https://doi.org/10.31891/2307-5732-2026-365-82
УДК 637.5.02

NAUMENKO VITALII
National Aerospace University – KhAI
https://orcid.org/0009-0005-8426-6635
e-mail: v.m.naumenko@khai.edu
NAUMENKO VIKTORIIA
National Aerospace University – KhAI
https://orcid.org/0000-0002-5291-6032
e-mail: v.naumenko@khai.edu
ABRAMOV SERGIY
National Aerospace University – KhAI
https://orcid.org/0000-0002-8295-9439
e-mail: s.abramov@khai.edu
LUKIN VOLODYMYR
National Aerospace University – KhAI
https://orcid.org/0000-0002-1443-9685
e-mail: v.lukin@khai.edu

COMPARATIVE ANALYSIS OF LIGHTWEIGHT TRACKERS UNDER CORRELATED AND UNCORRELATED NOISE

The research focuses on the robustness of lightweight single-object trackers that can operate in (near) real time on weak CPUs such as Raspberry Pi 4 and 5 when the input video is corrupted by noise. The goal is to compare how different families of embedded-friendly trackers – correlation-filter, Siamese and transformer-based – degrade under additive white Gaussian noise (AWGN) and salt-and-pepper noise (S&P), each applied in four combinations of temporal and channel-wise correlation, and to identify algorithms that offer the best robustness–efficiency trade-off for embedded vision systems. The results show that all trackers exhibit monotonic AUC degradation as noise intensity increases, with Gaussian noise causing only moderate accuracy loss and salt-and-pepper noise leading to much steeper performance drops. HiT, ViTTrack and NanoTrack v3 achieve the highest accuracy on clean or mildly degraded data; however, HiT and NanoTrack v3 degrade faster than NanoTrack v2 and are overtaken by NanoTrack v2 at medium and high noise levels in all settings except temporally correlated AWGN for NanoTrack v3. ViTTrack is the only tracker that consistently maintains higher AUC than NanoTrack v2 across all noise intensities and correlation regimes, providing the most stable performance. ECO and DaSiamRPN lose accuracy much faster, especially under salt-and-pepper noise. Salt-and-pepper noise is generally more destructive than AWGN, while the temporal and channel-wise correlation of noise causes noticeable but moderate shifts in the degradation profiles without changing the overall ranking of trackers.

***Keywords**: visual object tracking, single object tracking, AWGN noise, salt and pepper noise.*

НАУМЕНКО ВІТАЛІЙ, НАУМЕНКО ВІКТОРІЯ, АБРАМОВ СЕРГІЙ, ЛУКІН ВОЛОДИМИР
Національний аерокосмічний університет «ХАІ»

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЛЕГКИХ ТРЕКЕРІВ У УМОВАХ КОРЕЛЬОВАНОГО ТА НЕКОРЕЛЬОВАНОГО ШУМУ

Дослідження зосереджується на надійності легких трекерів одного об'єкта, які можуть працювати в (майже) реальному часі на слабких процесорах, таких як у Raspberry Pi 4 і 5, коли вхідне відео пошкоджене шумом. Метою є порівняння того, як різні сімейства вбудованих трекерів – кореляційний фільтр, на основі сіамських мереж і на основі трансформера – погіршують якість зображення під впливом адитивного білого гауссового шуму (AWGN) і шуму «сіль і перець» (S&P), кожен з яких застосовується в чотирьох комбінаціях темпоральної та канальної кореляції, а також визначення алгоритмів, які забезпечують найкращий компроміс між надійністю та ефективністю для вбудованих систем зору. Результати показують, що всі трекери демонструють монотонне погіршення AUC із збільшенням інтенсивності шуму, причому гаусівський шум спричиняє лише помірну втрату точності, а шум типу «сіль і перець» призводить до набагато більш різкого падіння ефективності. HiT, ViTTrack і NanoTrack v3 досягають найвищої точності на чистих або злегка спотворених даних; однак HiT і NanoTrack v3 погіршуються швидше, ніж NanoTrack v2, і поступаються NanoTrack v2 при середньому та високому рівнях шуму в усіх налаштуваннях, крім темпорально корельованого AWGN для NanoTrack v3. ViTTrack є єдиним трекером, який стабільно підтримує вищий AUC, ніж NanoTrack v2, для всіх рівнів шуму та режимів кореляції, забезпечуючи найстабільнішу роботу. ECO та DaSiamRPN втрачають точність набагато швидше інших, особливо в умовах шуму типу «сіль і перець». Шум типу «сіль і перець» зазвичай є більш руйнівним, ніж AWGN, тоді як темпоральна та канальна кореляція шуму спричиняє помітні, але помірні зміни в профілях погіршення якості без зміни загального рейтингу трекерів.

***Ключові слова**: візуальне відстеження об'єктів, відстеження одного об'єкта, шум AWGN, шум «сіль і перець»*

Стаття надійшла до редакції / Received 09.03.2026
Прийнята до друку / Accepted 11.04.2026
Опубліковано / Published 28.25.2026

Problem overview

Visual object tracking (VOT) is a fundamental problem in computer vision. Given the initial position of a target in the first frame, a tracker must estimate its state in subsequent frames under a wide range of appearance changes. VOT is a key component of many real-world systems, including autonomous driving, UAV-based inspection, robotics, intelligent video surveillance, and augmented reality, where it is often embedded into larger perception and control pipelines. In many of these use cases, the tracking algorithm has to operate on inexpensive embedded hardware, with

limited CPU resources, tight power budgets, and often without access to a GPU. Ensuring robust tracking under such constraints is therefore a key challenge. On the other hand, in some safety-critical scenarios, such as driver-assistance or UAV navigation, tracking failures caused by adverse imaging conditions may directly translate into unsafe behaviour. This makes not only raw accuracy, but also robustness to realistic image degradations, an essential property of practical trackers.

By focusing on trackers that can operate in near real time on resource-constrained processors, our research complements previous reliability-oriented work that targets high-performance systems. It provides practitioners with empirical guidance on which tracking paradigm offers the best compromise between reliability and efficiency in noisy environments, and indicates where additional architectural or algorithmic improvements are most needed for reliable tracking on embedded devices.

