Gomez-Nieto (roger.gomez@javerianacali.edu.co)

This work was supported by the MINCIENCIAS Colombia and the Pontificia Universidad Javeriana Seccional Cali through the Project Vigilancia Inteligente para la red de cámaras de la Policía Metropolitana de Cali under Grant FP44842-035-2017.

ABSTRACT The existing body of work on video object tracking (VOT) algorithms has studied various image conditions such as occlusion, clutter, and object shape, which influence video quality and affect tracking performance. Nonetheless, there is no clear distinction between the performance reduction caused by scene-dependent challenges such as occlusion and clutter, and the effect of authentic in-capture and post-capture distortions. Despite the plethora of VOT methods in the literature, there is a lack of detailed studies analyzing the performance of videos with authentic in-capture and post-capture distortions. We introduced a new dataset of authentically distorted videos (AD-SVD) to address this issue. This dataset contains 4476 videos with different authentic distortions and surveillance activities. Furthermore, it provides benchmarking results for evaluating ten state-of-the-art visual object trackers (from VOT 2017-2018 challenges) based on the proposed dataset. In addition, this study develops an approach for performance prediction and quality-aware feature selection for single-object tracking in authentically distorted surveillance videos. The method predicts the performance of a VOT algorithm with high accuracy. Then, the probability of obtaining the reference output is maximized without executing the tracking algorithms. We also propose a framework to reduce video tracker computation resources (time and video storage space). We achieve this by balancing processing time and tracking accuracy by predicting the performance in a range of spatial resolutions. This approach can reduce the execution time by up to 34% with a slight decrease in performance of 3%.

INDEX TERMS Video object tracking, in-capture and post-capture distortions, video quality assessment, video tracking prediction.

I. INTRODUCTION

Video object tracking (VOT) is one of the most studied areas in computer vision and multimedia processing. VOT is a complex computational process that makes possible to locate and follow one object over time using video streaming. This can be applied in human-computer interaction, robotics, and video surveillance.

Recently, several studies and competitions have proposed different VOT approaches [1]–[4]. Nonetheless, to the best of our knowledge, these approaches have not modeled and quantified the influence of post-capture and in-capture distortions on video object tracking performance. Post-capture distortion refers to those quality impairments (i.e., blur, compression artifacts, and additive noise) introduced synthetically after capturing the video. By contrast, in-capture distortions are authentic quality impairments or combinations such as overexposure, low exposure, and defocus aberration acquired during the video capture. State-of-the-art trackers usually

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

perform well on videos with few or no distortions. However, the performance of VOT algorithms can decrease when they are tested on videos affected by authentic distortions such as lack of exposure, over-exposure, and defocus.

A. LITERATURE REVIEW

The correlational filter (CF) technique has been used successfully in several VOT algorithms, such as DeepSTRCF [5], [6], owing to its low computational cost and speed. From the first frame, the object is tracked by correlating the filter in regions close to the detected object across the following frames [7]. The correlation is carried out in the Fourier domain using the Fast Fourier Transform (FFT) to increase the tracker speed. CF can be divided into four categories according to their components [8]: categorized features [9], space weight factors [10]–[12], scale factors [13], and expert strategies [14]. Yuan *et al.* [15] propose the self-supervised deep correlation tracker (self-SDCT) that exploits internal correlations by using a Siamese network and generating pseudo-labels of consecutive frames. Also, as an alternative to CF-based VOT algorithms, Chen *et al.* [16] introduce an attention-based feature fusion network to learn long-term relationships, named TransT inspired by the transformer architecture. TransT outperforms several state-of-the-art tracking algorithms on six challenging datasets.

Previous studies on the impact of distortions on the performance of machine vision algorithms have addressed tasks such as object and face detection [17], dermoscopy [18], and face recognition in long-wave infrared (LWIR) images [19]–[21]. These approaches are usually based on natural scene statistics (NSS) or deep relevant quality features that account for post-capture distortions such as blur, additive noise, and uneven illumination. In addition, most existing literature on the effect of distortions on VOT performance [22]–[24] deals with post-capture distortions, such as blurring caused by shaking motion [25], and deblurring [26].

In [27], the authors proposed a method for the robustness measurement of VOT algorithms based on accuracy rate and performance stability. The performance of ten existing visual tracking algorithms was evaluated by the proposed assessment method, using the Quality-degraded Video Database for Visual Tracking (QDVD-VT). This resource contains videos affected by post-capture distortions such as compression, contrast changes, resolution variation, white noise, and frame rate changes. This study concludes that it is challenging to track objects in distorted videos using the tested visual tracking algorithms.

In [26], Guo *et al.* present a benchmark dataset containing 500 videos with different levels of motion blurs for 100 scenes. They evaluated 25 tracking algorithms on this dataset and group them into four classes according to the representations used: i) intensity-based features [28]–[32], ii) HoG Features [5], [10], [33]–[37], iii) Deep Features [9], [38]–[45], and iv) mixed features [46]–[48]. The authors in [26] concluded that the light motion blur improved in most of the trackers, while heavy blur significantly decreased their accuracy. Similarly, they studied the effects of two state-of-the-art deblurring methods [49], [50], concluding that deblurring can improve tracking accuracy on heavily blurred videos while having little effect on those with light blur impairments. Finally, they proposed a new GAN-based tracking scheme that adopts the fine-tuned discriminator DeblurGAN [49] as an adaptive blur assessor to selectively deblur frames, improving the accuracy of six state-of-the-art trackers [5], [32], [36], [38], [40], [46]. However, the blur distortions studied in [26], despite being called ‘‘realistic’’ blur in some studies [51], [52], are properly classified as post-capture distortion because they were added after capturing the video. Furthermore, based on the findings of [26], it remains an open question whether deblurring methods as [53], [54] could improve the performance on tracking tasks.

B. CONTRIBUTIONS

To the best of our knowledge, this is the first work that introduces a database dedicated to modeling the effects of authentic in-capture distortions on VOT. The proposed dataset is Authentically Distorted Surveillance Videos Dataset (AD-SVD) and contains 4376 authentically distorted videos with different visual content and activities.

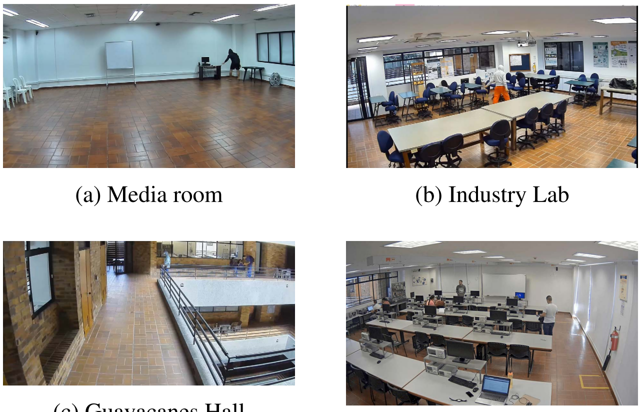
In addition, we developed a framework to predict the performance of VOT algorithms on authentically distorted videos and reduce video tracker computation resources. These include execution time and the disk space required for storage by predicting the VOT algorithm performance and determining the optimal spatial scale to process a video. Furthermore, this approach complements our previous work in which we demonstrated the impact of authentic distortions on state-of-the-art video trackers and developed a quality-aware-tracker for post-capture distortions [55], [56].

The remainder of this paper is organized as follows: Section II presents the proposed AD-SVD dataset and the benchmarking of video trackers, Section III describes the details of our video tracker performance prediction method, Section IV discusses our proposed method for video tracker execution time reduction, Section V analyzes the experimental results, and Section VI concludes the paper.

II. AUTHENTICALLY DISTORTED SURVEILLANCE VIDEOS DATASET

Because a similar resource is not available, we created an Authentically Distorted Surveillance Videos Dataset (AD-SVD) acquired by four different surveillance cameras (VIVOTEK IP8165HP, VIVOTEK IB8367A, VIVOTEK IB8381, AXIS P14), and affected by several levels of in-capture distortions. This dataset is publicly available at IEEE DataPort.[1] It contains 4476 videos recorded at three outdoor and four indoor locations, containing a variety of activities as shown in Figures 1, 2, and 3. Written informed consent was obtained from all participants.

[1] https://ieee-dataport.org/open-access/authentically-distorted-surveillance-videos-dataset



(a) Media room

(b) Industry Lab

(c) Guayacanes Hall

(d) Electronics Lab

FIGURE 1. Indoor locations.

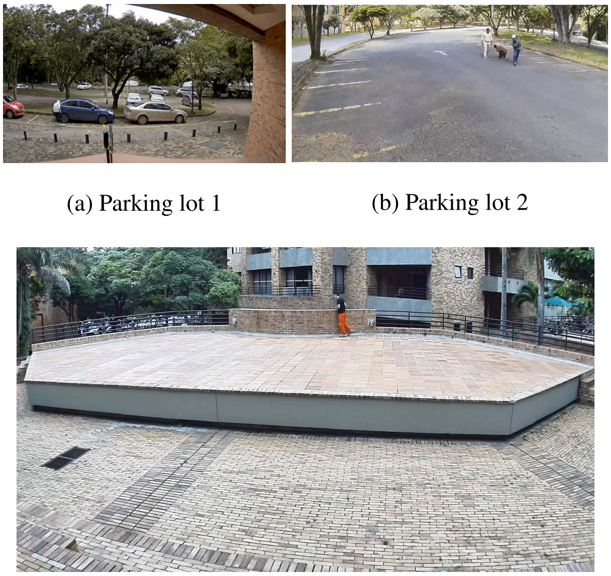
Table 1 shows the number of videos grouped according to condition, location, and activity. AD-SVD contains the activities listed below [57]:

- **Fighting in Group (FG):** 3 or 4 people fighting each other.
- **Leaving Package in a Public Place (LPP):** A person leaving a suspicious package in a public place.
- **Passing Out (PO):** A person who faints.
- **Person Pushing Person (PPP):** A person is pushing another person.
- **Person Running (PR):** A person is running in a closed loop.
- **Prowl (PW):** A person makes suspicious movements in search of something or someone.
- **Robbery with Knife (RK):** Simulation of robbery with a knife where a person assaults another person.
- **Walking (WL):** A person is walking in a closed loop.

Several datasets are currently used to evaluate the VOT algorithms. Figure 4 summarizes the characteristics of 11 of the most commonly used video datasets. At the same time, Table 2 presents the average number of video frames and the number of videos per dataset. *LaSOT* [63], *VOT2020-LT* [2] and *UAV20L* [60] are used for long-term tracking (LTT) evaluation. By contrast, AD-SVD was created for short-term tracker (STT) assessment. AD-SVD and *LaSOT* [63] stand out among the other benchmarks for their number of videos and frames. To the best of our knowledge, AD-SVD is the largest, densely annotated, and authentically distorted video object tracking benchmark for STT. Even though recent works on benchmarking of thermal VOT algorithms have been proposed in [67]–[69] that address challenges such as real-world scenarios along with deformable and blurry targets, in this work we focus on authentically distorted visible light surveillance videos.

A. BOUNDING BOX ANNOTATIONS

In AD-SVD, each video has an associated *.txt* file containing per frame annotations. The notation used is $[x, y, w, h]$, where x and y are the coordinates of the upper-left corner of the rectangle (*Bounding Box*), w is the width and h is its height. The labeling process relies on the *DarkLabel* tool [70]

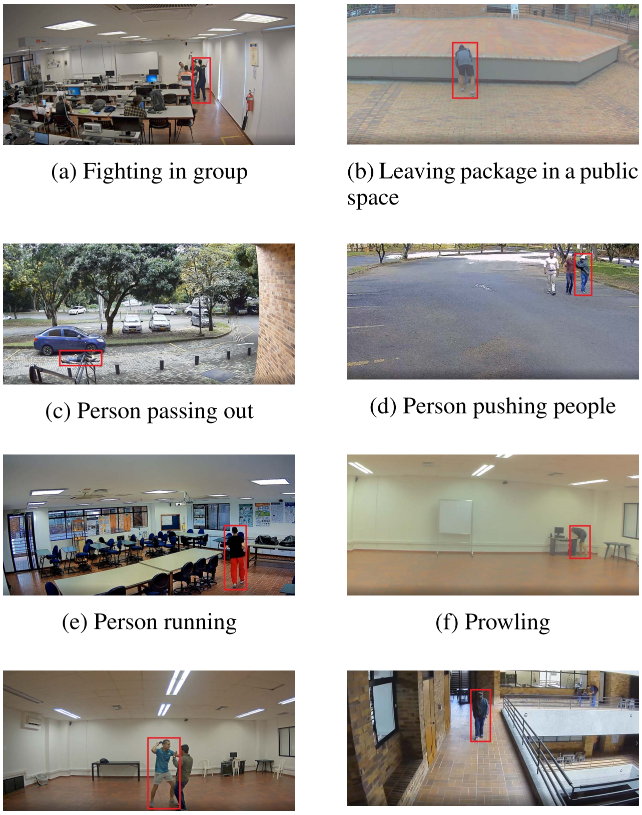


(a) Parking lot 1

(b) Parking lot 2

(c) Theater

FIGURE 2. Outdoor locations.



(a) Fighting in group

(b) Leaving package in a public space

(c) Person passing out

(d) Person pushing people

(e) Person running

(f) Prowling

(g) Robbing with knife

(h) Person walking

FIGURE 3. Activities recorded in AD-SVD dataset.

TABLE 1. AD-SVD specifications: number of videos per condition, scenario, and activity.

Condition	# Videos
Pristine	160
Exposure	1457
Defocus	1409
Mixed	1450
Total Videos	**4476**
Location	**# Videos**
1. Theater	856
2. Parking Lot 2	41
3. Parking Lot 1	801
4. Media Room	851
5. Industrial Lab	889
6. Guayacanes Hall	152
7. Electronics Lab	886
Total Videos	**4476**
Activity	**# Videos**
FG	546
LPP	564
PO	571
PPP	558
PR	549
PW	567
RK	562
WL	559
Total Videos	**4476**

TABLE 2. Tracking benchmarks summary.

Benchmark	Av. Video Length	# Videos
TrackingNet [65]	456	30643
Got-10k [66]	150	10000
AD-SVD	**450**	**4476**
LaSOT [63]	2500	1400
NUS-PRO [62]	542	365
TC128 [61]	429	129
UAV123 [60]	915	123
OTB100 [59]	590	100
VOT2020-ST [2]	332	60
OTB50 [59]	578	51
VOT2020-LT [2]	4306	50
VOT2014 [58]	416	25
UAV20L [60]	2934	20

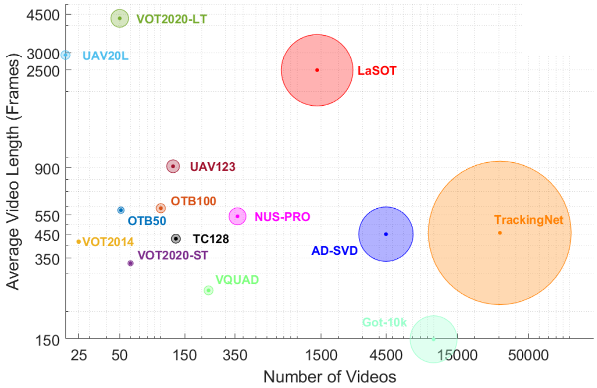


FIGURE 4. VOT benchmarks with high quality dense (per frame) annotations, including VOT-2014 [58], VOT-2020ST [2], VOT-2020LT [2], OTB50 [59], OTB100 [59], UAV20L [60], UAV123 [60], TC128 [61], NUS-PRO [62], LaSOT [63], VQUAD [64], TrackingNet [65], Got-10k [66] and AD-SVD. The circle diameter is proportional to the number of frames in a benchmark. The proposed AD-SVD has a higher number of videos than the other VOT datasets, except by TrackingNet and Got-10K.

annotating every five frames. An interpolator algorithm was used to obtain all the labels for each frame in the video, based on those five-frames annotations. Since AD-SVD evaluates STT algorithms, the region of interest (ROI) is present throughout the entire sequence, and each frame is labeled.

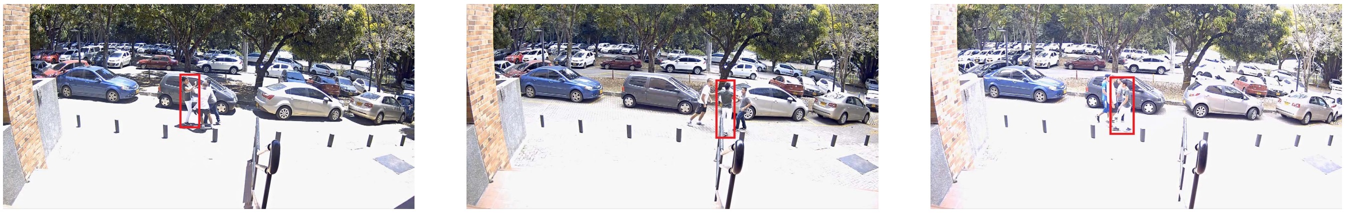
B. AUTHENTIC VIDEO DISTORTIONS

The authentic distortions affecting the recorded videos are defocus aberration (Defocus), over-exposure, sub-exposure (Exposure), and a combination of Defocus and Exposure, hereafter referred to as Defocus+Exposure. We selected these authentic impairments because they allow us to analyze different distortion levels (which is more difficult with other impairments such as color or artifacts). Figures 5, 6 and 7 illustrate distortions at three levels. We exported distorted videos (according to the compression standard H.264) into three different qualities: 100%, 75%, and 50%. Each distortion is categorized as low (1), medium (2), or high (3).

Since the configuration of the distortion levels in the four video cameras is not identical, and there are different brands and models, we used the No-Reference Video Quality Assessment (VQA) metric V-BLIINDS [71] to test the consistency of the parameter settings of the cameras used to record the AD-SVD videos. Figure 8 shows the box plots of V-BLIINDS values on the AD-SVD and the VOT 2018 datasets. We randomly selected 892 videos (20% of the total number of videos 4476) in the AD-SVD dataset to carry out this analysis. We chose this reduced set because V-BLIINDS is computationally expensive. Defocus, Exposure, and Defocus+Exposure distortions and pristine videos were represented by 283, 292, 284, and 33 videos, respectively. The higher the V-BLIINDS values, the worse the perceptual visual quality of the video. We observe that perceptual quality decreases (V-BLIINDS scores increase) in the following order: pristine, Exposure, defocus, and combined Exposure, and defocus distortions, as expected. Videos affected by exposure distortions exhibit more considerable variability in V-BLIINDS scores for AD-SVD. By contrast, V-BLIINDS [71] scores of videos affected by defocus and commingled defocus and exposure impairments show minor variance.

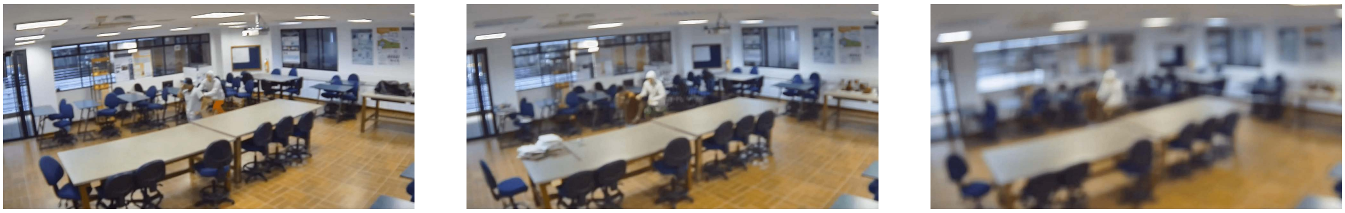
C. BENCHMARKING OF VIDEO OBJECT TRACKERS

Despite the plethora of competitive VOT methods presented in contests such as VOT 2017 [72] and 2018 [4], there is a lack of detailed studies analyzing performance on videos with authentic in-capture and post-capture distortions. To conduct



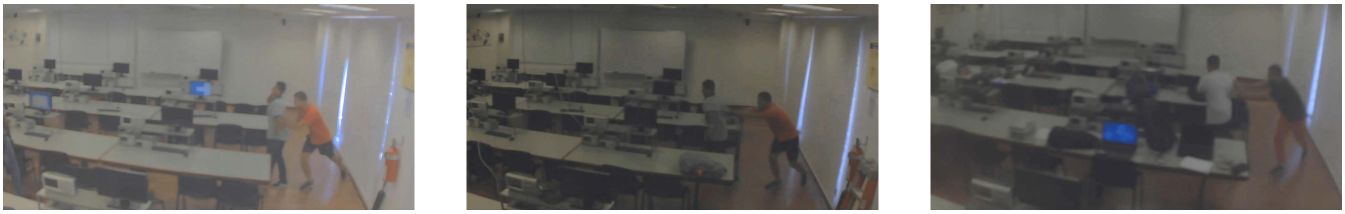
(a) Low exposure level (b) Medium exposure level (c) High exposure level

FIGURE 5. Examples of frames with different exposure levels.



(a) Low defocus level (b) Medium defocus level (c) High defocus level

FIGURE 6. Defocus levels.



(a) Low defocus+exposure (b) Medium defocus+exposure (c) High defocus+exposure level

FIGURE 7. Defocus+Exposure levels.

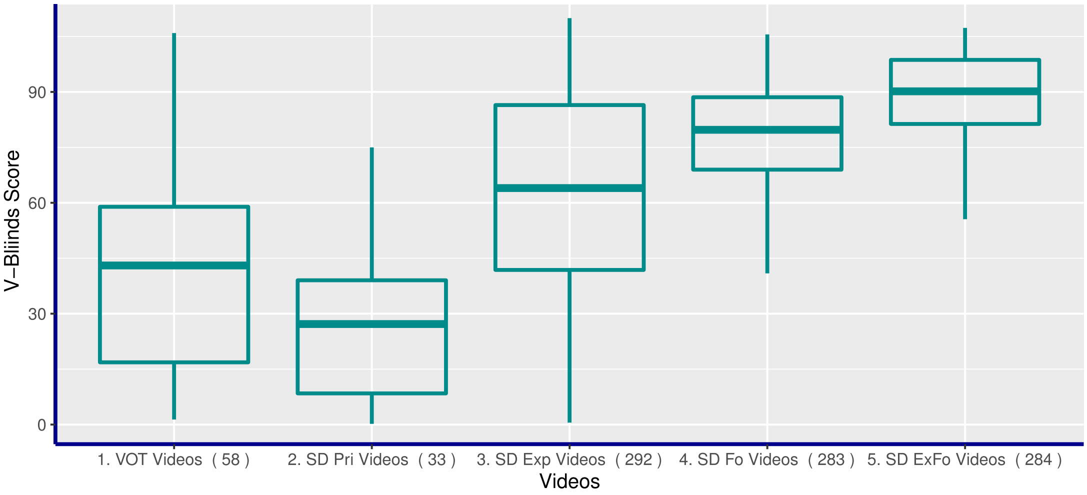


FIGURE 8. V-BLIINDS distributions on AD-SVD and VOT 2018 datasets, where the V-BLIINDS score is inversely proportional to the video quality.

this study, we selected 10 of the best algorithms (fast trackers with publicly available source code) of the VOT short-term challenge: three taken from the 2017 contest, seven selected from the 2018 contest, and one additional tracker scale DL-SSVM [73]. Table 3 describes the algorithms chosen along with the VOT ranking and features.

TABLE 3. Evaluated trackers.

Tracker	Contest	VOT Ranking	Features[2]
CFWCR [74]	VOT 2017	2	CNN/ECO
Gnet [75]	VOT 2017	5	CNN/DCF
MCCT [76]	VOT 2017	6	DCF
LADCF [77]	VOT 2018	1	DCF/HOG
MFT [78]	VOT 2018	2	DCF/ECO
RCO [4]	VOT 2018	5	DCF/CNN
DeepSTRCF [5]	VOT 2018	7	DCF/CNN
CPT [79]	VOT 2018	8	DCF/VGG
SRCT [80]	VOT 2018	12	SRB/ECO
CPT_fast [79]	VOT 2018	14	DCF/VGG
Scale DLSSVM [73]	-	-	MSE/LK

The proposed AD-SVD dataset is unique because it contains authentic in-capture and post-capture distortions in controlled levels in videos with surveillance activities. Figure 9 shows the performance of seven state-of-the-art trackers (LADCF [77], MFT [78], DeepSTRCF [5], CPT [79], DLST [81], RCO [4], and SRCT [80]) in 8 selected videos, at the original scale, in the AD-SVD dataset.

We measured the performance of the 11 trackers in the AD-SVD using the success rate, defined as the percentage of frames with an overlap greater than θ_i, where θ_i is the overlap threshold. The success rate for different values of θ is called the success plot as shown in Figure 10

The area under the curve (AUC) (i.e., area under the success plot) of each tracker allows us to understand the algorithm attaining a higher percentage of successful matches ($Overlap > \theta_i$) as the threshold increases. The higher the AUC, the more accurate the video tracker. Table 4 organizes the trackers according to their AUCs. In line with the AUC of each algorithm, the SRCT [80] tracker achieved the best performance, while the Scale DLSSVM [73] tracker yielded the worst performance. The winners of both contests (VOT 2017 [72] and VOT 2018 [4]) did not perform as well as the trackers with lower ranks. These results demonstrate the impact of authentic distortions on VOT performance. The best-performing tracker in AD-SVD (SRCT [80]) uses a combination of Salient Region-Based (**SRB**) and Efficient Convolution Operators (**ECO** [38]) techniques, which have also been the basis of other state-of-the-art video trackers.

High-ranked VOT algorithms in the VOT 2018 and 2017 challenges such as LADCF [77], MFT [78], and CFWCR [74] are based on discriminative correlation filters. These algorithms do not outperform the others in AD-SVD. Their feature representation relies on hand-crafted (HOG and color features) and deep features extracted from shallow layers such as conv-3 in VGG network that provide spatial information instead of semantic information. By contrast, SRCT, MCCT, and CPT_fast ranked high in AD-SVD but relied on salient-region modeling and discriminative correlation filters. Nonetheless, their feature representation is based on an object shape model or deeper layers (conv-3 VGG) encoding richer semantic features than shallower layers. We hypothesize that deep convolutional features and hand-crafted features representing object semantic traits properly encode a valuable representation, making VOT algorithms more robust with respect to authentic video distortions.

TABLE 4. Evaluated trackers in AD-SVD: AUC.

Tracker	AUC	AD-SVD Ranking	VOT Ranking
SRCT [80]	0.5069	1	12 (2018)
MCCT [76]	0.4988	2	6 (2017)
CPT_fast [79]	0.4936	3	14 (2018)
DeepSTRCF [5]	0.4897	4	7 (2018)
CPT [79]	0.4757	5	8 (2018)
CFWCR [74]	0.4611	6	2 (2017)
Gnet [75]	0.4577	7	5 (2017)
LADCF [77]	0.4408	8	1 (2018)
MFT [78]	0.4287	9	2 (2018)
RCO [4]	0.4267	10	5 (2018)
S. DLSSVM [73]	0.4139	11	-

The metrics robustness and accuracy are used to evaluate tracker performance per distortion. Robustness R is the number of times the tracker failed and had to be reinitialized. A video tracker fails (and a reinitialization is triggered) when the overlap ϕ_i (Eq. 1) drops to 0. A_t^G and A_t^T are the areas of the ground truth and detected target, respectively. The failure rate F_k increases with each reinitialization. R is the probability that the tracker will still successfully track the object up to the S frames from the last failure. Once the complete video sequence is evaluated, R (Eq. 2) is calculated, assuming a uniform failure distribution that does not depend on previous failures [82]. Accuracy A in Eq. 3 is the average overlap over all the frames in a video sequence [4], [82], where the number of frames is N_{frames}.

$$\phi = \frac{|A_t^G \cap A_t^T|}{|A_t^G \cup A_t^T|} \tag{1}$$

$$R_k = e^{(\frac{-SF_k}{N_{frames}})} \tag{2}$$

$$A = \frac{1}{N_{frames}} \sum_{i=1}^{N_{frames}} \phi_i \tag{3}$$

Table 5 presents the most accurate and robust trackers for the distortion. Concerning robustness, trackers CPTfast [79] and CPT [79] exhibited an outstanding performance. On the other hand, the most accurate trackers were DeepSTRCF [5], MCCT [76], SRCT [80], and CPTfast [79]. Table 5 shows that the deep feature-based models, DeepSTRCF and Gnet, are accurate and robust when used to track objects in the pristine videos of the AD-SVD dataset. Furthermore, the most accurate tracker under defocus and defocus+exposure

[2] **CNN**: Convolutional Neural Network. **ECO**: Efficient Convolution Operators. **DCF**: Discriminative Correlation Filters. **HOG**: Histogram of Oriented Gradients. **SRB**: Salient Region Based. **VGG**: Visual Geometry Group. **MSE**: Multi-Scale Estimation. **LK**: Linear Kernels.