Analysis of recent sources in visual object tracking methods

From a modelling viewpoint, single-object trackers are commonly grouped into generative and discriminative approaches (see Fig.1).

Generative object trackers build an explicit appearance model of the target and search for image regions that best reconstruct this model. A major line of work is sparse-representation tracking, where the target is expressed as a sparse linear combination of dictionary atoms. Structural and context-aware sparse trackers – such as Structural Sparse Tracking (SST) [1], Robust Structural Sparse Tracking (RSST) [2] and sparsity-based collaborative models (SCM) [3] – minimise reconstruction error under sparsity constraints.

A second branch uses subspace models, assuming that target appearances lie in a low-dimensional manifold. Incremental PCA-based trackers [4] and adaptive eigenbasis trackers [5] update this subspace online and measure reconstruction error after projecting candidate regions onto it.

Finally, generative modelling has also been combined with Bayesian state estimation. A representative example is the context-aware exclusive sparse tracker (CEST) [6], which formulates target, background and contextual regions within a unified exclusive sparse representation model.

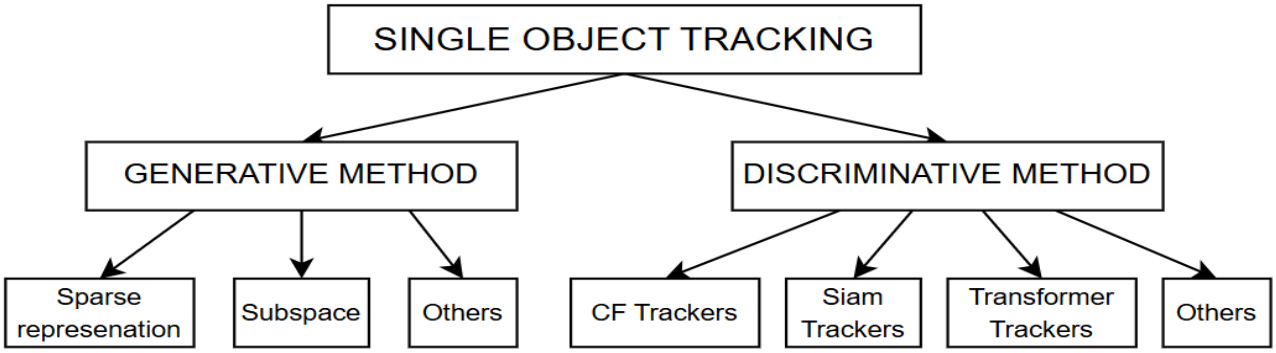

Fig. 1. Tracking methods classification

Discriminative trackers learn a classifier or regressor that separates the target from the background and then predict the most likely target state in each frame. Within this family, one of the most influential paradigms are discriminative correlation filters (DCF) [7, 8]. Early DCF-based trackers such as MOSSE [7] and KCF [9] achieve high frame rates by learning a convolution filter in the Fourier domain and exploiting fast frequency-domain operations. Later, CCOT (Continuous Convolution Operators for Tracking) [10] introduced continuous-domain convolution operators on multi-resolution feature maps, significantly improving accuracy but at a high computational cost. ECO (Efficient Convolution Operators for Tracking) [11] builds directly on CCOT and reduces its memory and runtime complexity by factorising the filters and using a compact generative model of training samples, while preserving most of the accuracy gains.

A second major group of discriminative methods are Siamese trackers [12]. Classical Siamese trackers such as SiamFC [13] and SiamRPN [14, 15] employ a two-branch network that processes the template and the search region and outputs a similarity map or bounding-box predictions. Trackers such as DaSiamRPN [16] extend this idea with a region proposal head and distractor-aware training, which improves robustness in cluttered scenes.

Subsequent work moved from pure Siamese matching to deeper discriminative architectures that still rely on a template–search formulation. ATOM (Accurate Tracking by Overlap Maximization) [17] learns an online classification head together with an IoU prediction network, explicitly optimising the predicted overlap between the estimated and ground-truth boxes. OSTrack [18] further merges the Siamese idea with vision transformers, using a unified transformer backbone that jointly encodes template and search tokens and directly regresses the target box.

Another important direction focuses on lightweight Siamese trackers for resource-constrained devices. SiamBAN [19] introduces an anchor-free Siamese box-adaptive network that predicts the target bounding box in a dense, one-stage manner, improving both accuracy and efficiency for real-time tracking on modern hardware. LightTrack [20] further explores efficient backbones and heads tailored for mobile platforms. NanoTrack v2 and NanoTrack v3 follow this line of work and implement ultra-compact MobileNet-like backbones with anchor-free heads and depthwise correlation, providing accurate tracking with real-time or near–real-time performance on CPU-only and embedded hardware, despite not being accompanied by a separate research paper.

In recent years, transformer-based trackers have emerged as a third important branch of discriminative methods. They use self- and cross-attention to jointly encode the template and search image and to capture long-range dependencies

in the scene. Early works such as TrTr [21] and STARK [22] introduced transformer encoder–decoder architectures for tracking, while more recent lightweight vision transformers, ViTTrack and the hierarchical HiT [23, 24] tracker, aim to bring the benefits of global attention into edge-friendly models by carefully reducing model size and computation while preserving accuracy.

Parallel to these trends, there is a line of work on extremely lightweight trackers based on perceptual image hashing [25]. Although hash-based trackers are usually less accurate than modern deep trackers, they illustrate the importance of algorithmic simplicity and tight compute constraints for embedded platforms [26].

In many robustness studies on image and video analysis, additive white Gaussian noise (AWGN) [27, 28] and salt-and-pepper (S&P) noise are used as simple but informative surrogate models of real acquisition and transmission artefacts. AWGN is a standard approximation for thermal, reset and read-out noise in CCD/CMOS image sensors and for channel noise in digital links [29], whereas S&P noise mimics dead or hot pixels, bit errors and transmission glitches that appear as random black-and-white speckles in individual frames [30, 31]. At the same time, several works have pointed out that real video noise is rarely i.i.d.: it often exhibits temporal correlation due to fixed sensor patterns, slow electronic drift and repeated codec artefacts, as well as correlation between colour channels originating from shared analogue circuitry and demosaicing in Bayer cameras [29, 32]. Modern video denoising methods based on non-local spatio-temporal filtering and motion-compensated 3D/4D transforms explicitly exploit these correlations instead of assuming frame-wise independent noise [33, 34]. These observations suggest that robustness of visual trackers should be analysed not only under i.i.d. AWGN and S&P noise, but also under noise models with controlled temporal and inter-channel correlation.