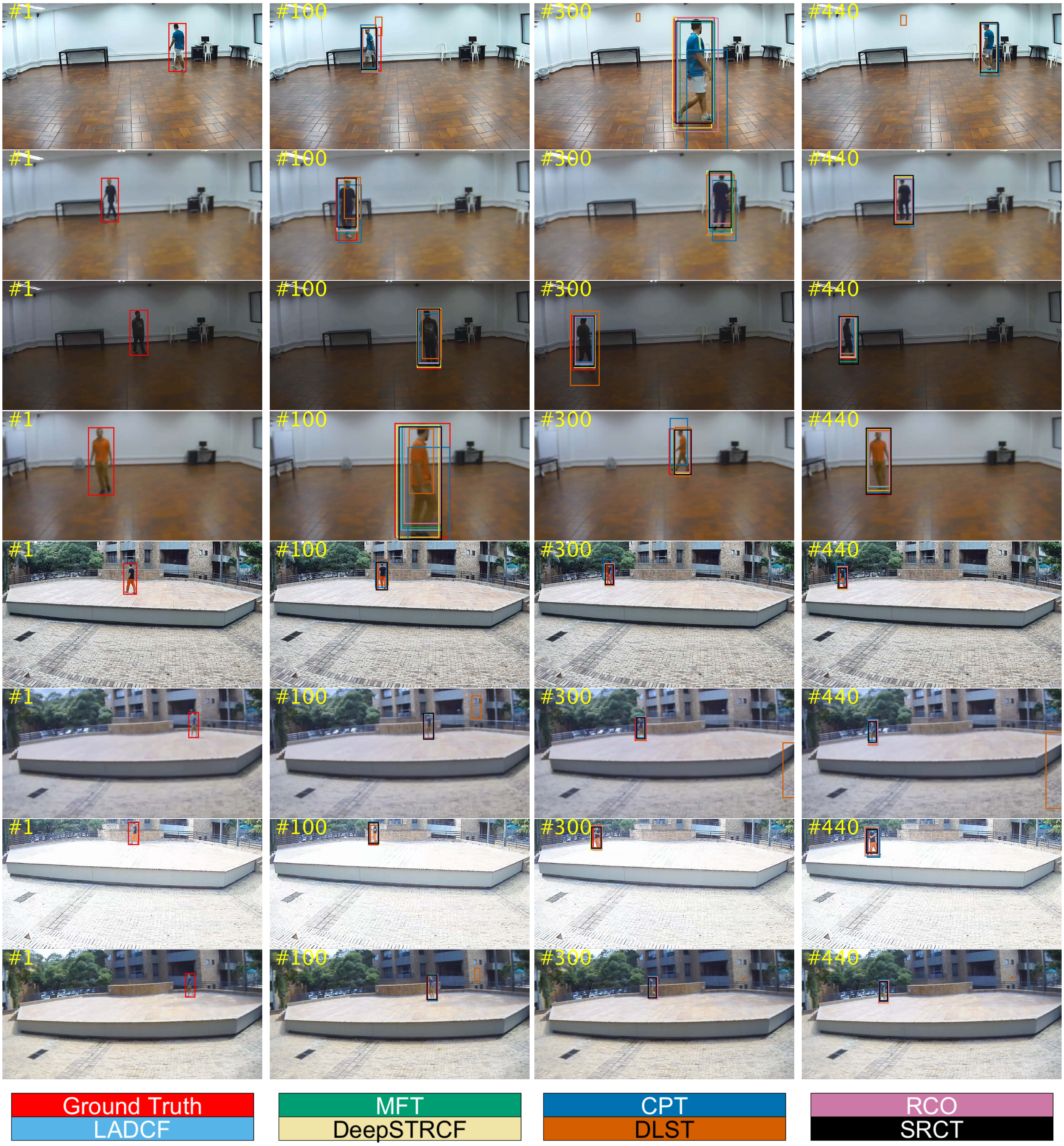


FIGURE 9. Qualitative comparison. The first column shows the target's initial position (ground-truth), and the other columns show the tracking results for seven trackers. These videos are chosen from AD-SVD dataset, which are 0274Pri, 0314Fo, 0302Exp, 0350ExFo, 1113Pri, 1149Fo, 1137Exp, 1185ExFo (Indoor - Outdoor) from top to bottom. The frame number is displayed in the top left corner.

distortion is SRCT, and the most robust tracker is CPT_fast with respect to all distortions. SRCT hinges on a probabilistic color and shape model to discriminate the target region from the background. Since the defocus distortion does not induce abrupt changes in either shape or color, SRCT handles this distortion properly but might be vulnerable to variations of illumination conditions presented in videos impaired by exposure distortions [83]. The feature representation of

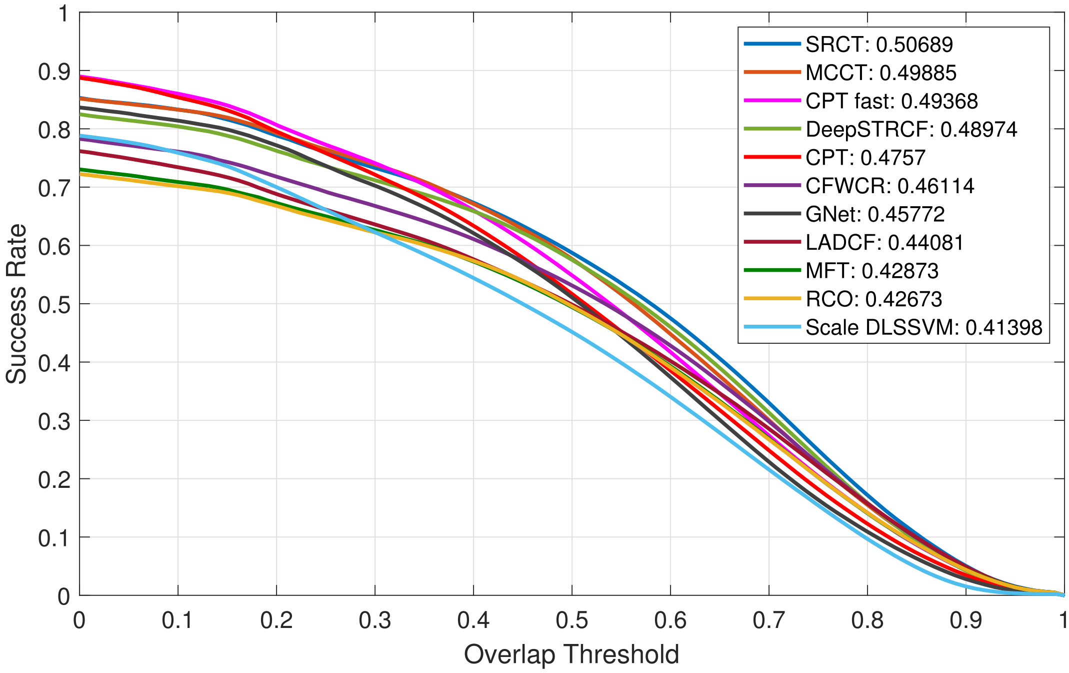


FIGURE 10. Success plot for each tracker after evaluating in the AD-SVD. Tracker:AUC.

CPT_fast relies on a dynamic channel number for selected convolutional layers. Hence, channel pruning tracker (CPT) via channel pruned model and feature maps use more deep convolutional layers with rich semantic features, making it suitable for object tracking in authentically distorted video sequences.

TABLE 5. Best performing trackers for each considered distortion.

Distortion condition	Most Accurate	Most Robust
Pristine	DeepSTRCF [5]	Gnet [75]
Exposure	CPT_fast [79]	CPT_fast [79]
Defocus	SRCT [80]	CPT_fast [79]
Defocus + Exposure	SRCT [80]	CPT_fast [79]

III. VIDEO TRACKER PERFORMANCE PREDICTION IN AUTHENTICALLY DISTORTED VIDEOS

Visual tracking is a very active area in the Computer Vision field. Choosing a tracker for a particular application is challenging, given the high number of competitive algorithms published each year [82]. To facilitate comparisons across different approaches, we designed a model-agnostic (independent of the tracker model [84]) framework that predicts performance without running the corresponding tracking algorithm. To this end, we learn a mapping [23] between the input video and the area under the curve (AUC) of the success plot. This process is carried out in two stages, as shown in Figure 11: i) extraction of a fixed-size set of features, and ii) AUC estimation using a support vector machine regression model.

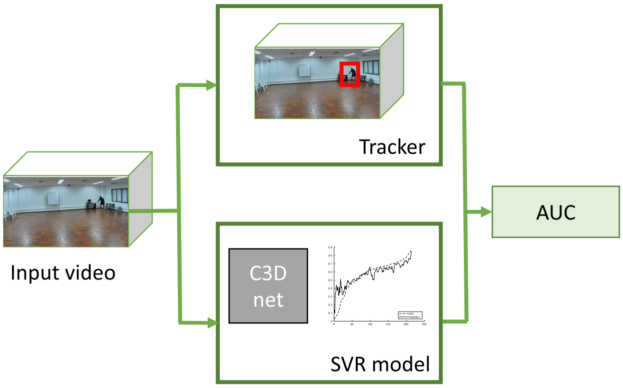


FIGURE 11. Performance prediction framework.

A. FEATURE EXTRACTION

C3D [85] spatio-temporal features extracted from the action recognition task in authentically distorted sports videos have shown useful representation abilities in tasks such as action recognition [86], action similarity labeling, scene classification, and object recognition [85]. C3D Network was trained using the Sports-1M dataset [87], which comprises more than 1 million YouTube videos belonging to 487 classes. Similarly, other studies have demonstrated that a CNN trained for object and video recognition could be useful in determining human perceptual characteristics [86], [88]–[90]. We believe that C3D low-level spatio-temporal features help learn perceptual quality features and predict VOT performance in authentically distorted surveillance videos.

Therefore, we represent a video $V \in \mathbb{R}^{h \times w \times n}$ with n frames ($h \times \; w$ size) by the feature vector $x \in \mathbb{R}^d$. We obtain this feature vector by averaging the deep convolutional 3-D (C3D) features [85] such that:

$$x = \frac{1}{n} \sum_{n} C3D(V).$$

In our experiments, we computed 4096 C3D features extracted from the sixth fully connected (fc6) layer, as is shown in Figure 12. We found better performance when using the corresponding 1024 principal components projections with a retained variance of 99%.

B. SUPPORT VECTOR MACHINE REGRESSION MODEL

Our regression task consists of estimating the AUC $z_i \in \mathbb{R}$ from a feature representation of the i-th video sequence $\mathbf{x}_i \in \mathbb{R}^d$. For this purpose, we trained a support vector machine regression (SVR) model using the formulation introduced in [91]:

$$\begin{aligned} \min_{\alpha,\alpha^*} f(\alpha, \alpha^*) &= \frac{1}{2}(\alpha - \alpha^*)^T K(\alpha - \alpha^*) \\ &\quad + \epsilon \sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i(\alpha_i - \alpha_i^*) \\ \text{subject to } &\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ &0 \le \alpha_i, \alpha_i^* \le C, \quad i = 1, \ldots, l, \end{aligned}$$

where α and α^* are learned weights, C is an upper bound, and $K(x_i, x_j)$ is a Gaussian radial basis function defined by $\exp(\frac{||x_i - x_j||^2}{2\sigma^2})$. To predict the new values, we used

$$\hat{y} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i, x) + b$$

where $\hat{y}$ is the estimated AUC performance and b is a bias term. We set the hyperparameter $C = 50$ as the one that performs the best in a search in $\{0.01, 0.1, 0.5, 0, 10, 50, 100\}$.

IV. VIDEO TRACKER EXECUTION TIME REDUCTION

Video storage and speed demands for video surveillance applications are challenging. For instance, the storage of videos acquired by surveillance cameras may require tens of gigabytes per day. This storage demand can be alleviated by applying compression techniques. Compression can be obtained by increasing the quantization factor, changing the frame rate, or decreasing the frame resolution [92]. Because these compression alternatives can reduce not only the video quality perceived by a user but also the performance of analysis algorithms such as VOT [27], [93], it is essential to monitor and predict these performance changes. This section focuses on developing a framework to reduce video tracker computation resources (such as time and disk space required for storage). This is achieved by predicting the VOT performance on authentically distorted surveillance videos to determine the optimal frame resolution scale for processing the video. This optimal scale reduces the video storage demands and execution time of the video tracker, preserving its performance.

As the experiments presented in Section V show, a reduction in the spatial resolution of the videos typically implies a reduction in the performance of a tracker. However, the performance loss (pl) depends on the tracker, video, and spatial resolution reduction. We proposed a predictor of the performance loss of a tracker defined as

$$pl_{stv} = p_{1tv} - p_{stv},$$

where s is the resolution reduction scale, t represents one of the trackers, v is one of the videos of the dataset, and p_{stv} is the AUC obtained from the tracker t in the video v in the scale s. p_{1tv} is the AUC obtained at the original resolution of the video. Suppose pl is known before using the tracker. In that case, it is possible to decide whether the tracker should be used on the original video or a compressed version, with a controlled loss in performance but with a gain in processing time. The decision criterion is the threshold for pl. Figure 13 illustrates this process.

Similar to Section III, we chose an SVR as our model type and we used the C3D features to train the models. We followed the ensemble model approach in which 10 SVR models were trained and they differed in the hyperparameter and training set used. The final prediction was the average value of the 10 estimates. The hyperparameters were selected based on the performance results of the models on a validation set. We explored two different types of kernel functions: $K_1(x_i, x_j) = \exp(\frac{||x_i - x_j||^2}{2\sigma^2})$ and $K_2(x_i, x_j) = (\frac{<x_i, x_j>}{2\sigma^2})^d$ with $d \in \{1, 2, 3, 4, 5\}$. We varied the hyperparameter C with the values in $\{0.001, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 1.5, 2\}$. For each different scale and tracker we trained a different predictor.

V. EXPERIMENTS

A. VOT AUC PREDICTION ON THE AD-SVD DATASET

Tables 6 to 12 tabulate the Spearman-Rank Correlation Coefficient (SRCC), Pearson Linear Correlation Coefficient (PLCC), and the Root Mean Squared Error (RMSE) calculated between the predicted $\hat{y}$ AUC performance and the actual value y obtained after applying each tracker on a given test set in the seven locations listed in Table 1 (blue/red indicate the best/worst performance, respectively). We changed the size of the test set in such a way that the training set contains 75%, 25%, 5% and 0%, of the videos recorded in a given outdoor-indoor location as shown in Figures 1 and 2. When the set size represented in training set increased, the prediction of the AUC became more accurate. For instance, the correlations (PLCC, SRCC) of GNet in Theater location (Table 6) were (0.9034,0.8584), (0.7216,0.7279),

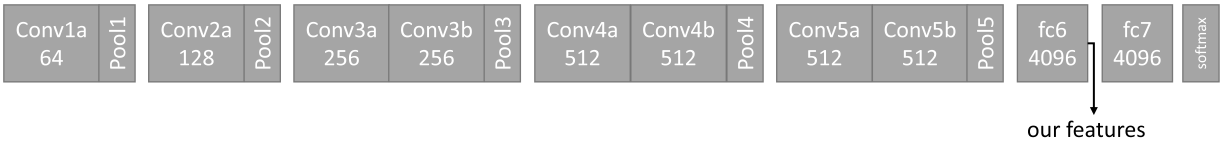


FIGURE 12. C3D architecture [85].

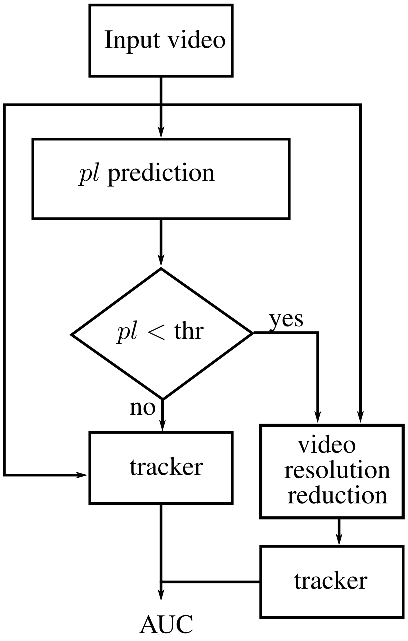


FIGURE 13. Framework for time reduction - *thr* represents the threshold for maximum performance loss that the system will allow.

(0.4837,0.4467), and (0.1763,0.1547) for 75%, 25%, 5% and 0% distribution mixes, respectively. DeepSTRCF exhibited acceptable performance in most cases, outperforming the others in Tables 7, 8, 10, and 11. Moreover, the performance of all trackers on the Parking Lot 2 location was low, even when the set size was 75%. This location represents a challenge because of the limited number of videos available for this scenario (41 videos as shown in Table 1).

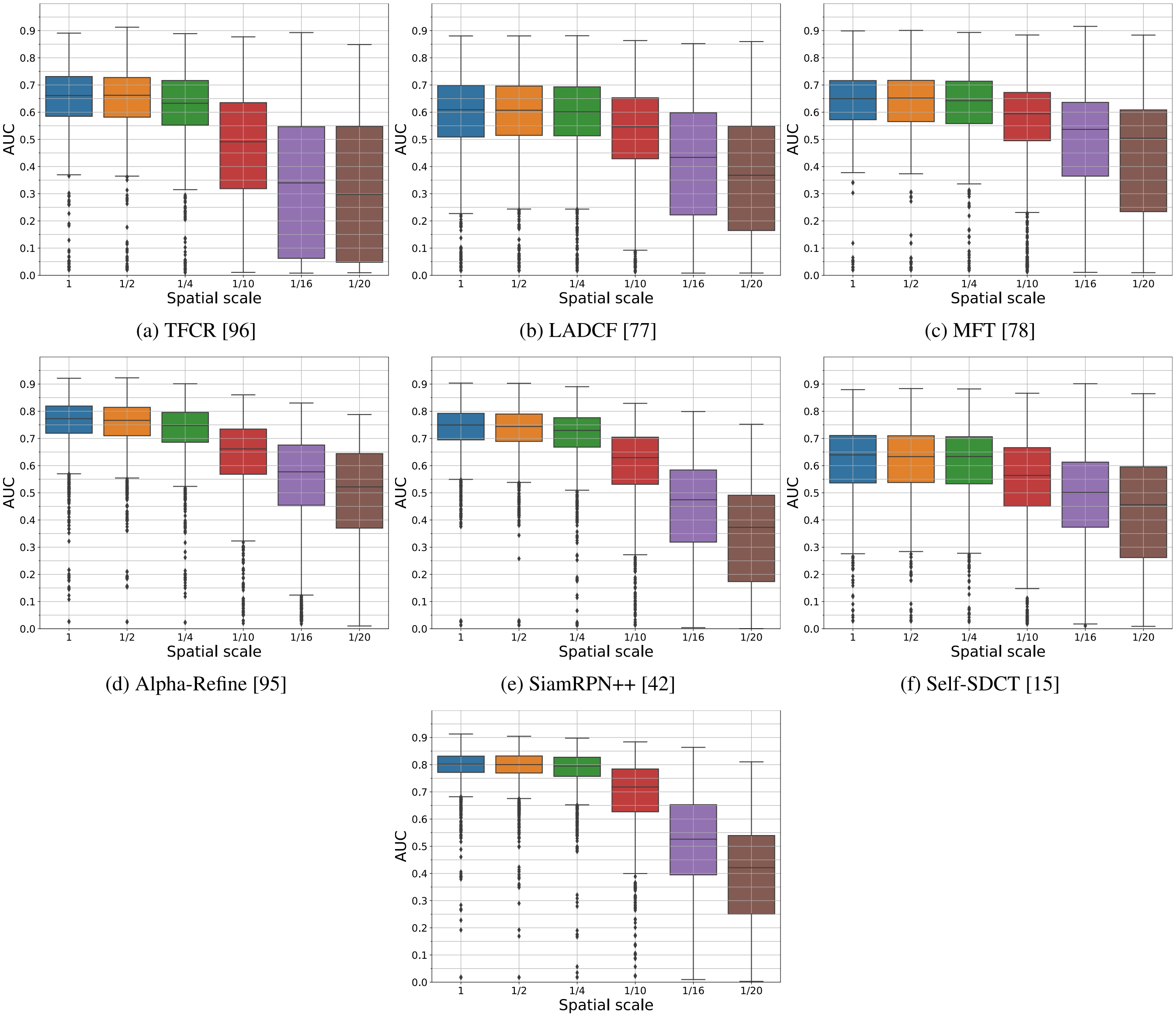
As an alternative to the C3D features, we used a Two-Level Video Quality Model (TLVQM) based representation [94], but the AUC prediction was unsuccessful. TLVQM is a feature encoder that extracts so-called low complexity features (computed on the whole sequence) and high complexity features (calculated on a subset of representative frames). These results confirm our hypothesis that deep convolutional 3D features properly encode a valuable representation that can be used to predict VOT performance on authentically distorted videos.

B. FRAME SPATIAL RESOLUTION AND TIME REDUCTION

Figure 14 depicts the performance of the trackers when the spatial resolution of each frame in the video was reduced to 1/2, 1/4, 1/10, 1/16, and 1/20 of the original resolution. For these experiments, 1385 videos from AD-SVD were selected such that they had 30 fps. Video trackers Alpha-Refine [95], SiamRPN++ [42], MFT [78], TFCR [96], LADCF [77], self-SDCT [15], and TransT [16] were selected, because they have been recently proposed and deliver state-of-the-art performance at the original video resolutions. The results indicate that Alpha-Refine achieved the best performance on all the spatial scales tested.

We executed VOT algorithms in different computer environments. The experiments of LADCF (implemented in MATLAB), SiamRPN++, TransT, and Alpha-Refine trackers were implemented in a computer with the following specifications: processor Intel I7-8750H, 64 GB DDR4-2666 RAM, SSD disk of 512 GB M2, GPU NVIDIA Geforce GTX 1060 with 6GB memory, OS Ubuntu 18.04 LTS (Alpha-Refine) and Windows 10 OS (LADCF, SiamRPN++). MFT and self-SDCT experiments were conducted on a computer with the following specifications: Processor Intel I7-8700K, 40 GB DDR4-2666 RAM, SSD disk of 2 TB, GPU NVIDIA Geforce Titan XP with 12 GB memory, OS Ubuntu 18.04 LTS. TFCR experiments were carried out on the Frontera Computing System at UT Austin, the 13th most powerful supercomputer globally. The specifications for the used Maverick node in Frontera are two (2) Processors Xeon(R) Platinum 8160 CPU @ 2.10GHz with 24 cores, RAM 192 GB, and two (2) Nvidia Tesla P100 16 GB GPUs.

Figure 15 shows the median time required by the trackers to process the video. These results indicate that TFCR and MFT need more time and have lower performance than Alpha-Refine [95] and TransT [16]. The average time required to calculate C3D Features measured in the computer with the Intel I7-8700K processor for a 1920×1080 frame is 0.0048 seconds. Therefore, 2.1793 seconds are required to calculate C3D features for all the 451 frames of one video from AD-SVD. These times include a post-processing stage, using MATLAB, to generate the matrix containing the features for the whole video. The video set used to measure processing time comprises a subset of 140 videos, processed serially to guarantee that the computer/node used did not execute other demanding tasks simultaneously. Once the C3D features have been calculated, the median time needed to predict the AUC is 15 ± 4.4 ms. We measured this median time from the execution of models which used C3D features and predicted the performance of trackers if the resolution of the videos were reduced to 1/10 of their original resolution.



(a) TFCR [96]

(b) LADCF [77]

(c) MFT [78]

(d) Alpha-Refine [95]

(e) SiamRPN++ [42]

(f) Self-SDCT [15]

(g) TransT tracker [16]

FIGURE 14. Tracker performance with spatial scale variation. The original resolution of the videos was reduced at 1/2, 1/4, 1/10, 1/16, 1/20 of the original resolution.

TABLE 6. Theater location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.8264	0.8778	0.0080	0.6175	0.5935	0.0204	0.3786	0.3278	0.0328	0.1874	0.1665	0.1157
CPT [79]	0.8123	0.8677	0.0110	0.6431	0.6756	0.0181	0.4000	0.3625	0.0258	-0.0504	-0.0729	0.0709
DLSSVM [73]	0.7508	0.7989	0.0136	0.6413	0.6408	0.0196	0.3945	0.3627	0.0321	0.0511	0.0399	0.0875
DeepSTRCF [5]	0.8693	0.8906	0.0082	0.7157	0.6757	0.0201	0.4618	0.4147	0.0314	0.1976	0.1559	0.0930
GNet [75]	0.9034	0.8584	0.0059	0.7216	0.7279	0.0179	0.4837	0.4467	0.0288	0.1763	0.1547	0.0804
LADCF [77]	0.8000	0.8510	0.0143	0.5767	0.5697	0.0292	0.4718	0.4030	0.0366	0.1174	0.1019	0.0907
MCCT [76]	0.7803	0.8339	0.0142	0.7038	0.6841	0.0175	0.5047	0.4542	0.0273	0.1403	0.0906	0.0731
MFT [78]	0.8655	0.8891	0.0118	0.7603	0.7181	0.0218	0.4345	0.3664	0.0449	0.1417	0.1604	0.1430
RCO [4]	0.7777	0.8550	0.0202	0.7556	0.7057	0.0245	0.4342	0.3632	0.0468	0.0780	0.1311	0.1398
SRCT [80]	0.8226	0.8575	0.0076	0.6169	0.5673	0.0171	0.3617	0.2629	0.0264	0.0537	0.0159	0.0772

To train and obtain predictions for the performance loss, we used a 10-fold cross-validation approach. For the training, we used, as software tools, Python version 3.7.9 with the library scikit-learn version 0.24.1 [97]. For each

TABLE 7. Parking Lot 2 location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	-0.1385	-0.3091	0.0508	0.3125	0.2508	0.0325	0.1434	0.0162	0.0627	0.1228	0.0091	0.0706
CPT [79]	0.4249	0.2848	0.0271	0.5147	0.4657	0.0436	0.4003	0.3636	0.0384	0.4574	0.4382	0.0386
DLSSVM [73]	0.1521	0.2242	0.0209	0.3371	0.4016	0.0270	0.1282	0.1617	0.0422	0.1416	0.1685	0.0571
DeepSTRCF [5]	0.5347	0.4303	0.0313	0.6030	0.5641	0.0324	0.6123	0.5538	0.0317	0.6221	0.5672	0.0297
GNet [75]	0.4110	0.3091	0.0159	0.3452	0.3593	0.0315	0.3017	0.2425	0.0296	0.3012	0.2639	0.0307
LADCF [77]	0.1974	0.0424	0.0366	0.2410	0.3020	0.0295	0.1762	0.1561	0.0386	0.1836	0.1155	0.0570
MCCT [76]	0.2271	0.1152	0.0426	0.6184	0.6230	0.0321	0.3608	0.3571	0.0379	0.4072	0.4017	0.0448
MFT [78]	0.3217	0.3576	0.0503	0.0563	0.0722	0.0404	0.1562	0.0921	0.0535	0.1787	0.1186	0.0520
RCO [4]	-0.0551	-0.1394	0.0566	0.0201	0.0379	0.0451	0.1269	0.0603	0.0587	0.1351	0.0303	0.0596
SRCT [80]	0.1801	-0.0303	0.0272	0.2271	0.1617	0.0505	0.3313	0.2856	0.0527	0.2975	0.1852	0.0643

TABLE 8. Parking Lot 1 Location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.8641	0.8425	0.0188	0.5761	0.5701	0.0473	0.3239	0.3006	0.0690	0.0685	0.0512	0.0885
CPT [79]	0.8264	0.8529	0.0148	0.4969	0.5141	0.0243	0.2022	0.1890	0.0342	0.0583	0.1113	0.0343
DLSSVM [73]	0.7771	0.7939	0.0168	0.5964	0.6066	0.0255	0.3552	0.3453	0.0355	0.1412	0.1614	0.0399
DeepSTRCF [5]	0.8856	0.8823	0.0150	0.5890	0.5800	0.0334	0.4452	0.4215	0.0432	0.3473	0.3319	0.0494
GNet [75]	0.8588	0.8442	0.0134	0.4951	0.4744	0.0337	0.3678	0.3258	0.0420	0.1953	0.1871	0.0645
LADCF [77]	0.7792	0.7827	0.0227	0.4994	0.4935	0.0435	0.2523	0.2403	0.0561	-0.0091	0.0321	0.0652
MCCT [76]	0.8676	0.8919	0.0153	0.5180	0.5175	0.0313	0.2817	0.2414	0.0415	0.1929	0.1891	0.0409
MFT [78]	0.8467	0.7869	0.0194	0.5699	0.5581	0.0471	0.3154	0.2717	0.0708	0.1487	0.1166	0.0915
RCO [4]	0.8296	0.7728	0.0200	0.5756	0.5688	0.0439	0.3036	0.2406	0.0669	0.1231	0.0879	0.0926
SRCT [80]	0.7839	0.8170	0.0203	0.5019	0.5053	0.0396	0.2549	0.2548	0.0528	0.1294	0.1241	0.0553