While benchmark datasets like UAV123 [35], LaSOT [36] and GOT-10k [37] have driven substantial progress in tracker design and evaluation, they mainly focus on “clean” or slightly degraded imagery. However, real-world acquisition pipelines often suffer from sensor noise, compression artifacts, transmission errors and hardware faults. Earlier work introduced the Visual Object Tracking – Robustness Toolkit (VOT-RT) [38], which distorts existing datasets with configurable noise types such as additive Gaussian noise and salt-and-pepper noise, and evaluates how state-of-the-art trackers degrade under such conditions.

More recently, the robust tracking module (RTM) [39] extended this line of research by inserting a lightweight denoising network in front of existing trackers and showing that a learned pre-processing stage can significantly mitigate the impact of diverse noise types while also providing an open robustness-evaluation toolkit.

Despite these advances, several gaps remain. First, most robustness studies focus on relatively heavy state-of-the-art trackers intended for GPU-equipped desktops, whereas many emerging applications—low-cost UAVs, Raspberry Pi-based robots, automotive prototypes—require trackers that run in (near) real time on a single low-power CPU core. Second, prior work mainly treats robustness as an add-on (e.g., via a denoising module), leaving the comparative behaviour of different tracking paradigms under noisy conditions insufficiently understood.

Analysis of recent sources in Lightweight single object trackers

In this section we briefly review the six trackers analysed in this work. They were chosen as representative members of three popular families of modern SOT methods – correlation-filter, Siamese-network and transformer-based trackers – for which open implementations and lightweight CPU-oriented models are available. All of them can be deployed on Raspberry-Pi-class hardware with (near) real-time performance.

ECO is a correlation-filter SOT tracker and the successor of CCOT. It learns continuous convolution operators in the Fourier domain on multi-layer features that combine hand-crafted descriptors (HOG, colour names) with CNN features. Filter factorisation and a compact generative model of training samples based on Gaussian mixture models (GMM) significantly reduce memory usage and computational cost. Target scale is handled by an additional DSST module. ECO is widely regarded as a strong baseline that offers high robustness at moderate computational complexity and serves here as a representative of advanced DCF-based methods.

DaSiamRPN is a Siamese, anchor-based tracker with a region-proposal head. During training it introduces semantic negative samples and performs distractor mining in order to suppress objects that are visually similar to the target. At test time the RPN head jointly predicts classification scores and bounding boxes for a dense set of anchors in the search region, that provides option of failure detection and global re-detection. In this study DaSiamRPN represents early high-performance Siamese trackers that are still relatively compact and can be run on modern CPUs.

NanoTrack v2 is an ultra-lightweight anchor-free Siamese tracker designed specifically for embedded devices. Its implementation is split into two small ONNX models: a tiny MobileNetV3-style backbone that extracts features from the template and search image, and a neck/head that performs depthwise cross-correlation and outputs a response map together with bounding-box offsets. The post-processing pipeline relies on simple but effective heuristics such as a scale penalty and a Hann window for spatial smoothing.

NanoTrack v3 is an evolution of NanoTrack v2. It keeps the same overall two-stage ONNX architecture and post-processing strategy, but employs a slightly larger MobileNetV3 backbone and a refined prediction head. These changes significantly improve the average overlap and success rate on standard benchmarks while retaining some computational footprint. NanoTrack v3 therefore represents a new generation of ultra-compact Siamese trackers that push the accuracy of lightweight models closer to that of heavier networks.

ViTTrack is a lightweight Vision Transformer-based tracker. Unlike Siamese CNNs with separate branches, ViTTrack jointly encodes the template and search region in a single transformer encoder and directly regresses the target bounding box and a confidence score in one forward pass. ViTTrack aims to balance the favourable modelling capabilities of transformers with strict constraints on model size, memory usage and latency on embedded devices.

HiT is an efficient hierarchical transformer tracker. It uses a multi-stage architecture with a bridge mechanism that injects high-level semantic information into shallow feature maps, as well as shared positional encodings for template and search tokens. This design allows HiT to capture long-range dependencies while preserving high spatial resolution in the final layers.

Formulation of the article's objectives

The objective of this research is to evaluate the robustness and performance of several lightweight single-object trackers under correlated and uncorrelated image noise. The study aims to identify which tracker architectures are most suitable for use in embedded vision systems running on Raspberry-Pi-class hardware, where both computational resources and input video quality can be limited.

Presentation of the main material

Since the focus of this work is on robustness to noise rather than runtime benchmarking, all experiments are carried out on a dual-socket Linux server equipped with two Intel Xeon E5-2680 v4 CPUs with 128GB RAM and 2TB free storage to be able to process the full set of noise-augmented dataset.

In this study, we use the official validation split of GOT-10k. No retraining or fine-tuning of the trackers is performed on this split. Each sequence contains a single target with dense ground-truth bounding boxes for all frames, which enables the computation of success plots and related metrics.

In practical imaging pipelines, additive white Gaussian noise and salt-and-pepper noise arise from different stages of acquisition and transmission. AWGN is widely used to model thermal, reset and read-out noise in CCD/CMOS image sensors, as well as electronic interference due to high sensor temperature, poor illumination and analogue circuitry imperfections in the camera front-end. Even when basic denoising or enhancement is applied, residual Gaussian-like fluctuations often remain in the video stream, especially in low-light conditions or on low-power embedded platforms that cannot afford heavy restoration.

Salt-and-pepper noise, in contrast, primarily reflects sparse, high-amplitude disturbances. It can be introduced during analogue-to-digital conversion or digital transmission, where bit errors and buffer overruns produce isolated pixels saturated to the minimum or maximum intensity. Dead or hot sensor pixels and occasional memory failures lead to similar black-and-white outliers in individual frames. Such impulsive artefacts are typically handled by median or more sophisticated edge-preserving filters, but in autonomous systems with strict latency and compute constraints they may go undetected and directly affect downstream tracking. For this reason, many robustness studies model acquisition and channel errors using AWGN and S&P, and we follow the same practice in this work.

In this work we focus on two classical synthetic noise models that are widely used to approximate sensor and transmission artefacts in image and video processing: additive white Gaussian noise and salt-and-pepper noise.

Let $S_t(x, y, c)$ denote the clean RGB frame at time t, spatial position (x, y) and colour channel $c \in \{R, G, B\}$, and let $I_t(x, y, c)$ be the observed noisy frame. We model the observation as

$$I_t(x, y, c) = S_t(x, y, c) + N_t(x, y, c),$$

where $N_t(x, y, c)$ is either Gaussian or impulse noise, depending on the experiment.

For AWGN, we use zero-mean Gaussian noise with variance σ^2. In the idealised i.i.d. case, noise samples are drawn independently for every pixel, frame and channel,

$$N_t(x, y, c) \sim \mathcal{N}(0, \sigma^2).$$

For salt-and-pepper noise we adopt the standard impulse model: each pixel is independently replaced by the minimum or maximum intensity with probability $p/2$ each and remains unchanged with probability $1 - p$. The parameters σ^2 (for AWGN) and p (for salt-and-pepper) control the noise intensity and vary from weak to strong distortion.

To analyse the impact of noise structure, we instantiate both noise types under two temporal regimes and two channel-wise regimes, leading to four distinct noise configurations. In all cases the noise is spatially white (uncorrelated across pixels within each channel); only temporal and inter-channel correlations are varied.

Temporal regimes:

1. *Temporally uncorrelated*: for each frame t we draw a new noise field, so $N_t(\cdot,\cdot,c)$ and $N_{t'}(\cdot,\cdot,c)$ are statistically independent for $t \neq t'$.
2. *Temporally correlated (fixed-pattern)*: for each sequence we draw a single spatial noise realization per channel and reuse it for all frames, $N_t(\cdot,\cdot,c) = N^{(c)}(\cdot,\cdot)$ for all t, which yields perfectly correlated noise along time.

Channel-wise regimes:

1. *Channel-wise uncorrelated*: the three-color channels are corrupted by independent implementations of the same noise process, i.e. $N_t(\cdot,\cdot,R)$, $N_t(\cdot,\cdot,G)$ and $N_t(\cdot,\cdot,B)$ are mutually independent.
2. *Channel-wise correlated: a single noise field is shared across channels, so that for each frame* t and spatial position (x, y) we have

$$N_t^R(x, y) = N_t^G(x, y) = N_t^B(x, y),$$

which simulates colour disturbances caused by the shared use of analogue circuits or the shared processing of RGB components.

Combining these axes, we obtain four noise configurations used in our experiments:

1. Temporally and channel-wise uncorrelated noise (fully i.i.d.): new, independent noise is sampled for each frame and each colour channel.
2. Temporally correlated, channel-wise uncorrelated noise (fixed pattern per channel): each channel has its own fixed noise pattern that is reused for all frames in the sequence.

3. Temporally uncorrelated, channel-wise correlated noise (shared pattern per frame): for each frame a single noise field is drawn and applied identically to all three channels, but the field changes independently from frame to frame.
4. Temporally and channel-wise correlated noise (shared fixed pattern): a single noise field is drawn and applied identically to all channels and all frames of a sequence.

During implementation, all noise fields are generated using pseudo-random number generators, which are deterministically initialized as a function of frame index and channel index, making each noise configuration fully reproducible while maintaining a predictable correlation structure over time and across channels.

For quantitative performance comparison we adopt the standard overlap-based measures commonly used in single-object tracking benchmarks such as GOT-10k. For each frame t we compute the Intersection-over-Union (IoU) between the predicted bounding box ROI_t and the ground-truth box ROI_t^{gt}:

$$\mathrm{IoU} = \frac{\left|\mathrm{ROI}_t \cap \mathrm{ROI}_t^{gt}\right|}{\left|\mathrm{ROI}_t \cup \mathrm{ROI}_t^{gt}\right|}$$

Based on IoU we define the success rate at a fixed IoU threshold τ as

$$S(\tau) = \frac{1}{T}\sum_{t=1}^{T} 1[\mathrm{IoU}_t \geq \tau],$$

where T is the number of frames in the sequence and $1[\cdot]$ is the indicator function.

To obtain a single scalar accuracy measure that is less sensitive to a particular threshold, we consider the success curve $S(\tau)$, which shows the success rate as a function of the IoU threshold $\tau \in [0,1]$. The main metric used in this paper is the Area Under Curve (AUC):

$$\mathrm{AUC} = \int_0^1 S(\tau)d\tau$$

In practice the integral is approximated numerically by evaluating $S(\tau)$ on a discrete set of thresholds and computing the corresponding area. This AUC metric is identical to the success-based overlap score (OS) used in prior robustness studies on visual object tracking and serves as our primary indicator of tracking performance under different noise types and noise-correlation regimes.

All trackers are evaluated in the standard single-object tracking setting:

1. the tracker is initialised in the first frame with the ground-truth bounding box;
2. for all subsequent frames it receives only the current image and its own previous state, without access to ground truth;
3. tracking is run once per sequence for each experimental condition (noise type, channel-correlation regime and noise intensity).

All experiments are managed using a configuration-driven workflow based on Hydra. Each experimental setting—defined by:

- tracker (ECO, DaSiamRPN, NanoTrack v2, NanoTrack v3, ViTTrack, HiT),
- noise type (additive white Gaussian noise, salt-and-pepper noise),
- channel-correlation regime (channel-uncorrelated / channel-correlated),
- noise intensity level,

is encoded as a separate Hydra configuration. This allows us to systematically sweep over the grid of parameters and guarantees reproducibility of all runs.