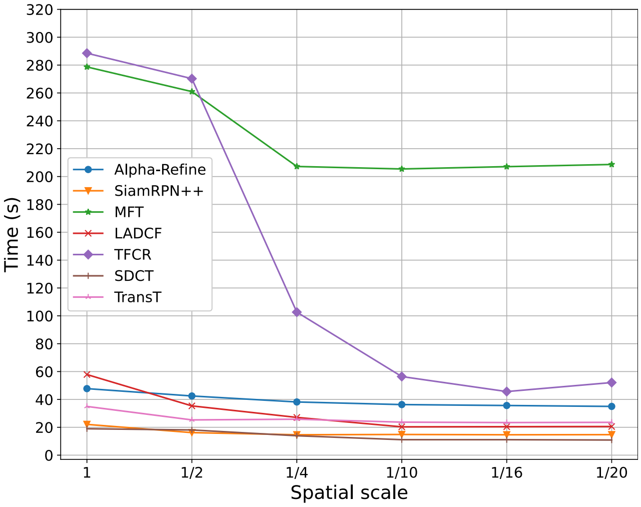
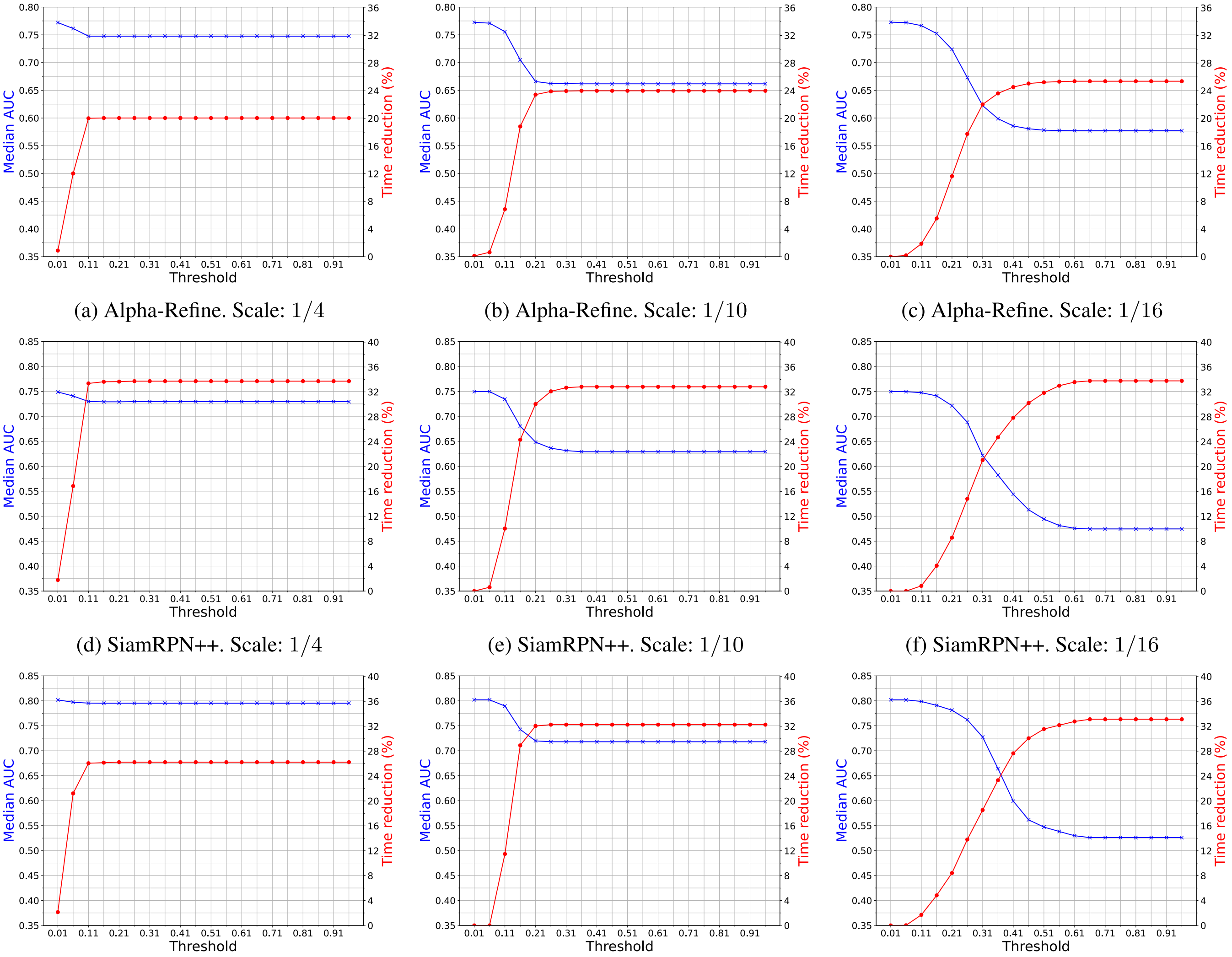


FIGURE 15. Median time (seconds) required by the trackers to process a video with 450 frames of AD-SVD.

of the 10 test sets, the remaining videos were used to build the training and validation sets. The selection of the ten training and validation sets for each test was carried out randomly. We used 139 videos for testing, 208 videos for validation, and 1038 videos for training, which corresponds to 10%, 15%, and 75% of the entire set of videos, respectively.

Figures 16 and 17 depict the results of the framework illustrated in Figure 13 with different values for the threshold. Figure 16d shows that SiamRPN++ tracker [42] dropped 0.025 in performance on a 1/4 scale while achieving a 34% time reduction in the total processing time. Figure 17d shows that the TFCR tracker [96] achieves a 65% time reduction with only 0.03 in performance loss measured by the median AUC. Nonetheless, it is important to consider that the median performance of SiamRPN++ changed from 0.75 to 0.725 at 1/4 spatial scale, which is still high performance. Meanwhile, at the spatial scale of 1/4 (Figure 17d), the TFCR changed from 0.66 to 0.62, which is comparable to the performance achieved by SiamRPN++ at the spatial scale of 1/10 (Figure 16e). TFCR achieves the largest improvement in time reduction with 80% and 84% at spatial scales of 1/10 (Figure 17e) and 1/16 (Figure 17f), respectively. These results can also be perceived from the drop in the median time required per video by TFCR, as depicted in Figure 15. However, the performance drop of the TFCR is 0.16 and 0.31, at spatial scales of 1/10 and 1/16, respectively. In general, the best results were obtained on scales 1/4 and 1/10 due to the requirements imposed by video tracker algorithms on input video resolution. A scale smaller than 1/4 or 1/10, depending on the tracker, does not imply a larger reduction in the median time needed per video. Hence, there may be a reduction in VOT performance but not in the time required for VOT algorithm execution.



(a) Alpha-Refine. Scale: 1/4 (b) Alpha-Refine. Scale: 1/10 (c) Alpha-Refine. Scale: 1/16

(d) SiamRPN++. Scale: 1/4 (e) SiamRPN++. Scale: 1/10 (f) SiamRPN++. Scale: 1/16

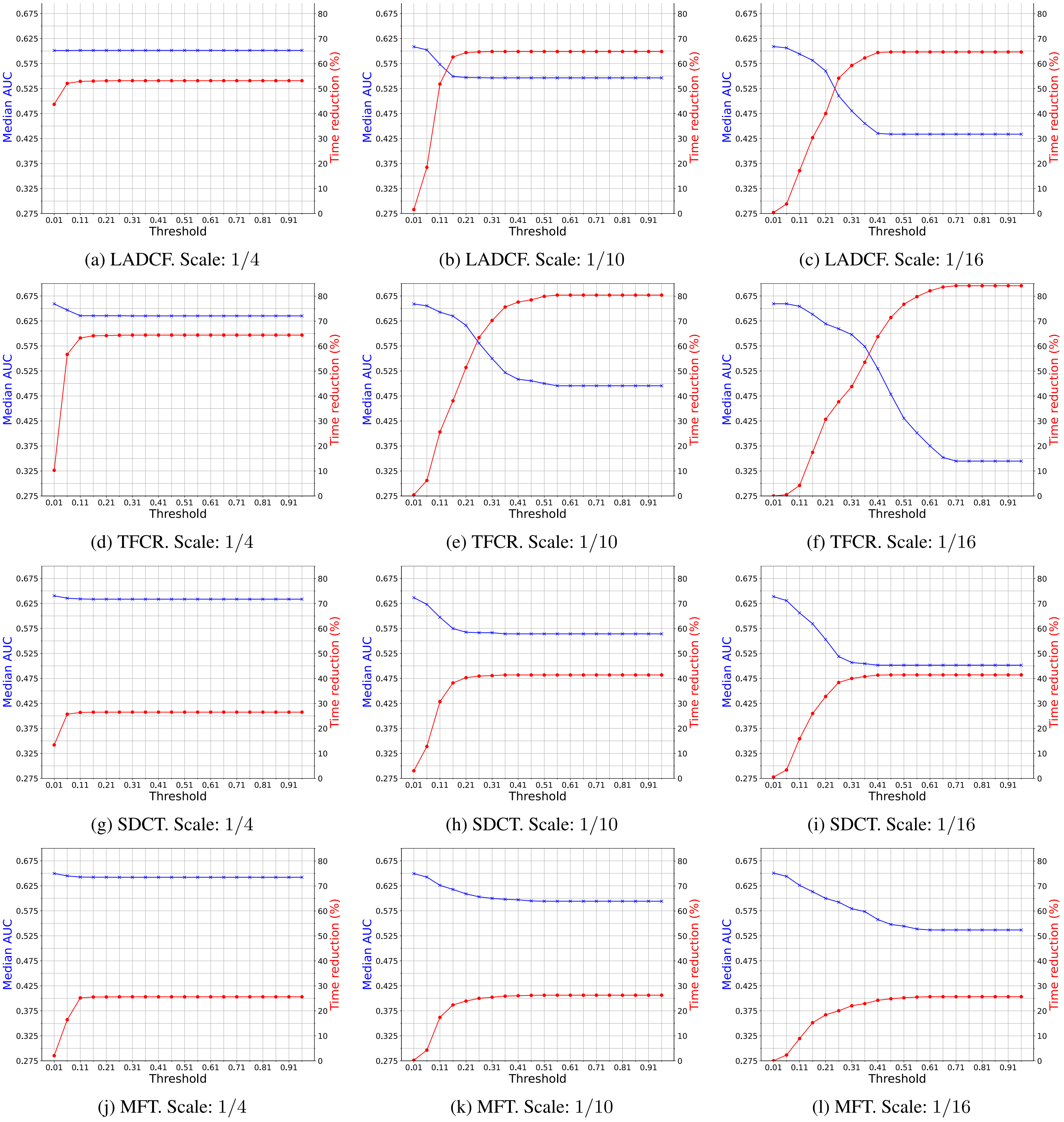
(g) TransT. Scale: 1/4 (h) TransT. Scale: 1/10 (i) TransT. Scale: 1/16

FIGURE 16. Time reduction vs performance loss.

TABLE 9. Media room location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.7943	0.7860	0.0177	0.6999	0.6091	0.0266	0.5117	0.4725	0.0414	0.1830	0.1542	0.0630
CPT [79]	0.7260	0.6844	0.0185	0.6322	0.5213	0.0224	0.4025	0.3472	0.0322	0.1528	0.1526	0.0474
DLSSVM [73]	0.7491	0.6952	0.0144	0.6321	0.5820	0.0194	0.4594	0.3694	0.0274	0.0621	0.0189	0.0432
DeepSTRCF [5]	0.8086	0.7675	0.0161	0.6889	0.5601	0.0231	0.5124	0.4657	0.0331	0.2865	0.2395	0.0519
GNet [75]	0.7197	0.6682	0.0210	0.6259	0.5646	0.0261	0.5442	0.4896	0.0317	0.1513	0.1387	0.0526
LADCF [77]	0.7851	0.7468	0.0218	0.5999	0.5360	0.0381	0.4914	0.4197	0.0482	0.1379	0.1076	0.0702
MCCT [76]	0.7729	0.7889	0.0176	0.6657	0.6248	0.0231	0.4724	0.3891	0.0352	0.2328	0.1881	0.0492
MFT [78]	0.8083	0.7818	0.0210	0.6298	0.5428	0.0355	0.4976	0.4323	0.0450	0.1131	0.0813	0.0742
RCO [4]	0.8215	0.8034	0.0203	0.6558	0.5716	0.0335	0.5575	0.4947	0.0418	0.0743	0.0458	0.0747
SRCT [80]	0.8474	0.8048	0.0129	0.6810	0.5894	0.0251	0.5259	0.4561	0.0368	0.2178	0.1714	0.0530

Figure 18 shows the results of the proposed framework on two of the trackers presented in Tables 3, 4, and Figure 10. Figure 18a depicts the results for the LADCF tracker with the performance loss threshold of 0.11. Figure 18b depicts the results for the MFT tracker with the performance loss threshold of 0.16. We selected a scale of 1/10 of the original



(a) LADCF. Scale: 1/4 (b) LADCF. Scale: 1/10 (c) LADCF. Scale: 1/16

(d) TFCR. Scale: 1/4 (e) TFCR. Scale: 1/10 (f) TFCR. Scale: 1/16

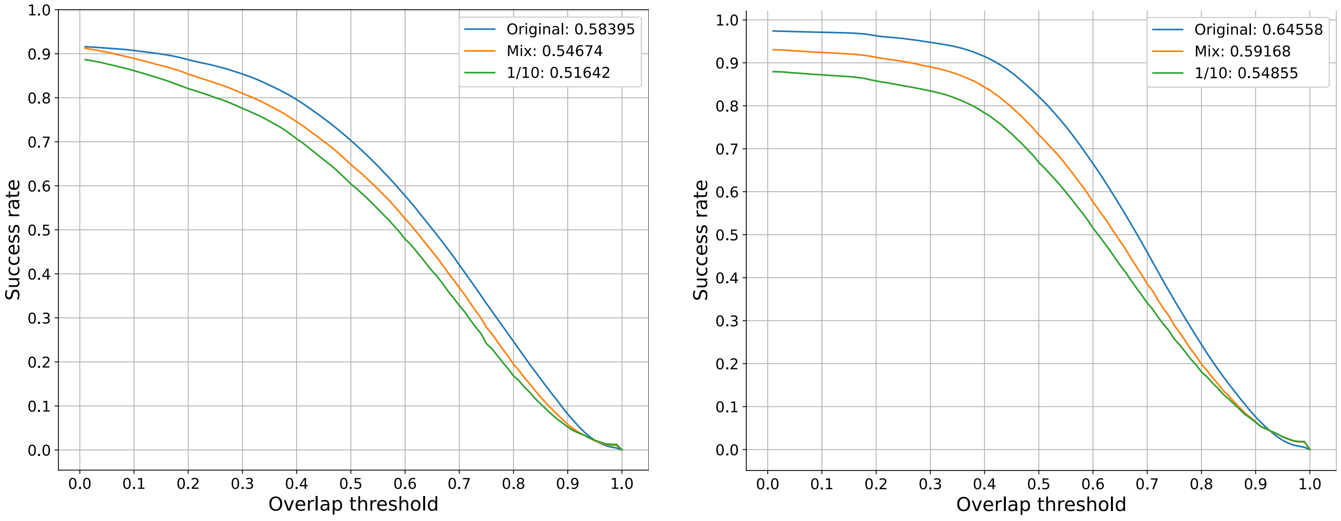
(g) SDCT. Scale: 1/4 (h) SDCT. Scale: 1/10 (i) SDCT. Scale: 1/16

(j) MFT. Scale: 1/4 (k) MFT. Scale: 1/10 (l) MFT. Scale: 1/16

FIGURE 17. Time reduction vs performance loss.

resolution for these demonstrations. The proposed framework allows to obtain performances similar to those observed at the original resolution and reduces the execution time and memory requirements for video storage.

We performed an additional test on the VOT-2020 Short Term Dataset [2] on the proposed method to reduce the scale and preserve the VOT algorithms performance. We used 58 videos of the VOT dataset, with spatial resolutions ranging from 640 × 480 to 1920 × 1080 (lower than the Full HD resolution of all videos in the AD-SVD dataset). These videos contain different scenes and objects' sizes. We concluded that the video resolution in the original scale and the bounding box's size significantly impact the proposed method's performance. In some cases, it was not possible to obtain results in



(a) LADCF

(b) MFT

FIGURE 18. Success plots of the proposed framework on trackers LADCF and MFT. Scale:AUC.

TABLE 10. Industrial Lab location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.6698	0.6562	0.0253	0.5022	0.4628	0.0363	0.1906	0.1683	0.0535	-0.0067	-0.0202	0.0702
CPT [79]	0.5655	0.5997	0.0174	0.3894	0.3876	0.0245	0.1871	0.1715	0.0299	0.1763	0.1672	0.0346
DLSSVM [73]	0.5078	0.4824	0.0234	0.3406	0.3327	0.0299	0.0928	0.0827	0.0426	-0.0006	-0.0058	0.0506
DeepSTRCF [5]	0.6842	0.6786	0.0209	0.5242	0.5236	0.0309	0.3509	0.3430	0.0386	0.0147	0.0106	0.0568
GNet [75]	0.5898	0.6072	0.0218	0.4673	0.4562	0.0261	0.2133	0.1737	0.0342	0.1074	0.1016	0.0345
LADCF [77]	0.6814	0.6720	0.0193	0.4302	0.4118	0.0315	0.2117	0.2001	0.0389	0.0370	0.0450	0.0502
MCCT [76]	0.5431	0.5511	0.0258	0.4770	0.4618	0.0303	0.2252	0.1990	0.0392	0.0566	0.0515	0.0453
MFT [78]	0.6657	0.6521	0.0267	0.4833	0.4433	0.0389	0.0833	0.0611	0.0586	-0.1293	-0.1208	0.0661
RCO [4]	0.5944	0.5763	0.0306	0.4861	0.4568	0.0373	0.1038	0.0797	0.0550	-0.1272	-0.1129	0.0697
SRCT [80]	0.5466	0.5550	0.0300	0.4609	0.4220	0.0356	0.1753	0.1357	0.0491	0.0672	0.0688	0.0566

TABLE 11. Guayacanes Hall location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.6924	0.7103	0.0386	0.5585	0.6005	0.0640	0.6209	0.6285	0.0469	0.3392	0.3368	0.1059
CPT [79]	0.4746	0.4573	0.0395	0.4564	0.5414	0.0413	0.5354	0.5359	0.0423	0.4753	0.4399	0.0549
DLSSVM [73]	0.6859	0.6722	0.0188	0.4447	0.4772	0.0257	0.4142	0.4239	0.0308	0.3373	0.3069	0.0374
DeepSTRCF [5]	0.8120	0.8087	0.0177	0.5912	0.5197	0.0342	0.6658	0.6553	0.0293	0.4060	0.4380	0.0682
GNet [75]	0.5620	0.6479	0.0396	0.4725	0.4789	0.0436	0.5167	0.5259	0.0391	0.3306	0.3035	0.0664
LADCF [77]	0.6304	0.6133	0.0244	0.6336	0.6483	0.0235	0.5767	0.5655	0.0255	0.3332	0.3835	0.0769
MCCT [76]	0.7633	0.7354	0.0216	0.6284	0.6665	0.0293	0.6551	0.6162	0.0411	0.4475	0.4573	0.1009
MFT [78]	0.7245	0.7842	0.0448	0.5799	0.5637	0.0605	0.5722	0.4818	0.0573	-0.0323	-0.0467	0.1231
RCO [4]	0.6859	0.7628	0.0444	0.4888	0.5554	0.0493	0.4720	0.4652	0.0614	-0.0371	-0.0682	0.1232
SRCT [80]	0.6556	0.6873	0.0396	0.5407	0.5938	0.0504	0.5057	0.4878	0.0676	0.3660	0.3463	0.1000

small scales such as 1/20 due to the small area of the bounding box (smaller than 20 pixels in width or height). In conclusion, the improvements of our proposed approach are more evident in high-resolution videos (e.g., 1920 × 1080).

As an exploratory experiment, we applied the DLSSVM tracker on 1000 videos at different temporal scales: 1, 1/2, 1/5, 1/10, 1/20, 1/30, and 1/40. 1/n denotes the proportion of frames with respect to the original video length. Figure 19 shows that the performance did not decrease even when using the 1/10 scale. This would enable us to speed up the process ten times, given that the time reduction is directly proportional to the scale. Thus, temporal downscaling can reduce the computational cost and storage requirements, which is an advantage in video surveillance. However, further research is needed to expand and test these preliminary findings on other state-of-the-art trackers.

TABLE 12. Electronics Lab location.

Tracker	Distribution mix											
	75%			25%			5%			0%		
	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE	PLCC	SRCC	RMSE
CFWCR [74]	0.6752	0.6881	0.0163	0.4304	0.3965	0.0273	0.1277	0.1222	0.0423	-0.0606	-0.0550	0.0528
CPT [79]	0.7267	0.7608	0.0095	0.4206	0.4019	0.0216	0.1614	0.1387	0.0272	0.0185	0.0248	0.0307
DLSSVM [73]	0.6391	0.6693	0.0116	0.3495	0.3415	0.0195	0.1048	0.1072	0.0264	0.0497	0.0394	0.0306
DeepSTRCF [5]	0.6894	0.7007	0.0150	0.3611	0.3356	0.0259	0.0020	0.0139	0.0368	-0.0514	-0.0195	0.0440
GNet [75]	0.7015	0.6360	0.0120	0.4362	0.3432	0.0212	0.0025	-0.0381	0.0313	-0.0877	-0.0701	0.0314
LADCF [77]	0.6779	0.6883	0.0188	0.4817	0.4070	0.0256	0.1500	0.1296	0.0412	-0.0155	-0.0007	0.0459
MCCT [76]	0.6476	0.6685	0.0189	0.4103	0.3367	0.0250	0.0228	0.0471	0.0354	-0.0374	0.0161	0.0396
MFT [78]	0.7466	0.7276	0.0153	0.4892	0.4469	0.0319	0.1202	0.0853	0.0512	-0.1170	-0.1087	0.0627
RCO [4]	0.7997	0.7616	0.0131	0.4845	0.4231	0.0314	0.0794	0.0485	0.0545	-0.1726	-0.1621	0.0686
SRCT [80]	0.6506	0.7265	0.0194	0.4225	0.4149	0.0272	0.2088	0.1870	0.0353	0.0408	0.0493	0.0446

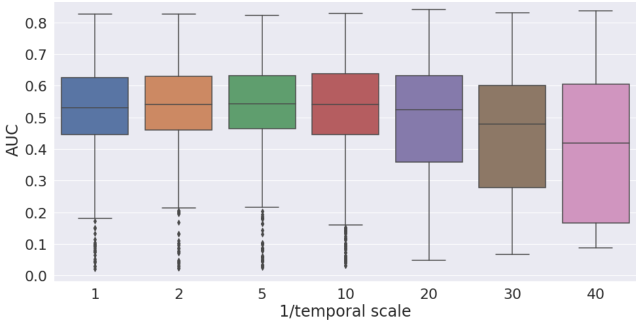


FIGURE 19. Temporal resolution analysis: AUC performance at different temporal scales.

VI. CONCLUSION

This study introduced the AD-SVD dataset, which models three levels of severity of in-capture distortions: exposure, lack of focus (defocus), and a combination of these impairments. AD-SVD is the largest, densely annotated, and authentically distorted video object tracking benchmark for STT. We also proposed and tested a performance prediction approach for single object tracking of authentically distorted surveillance videos. With a high level of accuracy, this framework predicts the performance of a VOT algorithm on several outdoor and indoor locations, different visual contents, and under diverse types and levels of authentic distortion. Furthermore, this framework reduces video tracker computation resources, such as time and disk space required for storage, by predicting VOT algorithm performance to determine the optimal spatial scale to process a video.

In addition, we carried out experiments to explore the effects on VOT performance by reducing the number of frames processed by a VOT algorithm. The proposed methodology preserves the VOT algorithm performance in these experiments for different frame rate reductions. These reductions were up to 1/10 of the original time resolution for a subset of 1000 videos extracted from AD-SVD, which contain various distortions and visual contents. Further studies to determine the usefulness of frame reduction in VOT might validate these encouraging results by incorporating more tracking algorithms and data scenes.

We also proposed and tested a performance prediction approach for single object tracking of authentically distorted surveillance videos. This framework predicts the performance of a VOT algorithm, with a high level of accuracy, on several outdoor and indoor locations, different visual contents, and under diverse types and levels of authentic impairments and spatial downsampling. To incorporate this tracking prediction framework, the VOT system must allow the offline processing of the original video that feeds the prediction model and select the optimal downsampling scale (that balances out accuracy and computational time).

We tested our proposed approach for video object tracker (VOT) algorithm prediction and video tracker execution time reduction under a wide variety of conditions. For instance, in three intensity levels of authentic distortions such as over and under exposure along with defocus and combinations of these impairments. Moreover, these distortions impair videos captured in three indoor and three outdoor scenarios with six actors playing more than eight activities. In addition, we tested the robustness of sixteen VOT algorithms on these authentically distorted videos. We propose, as future work, new analysis and results with respect to additional VOT conditions such as occlusion and clutter. These could provide a clearer understanding of the complex interactions between perceptual and machine vision quality. Indeed, we performed an additional test on the VOT-2020 Short Term Dataset on the proposed method to reduce the scale and preserve the VOT algorithms performance. We used 58 videos from the VOT dataset, with spatial resolutions ranging from 640 × 480 to 1920 × 1080 (lower than the Full HD resolution of all videos in the AD-SVD dataset). These videos contain different scenes and objects' sizes. We concluded that the video resolution in the original scale and the bounding box's size significantly impact the proposed method's performance. In some cases, it was not possible to obtain results in small scales such as 1/20 due to the small area of the bounding box (smaller than 20 pixels in width or height). In conclusion, the improvements of our proposed approach are more evident in high-resolution videos (1920 × 1080).

Incorporating the proposed tracking prediction tool requires the video to be available offline. However, this framework could be helpful on real-time VOT if the optimal scale is estimated in one of the earliest video chunks. Surveillance videos are commonly recorded at 5 FPS and below to reduce the storage requirements. Therefore, low-frame-rate videos are common, as large-scale camera networks cannot stream and store high-frame-rate videos gathered by thousands of cameras. Instead, cameras are often configured to send one frame every second or so over the network [98]. For this reason, a study that can reduce the FPS at which the trackers work without reducing performance would be beneficial in this type of application. Nonetheless, reducing the standard frame rate increases the possibility of missing important information from the original video sequence. For instance, if an object appears in three frames within a second when the frame rate is 25 FPS, the reduction to one FPS will result in a significant decrease in the probability of finding this object in one selected frame [99]. Hence, further research is needed to expand and test these preliminary findings on other state-of-the-art trackers on surveillance activities videos.