Tracking runs and evaluation results are logged with MLflow. For every configuration, MLflow stores:

- the experimental parameters (tracker, noise type, correlation regime, noise intensity);
- the aggregated metrics (AUC).

These logs are then used to generate the degradation curves presented in the results section, where we compare how quickly different trackers lose accuracy as the noise intensity increases under different noise types and four temporal/channel-wise noise-correlation regimes.

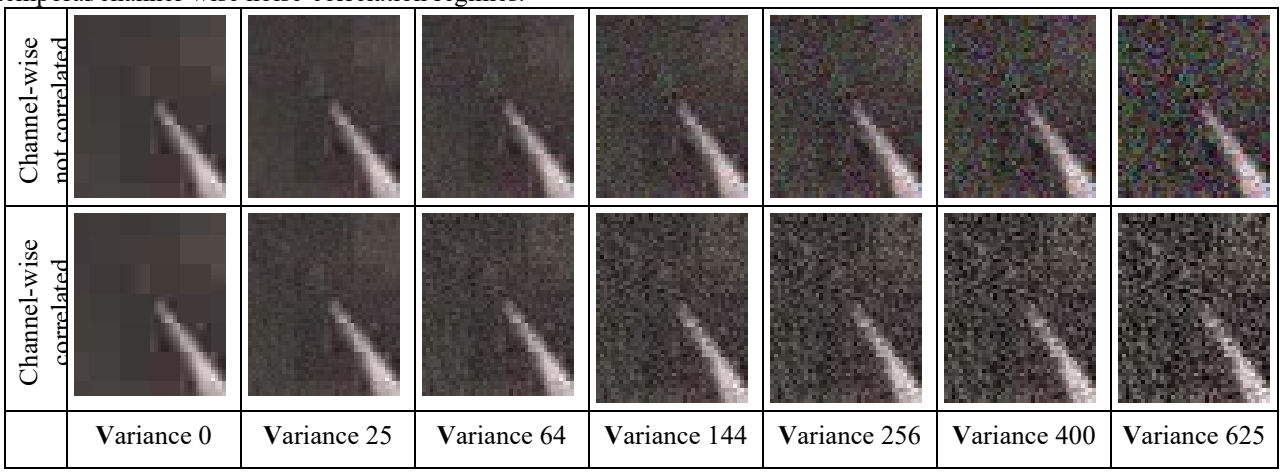

Fig. 2. Example frames of GOT10k dataset with varying levels of White Gaussian Noise in channel-wise uncorrelated and correlated regimes

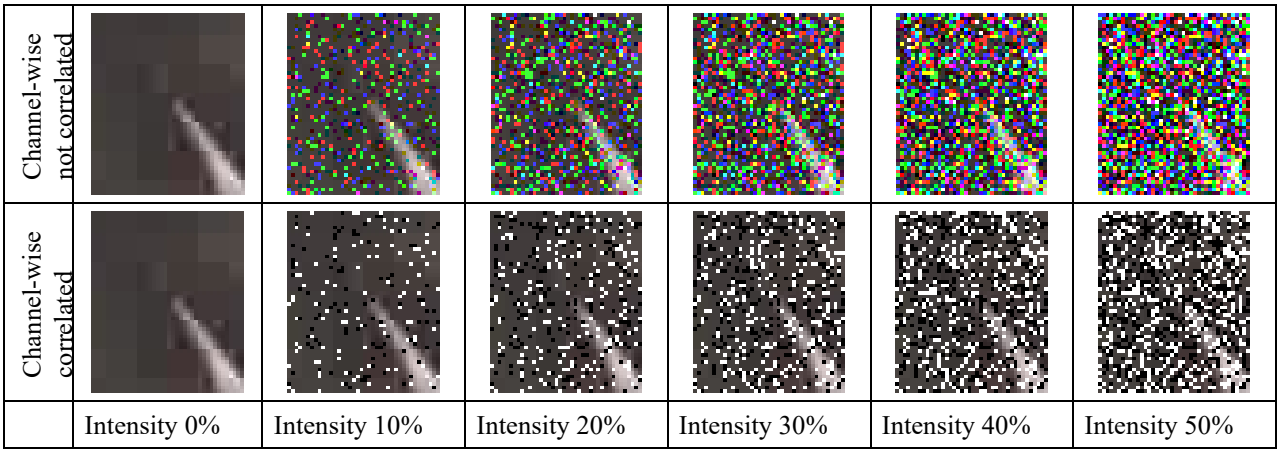

Fig. 3. Example frames of GOT10k dataset with varying levels of Salt and Pepper noise in channel-wise uncorrelated and correlated regimes

Simulation results

Figures 4–7 show the curves of deterioration of the AUC index as a function of noise intensity for additive white Gaussian noise and salt and pepper noise for four correlation modes: temporally correlated/uncorrelated across channels, temporally correlated/correlated across channels, temporally uncorrelated/correlated across channels, and temporally uncorrelated/uncorrelated across channels.

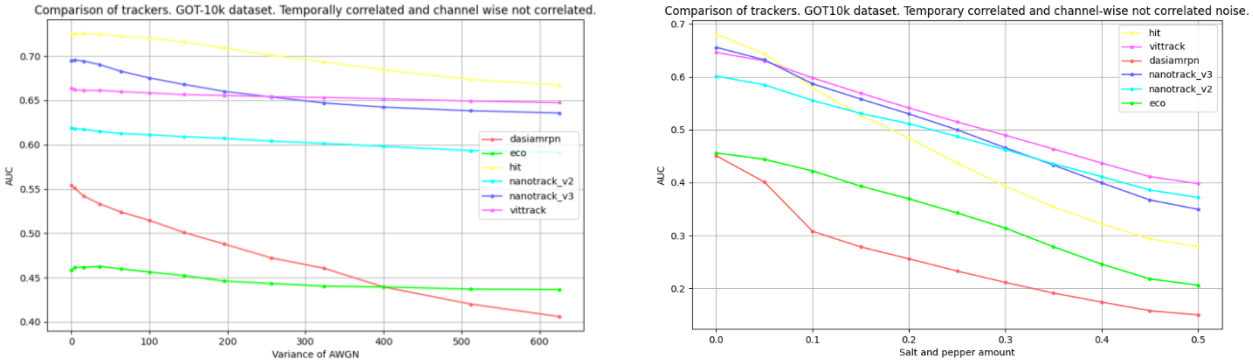

Fig. 4. Success curves for temporally correlated and channel-wise not correlated AWGN and S&P noises