APPENDIX A
TRACKERS FOR SPATIAL SCALE ANALYSIS

1) **TFCR [96]**: *Target-Focusing Convolutional Regression*: This tracker is based on a model that uses a target-focusing loss function to alleviate the influence of background noise on the response map, reducing the effects of the negative samples that act on the object appearance model. TFCR uses a target-focused regression model to train the convolutional neural network (VGGNet [100]), which pays more attention to the target sample and reduces the influence of the background samples on the target appearance model. TFCR extracts search patches at different scales with the exact central location and feeds them into the feature extractor to resolve scale-related challenges. Subsequently,the optimal scale factor was selected by searching for the maximum value in the prediction maps.
2) **Alpha-Refine [95]**: Alpha-Refine was the winner of the VOT2020 Real-Time Challenge with an EAO of 0.499. It is a module implemented in Pytorch [101], which refines the base tracker outputs and improves the tracking performance. This module consists of a pixel-wise correlation, a corner prediction head, and an auxiliary mask head (which can be deactivated at the inference stage to improve speed), introducing pixel-level supervision into the training as the core components. The Alpha-Refine modules were trained for 40 epochs and 500 iterations, each on eight NVIDIA 2080Ti GPUs. This module introduces additional computational loads of approximately 5-6 ms per frame. The Alpha-Refine module was tested on six trackers [38], [42], [44], [102], [103], trained on some segmentation datasets, and tested on multiple tracking benchmarks [2], [63], [65], [66], increasing up to 7.4% of the AUC of the original baseline tracker. In our experiments, we used SiamRPN++ [42] as the base tracker for the Alpha-Refine module.
3) **SiamRPN++ [42]**: SiamRPN++ is a tracker trained with a ResNet-driven deep Siamese network (> 20 layers), using a layer-wise feature aggregation structure for the cross-correlation operation. This network is pre-trained on ImageNet [104], trained with other sets [105], [106], and tested on tracking datasets [4], [59], [60]. SiamRPN++ replaces cross-correlation with depthwise correlation, reducing the computational cost and memory usage. SiamRPN++ operates at 35 Frames per Second (FPS), but it can be increased to 70 FPS using the MobileNet [107] backbone. SiamRPN++ had a 0.414 EAO score on VOT2018, which was 4.0% higher than that of the single-layer baseline.
4) **MFT [78]**: MFT was the winner of the VOT2018 challenge [4]. MFT, implemented on MATLAB, consists of hierarchical feature selection, independent group CF online learning, adaptive multi-branch CF fusion and a motion estimation module (which alleviates the problem of fast motion). This tracker uses multi-hierarchical deep features (ResNet [108] before ReLU, reduced by PCA-256) with different semantic information to track multi-scale objects. The motion estimation module (which improves the robustness to motion blur), based on Kalman filters, generates a Gaussian motion map. Then, hierarchical features from different layers are extracted by a ResNet and multiplied by the Gaussian motion map. These deep features are independently fed into different CFs to update the parameters, using weights to give attention to different channels. Finally, an adaptive weight scheme is utilized to generate a final score map to locate the target. This tracker benefits from online learning to adapt to appearance changes and to scale variances, but with the detriment of being computationally demanding.
5) **LADCF [77]**: LADCF (MATLAB implemented) constructs an appearance model using adaptive spatial feature selection (by lasso regularization) and temporal consistency-preserving spatial feature selection. LADCF uses hand-crafted (HOG, Colour-Names) and deep features of the middle convolutional layers (VGG network) as spatial features. LADCF can simultaneously activate specific spatial features corresponding to the target and background regions to form a robust pattern. It should be noted that only the relevant features are activated for each training sample, forming a low-dimensional feature representation. Finally, LADCF learns discriminative filters in the frequency domain (FFT transformed) using an augmented Lagrangian method, which is used to iteratively optimize the variables (using ADMM [109]).
6) **Self-SDCT [15]**: Self-SDCT (MATLAB implemented) is a multi-cycle consistency loss based self-supervised learning-based tracker embedded in a deep correlation framework. This scheme copes with the issue of requiring numerous manually annotated samples for training.

In addition, Self-DCT enriches the representational ability to reduce the overfitting risk by using a low similarity dropout and a cycle trajectory consistency loss to pre-train the feature extraction network jointly. Self-DCT generates pseudo-labels of these training samples by using a forward-backward prediction under a Siamese correlation based tracking framework. Finally, the Siamese correlation-based tracking architecture provides the basis for real-time tracking.

7) **TransT [16]**: TransT relies on a novel attention-based feature fusion network, which integrates the template and search region features by using attention. TransT consists of three components: the siamese like feature extraction backbone (ResNet50), the designed feature fusion network, and the prediction head. The attention mechanism creates long distance feature associations, making the tracker adaptively focus on useful and abundant semantic information. Several experiments show that TransT performs significantly better than the state-of-the-art algorithms while running at a real-time speed. Indeed, TransTM (a variation of TransT) was the top performer and the winner of the VOT-RT2021, while TransTM ranked in the top ten trackers in this contest.

ACKNOWLEDGMENT

The authors would like to thank NVIDIA Corporation for donating a TITAN XP GPU used in these experiments. They also thank the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper.

REFERENCES

[1] A. Dutta, A. Mondal, N. Dey, S. Sen, L. Moraru, and A. E. Hassanien, ''Vision tracking: A survey of the state-of-the-art,'' *Social Netw. Comput. Sci.*, vol. 1, no. 1, p. 57, Jan. 2020, doi: 10.1007/s42979-019-0059-z.

[2] M. Kristan, *The Eighth Visual Object Tracking VOT2020 Challenge Results BT*, A. Bartoli A. Fusiello, Eds. Cham, Switzerland: Springer, 2020, pp. 547–601, doi: 10.1007/978-3-030-68238-5_39.

[3] M. Kristan, J. Matas, and A. Leonardis, ''The seventh visual object tracking VOT2019 challenge results,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV Workshop)*, Oct. 2019, pp. 2206–2241, doi: 10.1109/ICCVW.2019.00276.

[4] M. Kristan, ''The sixth visual object tracking VOT2018 challenge results,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Sep. 2018, doi: 10.1007/978-3-030-11009-3_1.

[5] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, ''Learning spatial-temporal regularized correlation filters for visual tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913, doi: 10.1109/CVPR.2018.00515.

[6] X. Li, Q. Liu, N. Fan, Z. Zhou, Z. He, and X. Jing, ''Dual-regression model for visual tracking,'' *Neural Netw.*, vol. 132, pp. 364–374, Dec. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608020303348

[7] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, ''Visual object tracking using adaptive correlation filters,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2010, pp. 2544–2550, doi: 10.1109/CVPR.2010.5539960.

[8] S. Du and S. Wang, ''An overview of correlation-filter-based object tracking,'' *IEEE Trans. Computat. Social Syst.*, early access, Jul. 26, 2021, doi: 10.1109/TCSS.2021.3093298.

[9] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, ''Hierarchical convolutional features for visual tracking,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082, doi: 10.1109/ICCV.2015.352.

[10] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, ''Learning spatially regularized correlation filters for visual tracking,'' in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318, doi: 10.1109/iccv.2015.490.

[11] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, ''Convolutional features for correlation filter based visual tracking,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 621–629, doi: 10.1109/ICCVW.2015.84.

[12] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, ''Beyond correlation filters: Learning continuous convolution operators for visual tracking,'' in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 472–488, doi: 10.1007/978-3-319-46454-1_29.

[13] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, ''Accurate scale estimation for robust visual tracking,'' in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–5, doi: 10.5244/c.28.65.

[14] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, ''Hedged deep tracking,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 4303–4311, doi: 10.1109/CVPR.2016.466.

[15] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, ''Self-supervised deep correlation tracking,'' *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[16] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, ''Transformer tracking,'' in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 8122–8131.

[17] S. Gunasekar, J. Ghosh, and A. C. Bovik, ''Face detection on distorted images augmented by perceptual quality-aware features,'' *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2119–2131, Dec. 2014, doi: 10.1109/TIFS.2014.2360579.

[18] F. Xie, Y. Lu, A. C. Bovik, Z. Jiang, and R. Meng, ''Application-driven no-reference quality assessment for dermoscopy images with multiple distortions,'' *IEEE Trans. Biomed. Eng.*, vol. 63, no. 6, pp. 1248–1256, Jun. 2016, doi: 10.1109/TBME.2015.2493580.

[19] R. Soundararajan and S. Biswas, ''Machine vision quality assessment for robust face detection,'' *Signal Process., Image Commun.*, vol. 72, pp. 92–104, Mar. 2019, doi: 10.1016/j.image.2018.12.012.

[20] C. G. Rodríguez-Pulecio, H. D. Benítez-Restrepo, and A. C. Bovik, ''Making long-wave infrared face recognition robust against image quality degradations,'' *Quant. Infr. Thermography J.*, vol. 16, nos. 3–4, pp. 218–242, Oct. 2019, doi: 10.1080/17686733.2019.1579020.

[21] S. F. Dodge and L. J. Karam, ''Quality robust mixtures of deep neural networks,'' *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5553–5562, Nov. 2018, doi: 10.1109/TIP.2018.2855966.

[22] J. M. Irvine, R. J. Wood, D. Reed, and J. Lepanto, ''Video image quality analysis for enhancing tracker performance,'' in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2013, pp. 1–9, doi: 10.1109/AIPR.2013.6749326.

[23] A. Gala and S. Shah, ''Joint modeling of algorithm behavior and image quality for algorithm performance prediction,'' in *Proc. Procedings Brit. Mach. Vis. Conf.*, 2010, pp. 1–11, doi: 10.5244/c.24.31.

[24] J. M. Irvine and E. Nelson, ''Image quality and performance modeling for automated target detection,'' in *Automatic Target Recognition*, F. A. Sadjadi and A. Mahalanobis, Eds. Bellingham, WA, USA: SPIE, 2009, doi: 10.1117/12.818593.

[25] M. Dai, S. Cheng, X. He, and D. Wang, ''Object tracking in the presence of shaking motions,'' *Neural Comput. Appl.*, vol. 31, no. 10, pp. 5917–5934, Oct. 2019, doi: 10.1007/s00521-018-3387-3.

[26] Q. Guo, W. Feng, R. Gao, Y. Liu, and S. Wang, ''Exploring the effects of blur and deblurring to visual object tracking,'' *IEEE Trans. Image Process.*, vol. 30, pp. 1812–1824, 2021, doi: 10.1109/TIP.2020.3045630.

[27] Y. Fang, Y. Yuan, L. Li, J. Wu, W. Lin, and Z. Li, ''Performance evaluation of visual tracking algorithms on video sequences with quality degradation,'' *IEEE Access*, vol. 5, pp. 2430–2441, 2017, doi: 10.1109/ACCESS.2017.2666218.

[28] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, ''Incremental learning for robust visual tracking,'' *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008, doi: 10.1007/s11263-007-0075-7.

[29] K. Zhang, L. Zhang, and M.-H. Yang, ''Real-time compressive tracking,'' in *Computer Vision*. Berlin, Germany: Springer, 2012, pp. 864–877, doi: 10.1007/978-3-642-33712-3_62.

[30] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, ''Exploiting the circulant structure of tracking-by-detection with kernels,'' in *Computer Vision*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 702–715, doi: 10.1007/978-3-642-33765-9_50.

[31] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, ''Fast visual tracking via dense spatio-temporal context learning,'' in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 127–141, doi: 10.1007/978-3-319-10602-1_9.

[32] B. Ma, L. Huang, J. Shen, L. Shao, M.-H. Yang, and F. Porikli, ''Visual tracking under motion blur,'' *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5867–5876, Dec. 2016, doi: 10.1109/TIP.2016.2615812.

[33] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, ''Understanding and diagnosing visual tracking systems,'' in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3101–3109, doi: 10.1109/ICCV.2015.355.

[34] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, ''High-speed tracking with kernelized correlation filters,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015, doi: 10.1109/TPAMI.2014.2345390.

[35] Y. Li and J. Zhu, ''A scale adaptive kernel correlation filter tracker with feature integration,'' in *Computer Vision*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Cham, Switzerland: Springer, 2015, pp. 254–265, doi: 10.1007/978-3-319-16181-5_18.

[36] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, ''Discriminative scale space tracking,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2016, doi: 10.1109/TPAMI.2016.2609928.

[37] H. K. Galoogahi, A. Fagg, and S. Lucey, ''Learning background-aware correlation filters for visual tracking,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1152, doi: 10.1109/ICCV.2017.129.

[38] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, ''ECO: Efficient convolution operators for tracking,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939, doi: 10.1109/CVPR.2017.733.

[39] H. Nam and B. Han, ''Learning multi-domain convolutional neural networks for visual tracking,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302, doi: 10.1109/CVPR.2016.465.

[40] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, ''Fully-convolutional siamese networks for object tracking,'' in *Computer Vision*, G. Hua and H. Jégou, Eds. Cham, Switzerland: Springer, 2016, pp. 850–865, doi: 10.1007/978-3-319-48881-3_56.

[41] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. H. Lau, and M.-H. Yang, ''VITAL: VIsual tracking via adversarial learning,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8990–8999, doi: 10.1109/CVPR.2018.00937.

[42] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, ''SiamRPN++: Evolution of Siamese visual tracking with very deep networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4277–4286, doi: 10.1109/CVPR.2019.00441.

[43] Z. Zhang and H. Peng, ''Deeper and wider Siamese networks for real-time visual tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4586–4595, doi: 10.1109/CVPR.2019.00472.

[44] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, ''Learning discriminative model prediction for tracking,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6181–6190, doi: 10.1109/ICCV.2019.00628.

[45] M. Danelljan, L. Van Gool, and R. Timofte, ''Probabilistic regression for visual tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7181–7190, doi: 10.1109/CVPR42600.2020.00721.

[46] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, ''Staple: Complementary learners for real-time tracking,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 1401–1409, doi: 10.1109/CVPR.2016.156.

[47] M. Mueller, N. Smith, and B. Ghanem, ''Context-aware correlation filter tracking,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1387–1395, doi: 10.1109/CVPR.2017.152.

[48] A. Lukezic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, ''Discriminative correlation filter with channel and spatial reliability,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2017, pp. 4847–4856, doi: 10.1109/CVPR.2017.515.

[49] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, ''DeblurGAN: Blind motion deblurring using conditional adversarial networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Oct. 2018, pp. 8183–8192, doi: 10.1109/CVPR.2018.00854.

[50] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, ''Scale-recurrent network for deep image deblurring,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Dec. 2018, pp. 8174–8182, doi: 10.1109/CVPR.2018.00853.

[51] S. Nah, T. H. Kim, and K. M. Lee, ''Deep multi-scale convolutional neural network for dynamic scene deblurring,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 257–265, doi: 10.1109/CVPR.2017.35.

[52] M. Noroozi, P. Chandramouli, and P. Favaro, ''Motion deblurring in the wild,'' in *Pattern Recognition*, V. Roth and T. Vetter, Eds. Cham, Switzerland: Springer, 2017, pp. 65–77, doi: 10.1007/978-3-319-66709-6_6.

[53] J. Pan, H. Bai, and J. Tang, ''Cascaded deep video deblurring using temporal sharpness prior,'' in *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2020, pp. 3040–3048, doi: 10.1109/CVPR42600.2020.00311.

[54] L. Huang, Y. Xia, and T. Ye, ''Effective blind image deblurring using matrix-variable optimization,'' *IEEE Trans. Image Process.*, vol. 30, pp. 4653–4666, 2021, doi: 10.1109/TIP.2021.3073856.

[55] R. G. Nieto, H. D. B. Restrepo, and I. Cabezas, ''How video object tracking is affected by in-capture distortions?'' in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2227–2231, doi: 10.1109/ICASSP.2019.8683625.

[56] R. Gomez-Nieto, H. D. Benitez-Restrepo, and J. F. Ruiz-Munoz, ''Quality aware feature selection for video object tracking,'' *Electron. Imag.*, vol. 2020, no. 9, pp. 1–7, 2020, doi: 10.2352/ISSN.2470-1173.2020.9.IQSP-169.