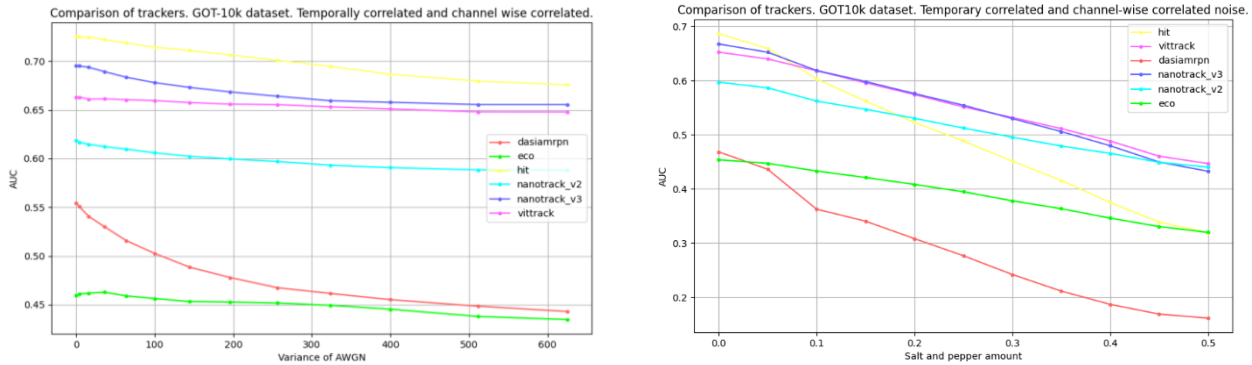

Fig. 5. Success curves for temporally correlated and channel-wise correlated AWGN and S&P noises

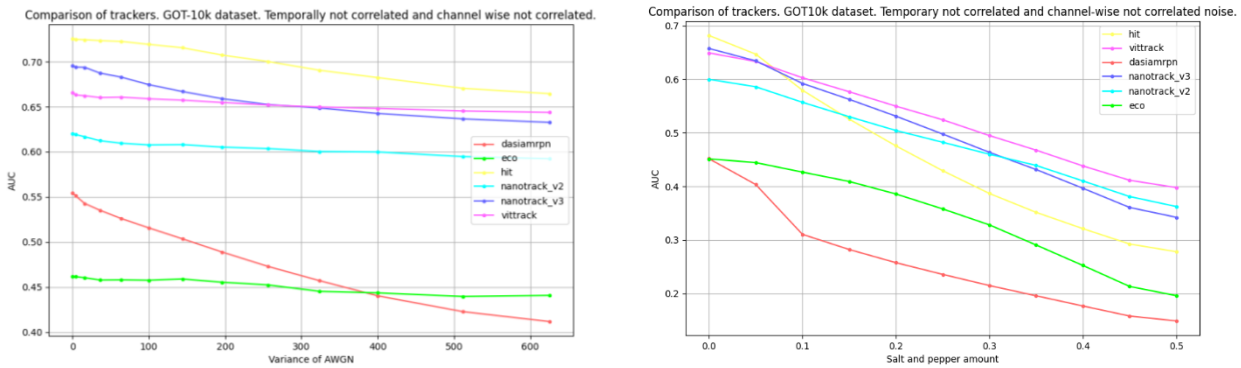

Fig. 6. Success curves for temporally not correlated and channel-wise not correlated AWGN and S&P noises

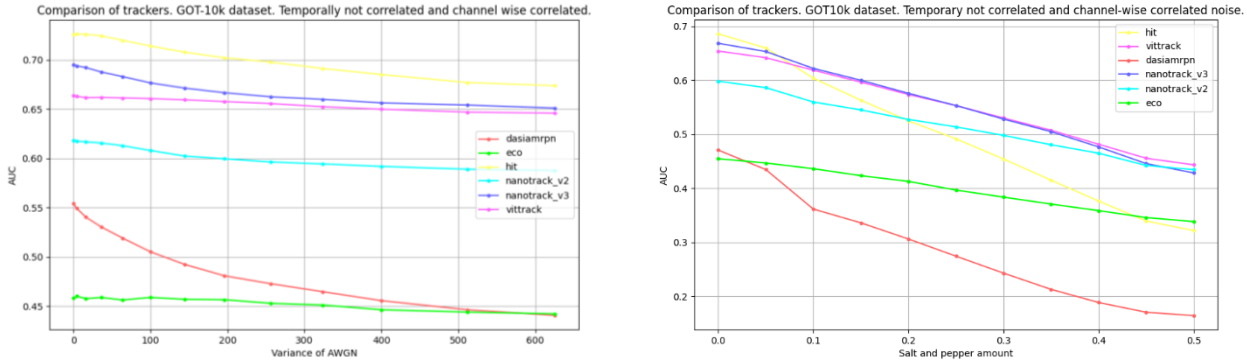

Fig. 7. Success curves for temporally not correlated and channel-wise correlated AWGN and S&P noises

For all trackers and all experimental settings, the AUC curves decrease monotonically with increasing noise levels, confirming the expected sensitivity of tracking accuracy based on overlap to image quality degradation. In the noise-free or very low-noise regime, the ranking of trackers is essentially the same in all figures and for both noise types. HiT, ViTTrack and NanoTrack v3 achieve the highest AUC values with very close scores on GOT-10k. NanoTrack v2 forms the second tier with slightly lower accuracy, while ECO and DaSiamRPN clearly lag behind. This is consistent with the stronger backbones and heads used in HiT, ViTTrack and NanoTrack v3 compared to NanoTrack v2, and with the older architectures of ECO and DaSiamRPN.

For AWGN, the degradation curves are relatively flat. Up to moderate variance, the AUC drop is negligible for all trackers: they retain a significant portion of their performance on clean data even at the highest tested levels of Gaussian noise. Therefore, the differences between trackers are mainly manifested in relative slopes rather than catastrophic failures. HiT and NanoTrack v3 start with the highest AUC, but their curves decline faster than those of NanoTrack v2. The AUC curves for ViTTrack are almost parallel to those for NanoTrack v2 and remain consistently higher for all variations and for all four correlation modes. This indicates that ViTTrack provides a very stable compromise between accuracy and reliability, improving on NanoTrack v2 without the gradual deterioration in quality seen in HiT and NanoTrack v3. ECO and DaSiamRPN form the least reliable group. Under AWGN conditions, their AUC decreases faster than that of NanoTrack and transformer-based trackers, and under high variance, they achieve the lowest accuracy among all methods. This confirms that older correlation filters and early Siamese designs are more sensitive to AWGN noise.

For salt-and-pepper noise, the situation is more serious. The AUC curves are significantly steeper than for AWGN, indicating that impulse noise is much more destructive for all trackers. This affects ECO and DaSiamRPN the most: their AUC drops to a small fraction of the value for clean data already at medium pulse density. NanoTrack v2, NanoTrack v3, ViTTrack, and HiT are more resilient, but still suffer a noticeable loss of AUC when the S&P density approaches 50%. As with AWGN, HiT and NanoTrack v3 start out better than NanoTrack v2 at low noise levels but deteriorate more quickly; in all four correlation modes, NanoTrack v2 outperforms them at medium and high S&P densities. ViTTrack again maintains the best AUC across the entire range of impulse noise levels, with curves that remain above and approximately parallel to those of NanoTrack v2.

For AWGN, the temporally correlated modes (fixed noise pattern over time) typically result in a slightly higher AUC than temporally uncorrelated modes with the same variance. When the same Gaussian model is retained across all frames, part of the distortion is absorbed by the pattern during updates, and the tracker typically treats it as a stable component of the appearance. This effect is most noticeable for ViTTrack, HiT, and NanoTrack v3, whose curves for temporally correlated AWGN are systematically higher than those for temporally uncorrelated AWGN, both in the case of channel correlation and in the absence of correlation.

For S&P noise, the impact of temporal and channel correlation is more ambiguous. In some settings, especially at medium pulse density, temporally correlated S&P noise slightly worsens the AUC compared to the temporally uncorrelated case, as persistent pulses contaminate the pattern and distort the trained model. In other settings, the difference between correlated and uncorrelated modes is negligible; the dominant factor remains the overall pulse density rather than their correlation model. Importantly, in all four correlation modes, the relative order of trackers remains almost unchanged: ViTTrack and NanoTrack v2 consistently form the most reliable pair, HiT and NanoTrack v3 occupy an intermediate position that combines the accuracy of clean data with reliability, and ECO and DaSiamRPN remain the most noise-sensitive methods.

Conclusions

From an embedded-vision perspective, these results lead to a nuanced conclusion. If the expected noise level is low, HiT or NanoTrack v3 may be attractive due to their slightly higher accuracy on clean sequences. If significant noise is expected—for example, in low-light or high-ISO settings, or with noisy transmission channels—NanoTrack v2 and especially ViTTrack become more useful, as they provide higher AUC at medium and high noise intensities across all temporal and channel-wise correlation regimes.

Overall, the results show that there is no single “best” lightweight tracker for all noise levels. ViTTrack provides the most stable ranking across the entire noise range, NanoTrack v2 is surprisingly competitive and often dominates at

high noise, whereas HiT and NanoTrack v3 are strong in the clean/low-noise regime but less robust when noise becomes severe. This highlights the importance of evaluating embedded-friendly trackers not only on clean benchmarks but also across a range of noise intensities.

As future research directions, testing on additional datasets could help understand in which scenarios each lightweight tracker fails first and whether it is possible to generalise the ranking observed on GOT-10k. In embedded systems, noise is often combined with motion blur, low-light noise, colour distortions, compression artefacts, and packet loss. In future work, it may be useful to evaluate trackers under conditions of combined degradation. Although all of the trackers considered are fast enough for Raspberry Pi-class devices, we did not measure frame rate, latency, or power consumption on the device. A promising direction is to combine reliability assessment with real-world hardware profiling.

References

1. T. Zhang, S. Liu, C. Xu, et al., Structural Sparse Tracking, 2015.
2. T. Zhang, C. Xu, and M.H. Yang, Robust Structural Sparse Tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, Vol. 41No. 2, pp. 473–486, 10.1109/TPAMI.2018.2797082.
3. W. Zhong, H. Lu, and M.-H. Yang, Robust object tracking via sparsity-based collaborative model, 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1838–1845, 2012, 10.1109/CVPR.2012.6247882.
4. S. Bai, R. Liu, Z. Su, C. Zhang, and W. Jin, Incremental robust local dictionary learning for visual tracking, Proceedings - IEEE International Conference on Multimedia and Expo, 2014, Vol. 2014, pp. 1–6, 10.1109/ICME.2014.6890262.
5. M.J. Black and A.D. Jepson, EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation, International Journal of Computer Vision, 1998, Vol. 26No. 1, pp. 63–84, 10.1023/A:1007939232436.
6. T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja, Robust Visual Tracking via Exclusive Context Modeling, IEEE transactions on cybernetics, 2015, Vol. 46, 10.1109/TCYB.2015.2393307.
7. D. Bolme, J. Beveridge, B. Draper, and Y. Lui, Visual object tracking using adaptive correlation filters, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2544–2550, 10.1109/CVPR.2010.5539960.
8. M. Danelljan, G. Häger, F. Khan, and M. Felsberg, Accurate Scale Estimation for Robust Visual Tracking, pp. 65.1-65.11, 2014, 10.5244/C.28.65.
9. J. Henriques, R. Caseiro, P. Martins, and J. Batista, High-Speed Tracking with Kernelized Correlation Filters, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, Vol. 37, 10.1109/TPAMI.2014.2345390.
10. M. Danelljan, A. Robinson, F. Khan, and M. Felsberg, Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking, 2016, 10.48550/arXiv.1608.03773.
11. M. Danelljan, G. Bhat, F.S. Khan, and M. Felsberg, ECO: Efficient Convolution Operators for Tracking, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6931–6939, 2017, 10.1109/CVPR.2017.733.
12. L. Bertinetto, J. Valmadre, J. Henriques, A. Vedaldi, and P. Torr, Fully-Convolutional Siamese Networks for Object Tracking, 2016, Vol. 9914, pp. 850–865, 10.1007/978-3-319-48881-3_56.
13. Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, SiamFC++: Towards Robust and Accurate Visual Tracking with Target Estimation Guidelines, Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 12549–12556, 10.1609/aaai.v34i07.6944.
14. B. Li, J. Yan, W. Wu, Z. Zheng, and X. Hu, High Performance Visual Tracking with Siamese Region Proposal Network, 2018, pp. 8971–8980, 10.1109/CVPR.2018.00935.
15. B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks, 2019, pp. 4277–4286, 10.1109/CVPR.2019.00441.
16. Z. Zheng, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, Distractor-Aware Siamese Networks for Visual Object Tracking: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part IX, 2018, pp. 103–119, 10.1007/978-3-030-01240-3_7.
17. M. Danelljan, G. Bhat, F. Khan, and M. Felsberg, ATOM: Accurate Tracking by Overlap Maximization, pp. 4655–4664, 2019, 10.1109/CVPR.2019.00479.
18. B. Ye, H. Chang, B. Ma, S. Shan, and X. Chen, Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework, pp. 341–357, 2022, 10.1007/978-3-031-20047-2_20.
19. Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, Siamese Box Adaptive Network for Visual Tracking, 2020, 10.1109/cvpr42600.2020.00670.
20. B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search, 2021, 10.48550/arXiv.2104.14545.
21. M. Zhao, K. Okada, and M. Inaba, TrTr: Visual Tracking with Transformer, 2021, 10.48550/arXiv.2105.03817.
22. B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, Learning Spatio-Temporal Transformer for Visual Tracking, 2021, 10.48550/arXiv.2103.17154.