[57] C. A. A. Franco and S. B. Vergara, ''Benchmarking of stateof-the-art single object video trackers in authentically distorted videos-Pontificia Universidad Javeriana, Cali,'' Pontificia Universidad Javeriana Colombia, Bogotá, Colombia, Tech. Rep., 2020.

[58] M. Kristan, J. Matas, A. Leonardis, and M. Felsberg, ''The visual object tracking VOT2015 challenge results,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Feb./Dec. 2015, pp. 564–586, doi: 10.1109/ICCVW.2015.79.

[59] Y. Wu, J. Lim, and M. H. Yang, ''Object tracking benchmark,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015, doi: 10.1109/TPAMI.2014.2388226.

[60] M. Mueller, N. Smith, and B. Ghanem, ''A benchmark and simulator for UAV tracking,'' in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 445–461, doi: 10.1007/978-3-319-46448-0_27.

[61] P. Liang, E. Blasch, and H. Ling, ''Encoding color information for visual tracking: Algorithms and benchmark,'' *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015, doi: 10.1109/TIP.2015.2482905.

[62] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan, ''NUS-PRO: A new visual tracking challenge,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 335–349, Feb. 2015, doi: 10.1109/TPAMI.2015.2417577.

[63] H. Fan, H. Ling, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, and C. Liao, ''LaSOT: A high-quality benchmark for large-scale single object tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378, doi: 10.1109/CVPR.2019.00552.

[64] I. Bezzine, Z. A. Khan, A. Beghdadi, N. Al-Maadeed, M. Kaaniche, S. Al-Maadeed, A. Bouridane, and F. A. Cheikh, ''Video quality assessment dataset for smart public security systems,'' in *Proc. IEEE 23rd Int. Multitopic Conf. (INMIC)*, Nov. 2020, pp. 1–5, doi: 10.1109/INMIC50486.2020.9318149.

[65] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, ''TrackingNet: A large-scale dataset and benchmark for object tracking in the wild,'' in *Computer Vision*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 310–327, doi: 10.1007/978-3-030-01246-5_19.

[66] L. Huang, X. Zhao, and K. Huang, ''GOT-10k: A large high-diversity benchmark for generic object tracking in the wild,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021, doi: 10.1109/TPAMI.2019.2957464.

[67] Q. Liu, Z. He, X. Li, and Y. Zheng, ''PTB-TIR: A thermal infrared pedestrian tracking benchmark,'' *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 666–675, Mar. 2020.

[68] Q. Liu, X. Li, Z. He, C. Li, J. Li, Z. Zhou, D. Yuan, J. Li, K. Yang, N. Fan, and F. Zheng, ''LSOTB-TIR: A large-scale high-diversity thermal infrared object tracking benchmark,'' in *Proc. 28th ACM Int. Conf. Multimedia*, New York, NY, USA, 2020, pp. 3847–3856, doi: 10.1145/3394171.3413922.

[69] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, and H. Wang, ''Learning deep multi-level similarity for thermal infrared object tracking,'' *IEEE Trans. Multimedia*, vol. 23, pp. 2114–2126, Jul. 2020.

[70] Darkpgmr. (2020). *Darklabel—Video/Image Labeling and Annotation Tool*. [Online]. Available: https://github.com/darkpgmr/DarkLabel

[71] M. A. Saad, A. C. Bovik, and C. Charrier, ''Blind prediction of natural video quality,'' *IEEE Trans. Image Process.*, vol. 23, no. 3, pp. 1352–1365, Mar. 2014, doi: 10.1109/TIP.2014.2299154.

[72] M. Kristan, ''The visual object tracking VOT2017 challenge results,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2017, pp. 1949–1972, doi: 10.1109/ICCVW.2017.230.

[73] J. Ning, J. Yang, S. Jiang, L. Zhang, and M.-H. Yang, ''Object tracking via dual linear structured SVM and explicit feature map,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4266–4274, doi: 10.1109/CVPR.2016.462.

[74] Z. He, Y. Fan, J. Zhuang, Y. Dong, and H. Bai, ''Correlation filters with weighted convolution responses,'' in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Dec. 2017, pp. 1992–2000, doi: 10.1109/ICCVW.2017.233.

[75] T. Kokul, C. Fookes, S. Sridharan, A. Ramanan, and U. A. J. Pinidiyaarachchi, ''Gate connected convolutional neural network for object tracking,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2602–2606, doi: 10.1109/ICIP.2017.8296753.

[76] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, ''Multicue correlation filters for robust visual tracking,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4844–4853, doi: 10.1109/CVPR.2018.00509.

[77] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, ''Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking,'' *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5596–5609, Nov. 2019, doi: 10.1109/TIP.2019.2919201.

[78] S. Bai, Z. He, Y. Dong, and H. Bai, ''Multi-hierarchical independent correlation filters for visual tracking,'' in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2020, pp. 1–6, doi: 10.1109/ICME46284.2020.9102759.

[79] M. Che, R. Wang, Y. Lu, Y. Li, H. Zhi, and C. Xiong, ''Channel pruning for visual tracking,'' in *Computer Vision*, L. Leal-Taixé and S. Roth, Eds. Cham, Switzerland: Springer, 2019, pp. 70–82, doi: 10.1007/978-3-030-11009-3_3.

[80] H. Lee and D. Kim, ''Salient region-based online object tracking,'' in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1170–1177, doi: 10.1109/WACV.2018.00133.

[81] L. Yang, R. Liu, D. Zhang, and L. Zhang, ''Deep location-specific tracking,'' in *Proc. 25th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2017, pp. 1309–1317, doi: 10.1145/3123266.3123381.

[82] L. Čehovin, A. Leonardis, and M. Kristan, ''Visual object tracking performance measures revisited,'' *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1261–1274, Mar. 2016, doi: 10.1109/TIP.2016.2520370.

[83] R. R. Varior, G. Wang, J. Lu, and T. Liu, ''Learning invariant color features for person reidentification,'' *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3395–3410, Jul. 2016, doi: 10.1109/TIP.2016.2531280.

[84] J. Jiarpakdee, C. K. Tantithamthavorn, H. K. Dam, and J. Grundy, ''An empirical study of model-agnostic techniques for defect prediction models,'' *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 166–185, Jan. 2022, doi: 10.1109/TSE.2020.2982385.

[85] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, ''Learning spatiotemporal features with 3D convolutional networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497, doi: 10.1109/ICCV.2015.510.

[86] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, ''Beyond short snippets: Deep networks for video classification,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2015, pp. 4694–4702, doi: 10.1109/CVPR.2015.7299101.

[87] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, ''Large-scale video classification with convolutional neural networks,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2014, pp. 1725–1732, doi: 10.1109/CVPR.2014.223.

[88] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, ''The unreasonable effectiveness of deep features as a perceptual metric,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595, doi: 10.1109/CVPR.2018.00068.

[89] A. Dosovitskiy and T. Brox, ''Generating images with perceptual similarity metrics based on deep networks,'' in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2016, pp. 658–666, doi: 10.5555/3157096.3157170.

[90] J. Johnson, A. Alahi, and L. Fei-Fei, ''Perceptual losses for real-time style transfer and super-resolution,'' in *Computer Vision*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 694–711, doi: 10.1007/978-3-319-46475-6_43.

[91] P.-H. Chen, R.-E. Fan, and C.-J. Lin, ''A study on SMO-type decomposition methods for support vector machines,'' *IEEE Trans. Neural Netw.*, vol. 17, no. 4, pp. 893–908, Jul. 2006, doi: 10.1109/TNN.2006.875973.

[92] L. Janowski and P. Romaniak, ''QoE as a function of frame rate and resolution changes,'' in *Future Multimedia Networking*, S. Zeadally, E. Cerqueira, M. Curado, and M. Leszczuk, Eds. Berlin, Germany: Springer, 2010, pp. 34–45, doi: 10.1007/978-3-642-13789-1_4.

[93] Z. Xu, R. Hu, J. Chen, H. Li, and H. Chen, ''How much bandwidth does surveillance system require?'' in *Proc. Int. Conf. Image Process. (ICIP)*, 2015, pp. 1762–1766, doi: 10.1109/ICIP.2015.7351103.

[94] J. Korhonen, ''Two-level approach for no-reference consumer video quality assessment,'' *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5923–5938, Dec. 2019, doi: 10.1109/TIP.2019.2923051.

[95] B. Yan, X. Zhang, D. Wang, H. Lu, and X. Yang, ''Alpha-Refine: Boosting tracking performance by precise bounding box estimation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 5289–5298.

[96] D. Yuan, N. Fan, and Z. He, ''Learning target-focusing convolutional regression model for visual object tracking,'' *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105526, doi: 10.1016/j.knosys.2020.105526.

[97] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, and P. Prettenhofer, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, no. 10, pp. 2825–2830, Jul. 2017.

[98] Z. Luo, P.-M. Jodoin, S.-Z. Su, S.-Z. Li, and H. Larochelle, ''Traffic analytics with low-frame-rate videos,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 4, pp. 878–891, Apr. 2018, doi: 10.1109/TCSVT.2016.2632439.

[99] A. Tsifouti, ''Image usefulness of compressed surveillance footage with different scene contents,'' Ph.D. dissertation, Univ. Westminster, London, U.K., 2016.

[100] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[101] A. Paszke, S. Gross, and F. Massa, ''PyTorch: An imperative style, high-performance deep learning library,'' in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, Eds. Red Hook, NY, USA: Curran Associates, 2019.

[102] I. Jung, J. Son, M. Baek, and B. Han, ''Real-time MDNet,'' in *Computer Vision*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 89–104, doi: 10.1007/978-3-030-01225-0_6.

[103] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, ''ATOM: Accurate tracking by overlap maximization,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4660–4669, doi: 10.1109/CVPR.2019.00479.

[104] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, and A. Karpathy, ''ImageNet large scale visual recognition challenge,'' *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[105] 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,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2017, pp. 7464–7473, doi: 10.1109/CVPR.2017.789.

[106] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, ''Microsoft COCO: Common objects in context,'' in *Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[107] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, ''MobileNets: Efficient convolutional neural networks for mobile vision applications,'' 2017, *arXiv:1704.04861*.

[108] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Oct. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[109] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. New York, NY, USA: Now, 2011, doi: 10.1561/2200000016.

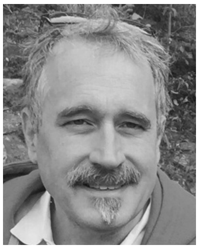
ROGER GOMEZ-NIETO (Member, IEEE) received the B.S. degree in electronics engineering from the Universidad del Quindio, Colombia, in 2014, and the M.S. degree in electrical engineering from the Universidad Tecnologica de Pereira, Risaralda, Colombia, in 2016. He is currently pursuing the Ph.D. degree in engineering and applied sciences with the Pontificia Universidad Javeriana (Pontifical Xavierian University), Colombia. His research interests include image processing, computer vision, and deep learning.

JOSÉ FRANCISCO RUIZ-MUÑOZ received the bachelor's degree in electronic engineering, the M.Eng. degree in industrial automation, and the Ph.D. degree in engineering from the Universidad Nacional de Colombia, Manizales, Colombia, in 2010, 2012, and 2017, respectively. In 2017, he was a full-time Lecturer at the Instituto Tecnologico Metropolitano, Medellin, Colombia. He was a Postdoctoral Associate at the Machine Learning and Sensing Laboratory, University of Florida, Gainesville, FL, USA, from 2017 to 2021. In 2021, he joined the Universidad Nacional de Colombia Sede de La Paz, as an Assistant Professor. His main research interests include machine learning and digital signal processing and their applications in image classification, audio recognition, time-series analysis, and video tracking.

JUAN BERON received the B.S. degree (Hons.) in electronics engineering and the B.S. degree (Hons.) in applied mathematics for his undergraduate projects and academic performance from the Pontificia Universidad Javeriana, Cali, Colombia, in 2019 and 2020, respectively. His main research interests include machine learning, deep learning, computer vision, image processing, and video processing.

CÉSAR A. ARDILA FRANCO received the B.S. degree (Hons.) in electronics engineering to his undergraduate research work from the Pontificia Universidad Javeriana, Cali, Colombia, in 2020. He is currently pursuing the master's degree in data analytics with the Universidad Nacional, Medellín, Colombia. His main research interests include big data analytics, image and video processing, and fraud analysis.

HERNÁN DARÍO BENÍTEZ-RESTREPO (Senior Member, IEEE) received the B.S. degree in electronics engineering from the Pontificia Universidad Javeriana, Cali, Colombia, in 2002, and the Ph.D. degree in electrical engineering from the Universidad del Valle, Cali, in 2008. Since 2008, he has been with the Department of Electronics and Computing, Pontificia Universidad Javeriana Sede Cali. From 2010 to 2014, he was an Adjunct Professor with the Laboratory of Computer Vision and Systems, Université Laval, Québec City, Canada. His main research interests include image and video quality assessment, infrared vision, and digital signal processing. He has been a member of the Scientific Editorial Board of the *Quantitative Infrared Thermography* journal, since 2014. He was a recipient of a Fulbright Visiting Researcher Scholarship to carry out research on video quality assessment with the Laboratory of Image and Video Engineering (LIVE), The University of Texas at Austin, in 2019. In 2011, he received a Merit Scholarship for short-term research from the Ministére de l'Education, du Québec to pursue research on infrared vision at the Laboratory of Computer Vision and Systems, Université Laval. He was the Chair of the Colombia's IEEE Signal Processing, from 2012 to 2017.



ALAN C. BOVIK (Fellow, IEEE) is currently a Cockrell Family Regents Endowed Chair Professor with The University of Texas at Austin. His research interests include image processing, digital television, digital streaming video, and visual perception. He was a recipient of the 2019 Progress Medal from The Royal Photographic Society, the 2019 IEEE Fourier Award, the 2017 Edwin H. Land Medal from the Optical Society of America, the 2015 Primetime Emmy Award for Outstanding Achievement in Engineering Development from the Television Academy, and the Norbert Wiener Society Award and the Karl Friedrich Gauss Education Award from the IEEE Signal Processing Society. A perennial Web of Science Group Highly-Cited Researcher, he has also received about ten best journal paper awards, including the 2016 IEEE Signal Processing Society Sustained Impact Award. His recent books include *The Essential Guides to Image and Video Processing*. He co-founded and was the longest-serving Editor-in-Chief of the IEEE Transactions on Image Processing, and also created/chaired the IEEE International Conference on Image Processing which was first held in Austin, TX, USA, in 1994.