23. B. Kang, X. Chen, D. Wang, H. Peng, and H. Lu, Exploring Lightweight Hierarchical Vision Transformers for Efficient Visual Tracking, pp. 9578–9587, 2023, 10.1109/ICCV51070.2023.00881.

24. B. Kang, X. Chen, J. Zhao, C. bo, D. Wang, and H. Lu, Exploiting Lightweight Hierarchical ViT and Dynamic Framework for Efficient Visual Tracking, International Journal of Computer Vision, 2025, Vol. 133, pp. 6689–6711, 10.1007/s11263-025-02500-9.

25. M. Fei, Z. Ju, X. Zhen, and J. Li, Real-time visual tracking based on improved perceptual hashing, Multimedia Tools and Applications, 2017, Vol. 76, 10.1007/s11042-016-3723-5.

26. V. Naumenko, S. Abramov, and V. Lukin, COMPARATIVE ANALYSIS OF IMAGE HASHING ALGORITHMS FOR VISUAL OBJECT TRACKING, 2025, 10.32620/reks.2025.1.09.

27. R. Gonzalez, R. Woods, and B. Masters, Digital Image Processing, Third Edition, Journal of Biomedical Optics, 2009, Vol. 14, pp. 29901, 10.1117/1.3115362.

28. Z. Wang, Q. An, Z. Zhu, H. Fang, and Z. Huang, Blind Additive Gaussian White Noise Level Estimation from a Single Image by Employing Chi-Square Distribution, Entropy, 2022, Vol. 24, pp. 1518, 10.3390/e24111518.

29. Y. Li, Z. Li, K. Wei, W. Xiong, J. Yu, and bo qi, Noise Estimation for Image Sensor Based on Local Entropy and Median Absolute Deviation, Sensors, 2019, Vol. 19, pp. 339, 10.3390/s19020339.

30. H. Liu, L. Hou, Z. Luo, Y. Zhou, X. Jing, and T.-K. Truong, Image Recovery with Data Missing in the Presence of Salt-and-Pepper Noise, Applied Sciences, 2019, Vol. 9, pp. 1426, 10.3390/app9071426.

31. C.-T. Lu, Y.-Y. Chen, L.-L. Wang, and C.-F. Chang, Removal of salt-and-pepper noise in corrupted image using three-values-weighted approach with variable-size window, Pattern Recognition Letters, 2016, Vol. 80, 10.1016/j.patrec.2016.06.026.

32. R. Ali and R. Hardie, Recursive non-local means filter for video denoising, EURASIP Journal on Image and Video Processing, 2017, Vol. 2017, pp. 29, 10.1186/s13640-017-0177-2.

33. M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, Video Denoising, Deblocking, and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms, IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, 2012, Vol. 21, pp. 3952–3966, 10.1109/TIP.2012.2199324.

34. W. Li, J. Zhang, and Q. Dai, Video denoising using shape-adaptive sparse representation over similar spatio-temporal patches, Sig. Proc.: Image Comm., 2011, Vol. 26, pp. 250–265, 10.1016/j.image.2011.04.005.

35. M. Mueller, N. Smith, and B. Ghanem, A Benchmark and Simulator for UAV Tracking, 2016.

36. H. Fan, H. Bai, L. Lin, et al., LaSOT: A High-quality Large-scale Single Object Tracking Benchmark, International Journal of Computer Vision, 2021, Vol. 129No. 2, pp. 439–461, 10.1007/s11263-020-01387-y.

37. L. Huang, X. Zhao, and K. Huang, GOT-10k: A Large High-Diversity Benchmark for Generic Object Tracking in the Wild, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, Vol. 43No. 5, pp. 1562–1577, 10.1109/TPAMI.2019.2957464.

38. I. Karakostas, V. Mygdalis, and I. Pitas, Explaining and verifying the robustness of Visual Object Trackers to noise, 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), pp. 1–5, 2022, 10.1109/IVMSP54334.2022.9816343.

39. I. Karakostas, V. Mygdalis, N. Nikolaidis, and I. Pitas, Enhancing visual object tracking robustness through a lightweight denoising module, The Visual Computer, 2025, Vol. 41, pp. 8627–8644, 10.1007/s00371-025-03888-8.